Review Article

Quantum machine learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications

Yaswitha Gujju[*]
Department of Computer Science, University of Tokyo 4-6-1, Shirokanedai, Minato-ku, Tokyo 108-8639, Japan

Atsushi Matsuo[†]
IBM Quantum, IBM Research – Tokyo, 19–21 Nihonbashi Hakozaki-cho, Chuo-ku, Tokyo, 103-8510, Japan

Rudy Raymond[‡,§]
Department of Computer Science, University of Tokyo 4-6-1, Shirokanedai, Minato-ku, Tokyo 108-8639, Japan
Global Technology and Applied Research, J.P. Morgan Chase & Co., New York, USA
and Quantum Computing Center, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama,
Kanagawa 223-8522, Japan

(Received 5 December 2023; revised 14 April 2024; accepted 16 April 2024; published 4 June 2024)

The past decade has witnessed significant advancements in quantum hardware, encompassing improvements in speed, qubit quantity, and quantum volume—a metric defining the maximum size of a quantum circuit effectively implementable on near-term quantum devices. This progress has led to a surge in quantum machine learning (QML) applications on real hardware, aiming to achieve quantum advantage over classical approaches. This survey focuses on selected supervised and unsupervised learning applications executed on quantum hardware, specifically tailored for real-world scenarios. The exploration includes a thorough analysis of current QML implementation limitations on quantum hardware, covering techniques like encoding, ansatz structure, error mitigation, and gradient methods to address these challenges. Furthermore, the survey evaluates the performance of QML implementations in comparison to classical counterparts. In conclusion, we discuss existing bottlenecks related to applying QML on real quantum devices and propose potential solutions to overcome these challenges in the future.

DOI: 10.1103/PhysRevApplied.21.067001

I. INTRODUCTION

Machine learning (ML) is ubiquitous, with applications spanning image recognition, healthcare diagnosis, text translation, anomaly detection, and physics. In parallel, near-term quantum devices have shown potential in addressing classically intractable problems, even with the challenges of noise and limited qubit connectivity [1,2]. While quantum factoring algorithms, such as Shor’s [3], remain challenging, there have been notable successes, like the factorization of $N = 15$ using nuclear spins as quantum bits with room-temperature liquid-state nuclear magnetic resonance (NMR) techniques [4]. The combination of quantum computing [5,6] and machine learning, termed quantum machine learning (QML) [7–9], has become an active research area, with great advancements being made in the last decade. Within QML, subdomains arise based on the data and algorithm types, whether classical or quantum. In this survey, we delve into different aspects of QML, specifically focusing on algorithms that leverage real quantum hardware, in either supervised or unsupervised contexts. In addition to the paradigms mentioned, reinforcement learning represents the third paradigm. Although not addressed in our current survey, we direct readers to [10–12] for a comprehensive overview of the literature on quantum reinforcement learning.

The past decade has witnessed significant advancements in the performance of quantum hardware, including the number of qubits, speed, and quantum volume [13]. Consequently, there has been an increase in the number of works implementing quantum machine learning on real hardware. The common objective of these works is to demonstrate the advantages of utilizing quantum computers, with their unique properties, such as entanglement and superposition, for practical machine learning tasks. To gain a comprehensive understanding of the current performance and limitations of near-term quantum devices in QML, it

[*] Corresponding author: yaswitha-gujju@g.ecc.u-tokyo.ac.jp, gujjunotgujju@gmail.com

[†] Corresponding author: matsuoa@jp.ibm.com

[‡] Corresponding author: raymond.putra@jpmchase.com

[§] Part of this work was written while R.R. was with IBM Research, Tokyo.

is necessary to conduct a thorough study. In this survey, we aim to consolidate and analyze works that involve the implementation of QML on real hardware to assess their performance.

There has been a growing trend in utilizing quantum computing for commercial and industrial applications, as evidenced by several studies [14–23]. In light of these recent publications, our focus is specifically directed toward exploring applications and techniques that hold relevance for real-world scenarios. Consequently, we have identified high-energy physics [24–26], finance [27–30], and healthcare [31–33] as the domains of interest for our survey. Moreover, we recognize quantum chemistry as another promising field where QML holds substantial potential. For example, a recent paper [34] claims to be the first instance of a quantum-classical generative model, trained on a 16-qubit IBM quantum computer, that yields experimentally confirmed biological hits for designing small molecules in cancer therapy, thereby indicating its practical potential in drug discovery. In particular, introducing innovative methodologies for molecular simulation [35–40] is a promising avenue. However, due to the similar nature of QML application in quantum chemistry and high-energy physics (HEP), both primarily focused on simulating complex quantum systems to understand their properties, we have chosen to exclude quantum chemistry from our current study.

Two main QML frameworks have gained widespread use due to their ability to be implemented with relative ease on quantum hardware, and their demonstrated capacity to work on general datasets. These frameworks are the quantum kernel methods [41–47] and the variational quantum algorithms [47–54]. The quantum kernel method involves building a kernel similar to the technique used in support vector machines (SVM) [55,56]. On the other hand, the variational quantum algorithm employs a parameterized quantum circuit (PQC) whose parameters must be optimized. An overview of the present paper is given in Fig. 1.

For a more thorough review of quantum machine learning frameworks designed to solve classification problems such as support vector machines, kernel methods, decision tree classifiers, nearest-neighbor algorithms, and annealing-based classifiers, we recommend referring to Ref. [57]. That article also discusses the vulnerability of quantum classifiers in adversarial learning.

The main aim of this present study is to comprehend and emphasize the constraints and methods applied in various fields that use different datasets and algorithms to run on the current ion-trap and superconducting-based quantum hardware. It is noteworthy that quantum computing includes hardware architectures beyond gate-based systems, such as D-Wave's quantum computer that employs quantum annealing [58–60]. However, for this study, we will focus on gate-based architectures, as they follow the circuit model paradigm, which makes them different from other approaches.

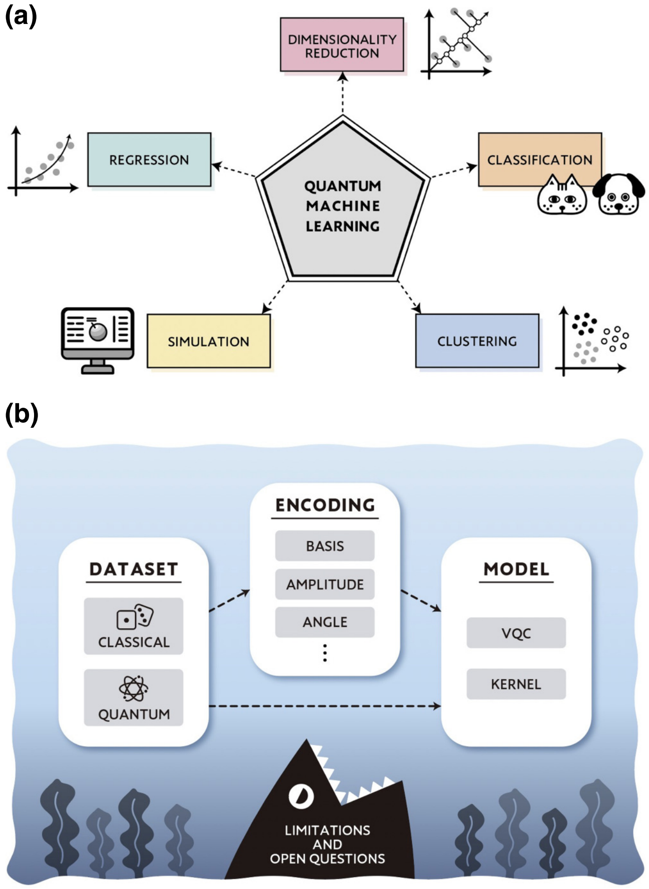


FIG. 1. (a) Applications of QML. These are the different subproblems identified among the papers surveyed for real-world domains that include high-energy physics, healthcare, and finance. (b) Outline of QML. The data can be inherently quantum or classical depending on the application. Consequently, we perform quantum state preparation for classical data. Based on the papers reviewed, we study different encoding techniques for classical data. We primarily focus on variational quantum circuit and kernel models, and study the drawbacks and current challenges in the field.

The current state of quantum machine learning [61–63] faces numerous challenges, largely tied to quantum hardware capabilities. These encompass limited qubit connectivity, noise, coherence times, and errors in both state preparation and measurement. Prolonged running times on quantum hardware further affect the execution and outcomes of QML algorithms. One core challenge is efficiently encoding classical data into quantum features. Alongside this, loading and storing prepared quantum states while resisting decoherence is a significant challenge. After preparing the states, it is crucial to develop efficient quantum algorithms. Challenges vary based on

TABLE I. Papers that deal with QML in the field of high-energy physics, highlighting the problem type, number of qubits, quantum hardware type, training approach, and specific QML methodologies employed in each paper.

Reference	Qubits	Hardware type	Trained on	Method
Classification				
Muten *et al.* [64]	1	Superconducting	Simulator	VQC
Blance and Spannowsky [65]	2	Superconducting	Simulator	VQC
Terashi *et al.* [66]	3	Superconducting	QPU	VQC
Wu *et al.* [67]	10	Superconducting	QPU	VQC
Wu *et al.* [68]	15	Superconducting	QPU	Kernel
Araz and Spannowsky [69]	6	Superconducting	QPU	QTN
Woźniak *et al.* [70]	8	Superconducting	QPU	Kernel
Li *et al.* [71]	5	Superconducting	QPU	Kernel
Bermot *et al.* [72]	3	Superconducting	QPU	QGAN
Cugini *et al.* [73]	5	Superconducting	Simulator	VQC
Peixoto *et al.* [74]	5	Superconducting	Simulator	VQC
Lazar *et al.* [75]	8	Superconducting	QPU	Parity
Data generation				
Pérez-Salinas *et al.* [76]	8	Superconducting	Simulator	VQC
Bravo-Prieto *et al.* [77]	3	Superconducting/Ion trap	QPU	QGAN
Chang *et al.* [78]	6	Superconducting/Ion trap	QPU	QGAN
Rehm *et al.* [79]	8	Superconducting	QPU	VQC
Clustering				
Ngairangbam *et al.* [80]	4	Superconducting	Simulator	VQC

the type of algorithm, such as kernel-based or variational quantum circuits. For instance, variational algorithms often struggle with issues like barren plateaus. Their training, alongside the choice of optimizers and loss functions, greatly influences efficiency. In contrast, with kernel techniques, selecting the right feature map is essential. Optimization, scalability, and the generalization capabilities of QML models are crucial. It is essential for these algorithms to scale effectively for real-world applications. Moreover, addressing the security and vulnerabilities of QML models is vital to prevent potential adversarial attacks.

The papers are grouped based on real-world applications, whose groups can be found in Table I for high-energy physics, Table II for finance, and Table III for healthcare. Additionally, we include papers using benchmark datasets, such as MNIST and Iris, in Table IV, along with references to papers using quantum datasets in Table V, and artificial datasets in Table VI. In this context, the term “quantum data” refers to data already embedded in a Hilbert space, represented as quantum states or unitaries. This differs from classical data, which require quantum system encoding. The tables provide a comprehensive overview of the included studies. They detail the reference, the number of qubits used, and the specific problem type addressed. Moreover, we specify the type of hardware employed (e.g., superconducting or ion-trapped) and whether the training was done on a quantum processing unit (QPU) or simulator. It is noteworthy that all tests cited in the papers were performed on a QPU. The tables

TABLE II. Papers that deal with QML applications in finance.

Reference	Qubits	Hardware type	Trained on	Method
Data loading				
Zoufal *et al.* [81]	3	Superconducting	QPU	QGAN
Classification				
Ray *et al.* [82]	3	Superconducting	QPU	VQC/Kernel
Suzuki *et al.* [83]	4	Ion trap	QPU	Kernel
Thakkar *et al.* [84]	8	Superconducting	Simulator	VQC
Dimensionality reduction				
Martin *et al.* [85]	4	Superconducting	QPU	QPCA
Feature selection				
Zoufal *et al.* [86]	20	Superconducting	QPU	VQC

TABLE III. Papers that deal with QML in the field of healthcare.

Reference	Qubits	Hardware type	Trained on	Method
Classification				
Yano *et al.* [87]	2	Superconducting	QPU	VQC
Acar *et al.* [88]	4	Superconducting	QPU	VQC
Mathur *et al.* [89]	9	Superconducting	QPU	VQC
Ren *et al.* [90]	10	Superconducting	QPU	VQC
Moradi *et al.* [91]	7	Superconducting	QPU	Kernel
Krunic *et al.* [92]	20	Superconducting	QPU	Kernel
Azevedo *et al.* [93]	4	Superconducting	Simulator	VQC
Vasques *et al.* [94]	5	Superconducting	QPU	Kernel
Mensa *et al.* [95]	8	Superconducting	QPU	Kernel
Innan *et al.* [96]	4	Superconducting	Simulator	VQC
Inference				
Benedetti *et al.* [97]	5	Superconducting	QPU	VQC

further delve into the quantum models utilized, including quantum generative adversarial network (QGAN), variational quantum circuit (VQC), quantum tensor network (QTN), quantum principal component analysis (QPCA), and quantum K means (Q-K means).

The paper is organized as follows. Firstly, we present a summary of the notation used in the paper in Sec. II. Subsequently, we offer an overview of fundamental concepts in classical machine learning in Sec. III, quantum computing in Sec. IV, and quantum machine learning in Sec. V. In Sec. VI, we explore the applications of quantum machine learning techniques, with a specific focus on kernel techniques and variational quantum classifiers, categorized into supervised and unsupervised learning. Then, Sec. VII delves into the limitations related to hardware and algorithms. We conclude with discussions on current bottlenecks and proposing possible solutions for future research in Sec. VIII.

TABLE IV. List of QML papers that use standard datasets such as Iris [98,99], MNIST [100], FashionMNIST [101], Titanic Survival [102], Astronomical [103], and Wine [104].

Reference	Qubits	Hardware type	Trained on	Method	Dataset
Classification					
Li *et al.* [105]	4	NMR	QPU	Kernel	Handwritten
Grant *et al.* [106]	4	Superconducting	Simulator	TN	Iris
Cappelletti *et al.* [107]	2	Superconducting	QPU	VQC	Iris
Thumwanit *et al.* [108]	3	Superconducting	QPU	VQC	Titanic Survival
Peters *et al.* [109]	17	Superconducting	QPU	Kernel	Astronomical
Abbas *et al.* [49]	4	Superconducting	QPU	VQC	Iris
Blank *et al.* [110]	5	Superconducting	QPU	Kernel	Iris/Wine
Ren *et al.* [90]	10	Superconducting	QPU	VQC	MNIST/FashionMNIST
Suzuki *et al.* [83]	4	Ion trap	QPU	Kernel	MNIST/FashionMNIST
Koyasu *et al.* [111]	3	Superconducting	QPU	VQC	MNIST/FashionMNIST
Haug *et al.* [112]	8	Superconducting	QPU	Kernel	MNIST
Simoes *et al.* [113]	5	Superconducting	QPU	Kernel/VQC	VLDS [114]/Iris
Chen *et al.* [115]	4	Superconducting	QPU	VQC	MNIST/FashionMNIST
Melo *et al.* [116]	9	Superconducting	QPU	Kernel	MNIST
Anagolum *et al.* [117]	10	Superconducting	QPU	VQC	MNIST
Shen *et al.* [118]	11	Superconducting	Simulator	VQC	FashionMNIST
Innan *et al.* [96]	4	Superconducting	Simulator	VQC	Iris
Clustering					
Khan *et al.* [119]	4	Superconducting	QPU	Q-K means	Iris/MNIST
Johri *et al.* [120]	8	Ion trap	QPU	Nearest centroid	Iris/MNIST
Simulation					
Huang *et al.* [121]	5	Superconducting	QPU	QGAN	Handwritten
Rudolph *et al.* [122]	8	Ion trap	QPU	QGAN	MNIST

TABLE V. QML papers that primarily utilize quantum data. These data can consist of intrinsically quantum information or classical information that is transformed into a quantum feature space.

Reference	Qubits	Hardware type	Trained on	Method
Simulation				
Gibbs *et al.* [123]	2	Superconducting	QPU	VQC
Bartkiewicz *et al.* [124]	3	Superconducting	QPU	VQC
Classification				
Blank *et al.* [41]	5	Superconducting	QPU	Kernel
Herrmann *et al.* [125]	7	Superconducting	QPU	VQC
Ren *et al.* [90]	10	Superconducting	QPU	VQC
Gong *et al.* [126]	61	Superconducting	QPU	VQC
Data generation				
Bartkiewicz *et al.* [124]	3	Superconducting	QPU	VQC
Training				
Pan *et al.* [127]	6	Superconducting	QPU	VQC
Clustering				
Huang *et al.* [128]	40	Superconducting	QPU	Kernel
Nakayama *et al.* [129]	4	Superconducting	QPU	Kernel

II. NOTATION

Throughout the paper, the dataset is denoted as $\mathcal{D} = \{(x^1, y^1), \ldots, (x^m, y^m)\}$, where $\mathcal{D}$ represents the dataset for supervised learning containing m samples or observations. For unsupervised learning, the data do not contain labels and are represented as $\mathcal{D} = \{x^1, \ldots, x^m\}$. Each x^i represents the ith input data sample and can be understood as a vector defined as $x^i = [x^i_1, x^i_2, \ldots, x^i_d]$, where d is the number of features in the input data. The corresponding label or output associated with x^i is represented by y^i.

Moving over to the vector spaces, we represent the N-dimensional Hilbert space as $\mathcal{H}^N$ for a system with n qubits such that $N = 2^n$. The complex space is represented using $\mathbb{C}$, while the real space is denoted as $\mathbb{R}$. The feature map is denoted as ϕ. The quantum gates are represented as H for the Hadamard gate, X for the Pauli-X gate, Y for the Pauli-Y gate, Z for the Pauli-Z gate, and U for the unitary operator. The gate parameters are denoted using θ. The depth of the encoder part of the circuit is represented as $N^{\text{in}}_{\text{depth}}$, while $N^{\text{var}}_{\text{depth}}$ is used to represent the depth of the variational part of the circuit. A list of the abbreviations and acronyms used throughout the paper, along with their full forms, is given in Table VII.

III. CLASSICAL MACHINE LEARNING

The field of artificial intelligence (AI) has become omnipresent, with many practical applications, such as automation of routine labor, speech recognition, computer

TABLE VI. Papers that use artificial datasets which are synthetically created to evaluate and benchmark QML algorithms.

Reference	Qubits	Hardware type	Trained on	Method
Classification				
Havlíček *et al.* [47]	2	Superconducting	QPU	Kernel/VQC
Bartkiewicz *et al.* [46]	2	Photonic	QPU	Kernel
Melo *et al.* [116]	5	Superconducting	QPU	VQC
Simoes *et al.* [113]	4	Superconducting	QPU	Kernel/VQC
Glick *et al.* [130]	27	Superconducting	QPU	Kernel
Heese *et al.* [131]	3	Superconducting	Simulator	QGAN
Gentinetta *et al.* [132]	7	Superconducting	QPU	Kernel
Data generation				
Huang *et al.* [133]	5	Superconducting	QPU	QGAN
Clustering				
Johri *et al.* [120]	8	Ion trap	QPU	Nearest centroid
Regression				
Kreplin *et al.* [134]	10	Superconducting	QPU	QNN

TABLE VII. The abbreviations and acronyms used throughout the paper, along with their expansions.

Acronym	Full form
CC	Classical computing
NN	Neural network
NN-GD	Neural network with gradient descent
QC	Quantum computing
VQC	Variational quantum circuit
VQC-QGD	Variational quantum circuit with quantum gradient descent
VQC-GD	Variational quantum circuit with gradient descent
QML	Quantum machine learning
VQA	Variational quantum algorithm
PQC	Parameterized quantum circuit
QKE	Quantum kernel estimation
QSVM	Quantum support vector machine
FOE	First-order expansion
SOE	Second-order expansion
HEP	High-energy physics
QAE	Quantum amplitude estimation
LHC	Large Hadron Collider
BDT	Boosted decision tree
DNN	Deep neural network
MSE	Mean-square error
DRC	Data reuploading-based classifier
TN	Tensor networks
QTN	Quantum-inspired tensor networks
TPR	True positive rate
FPR	False positive rate

vision, etc. To avoid depending on hard-coded knowledge, it is essential for these AI systems to acquire knowledge from their surroundings by solving a learning problem [135]. Machine learning [135–137] is an evolving branch of artificial intelligence that is essentially devoted to solving such problems where the goal is to improve some measure of performance when executing a task, through some type of training experience. ML models are trained on sample data, called training data, which enables them to learn properties of the data and make predictions or decisions accordingly. In this survey, we look at supervised and unsupervised learning.

A. Supervised learning

Here, the model is provided with labeled data. To measure the performance, the model is evaluated on unseen data, called testing data. Two common types of supervised learning algorithms include classification and regression. Training involves minimizing the cost function over the input data and adjusting its weights until the model has been fitted appropriately. Examples of common classifiers include linear classifiers, support vector machines, random forest, etc. In regression-type problems, the goal is to fit a function over the data (independent variables) to predict the output. Commonly used regression models include linear regression, support vector regression, etc.

B. Unsupervised learning

On the other hand, unsupervised learning involves training the model to analyze and cluster unlabeled datasets with the goal of discovering hidden patterns or structure in the data. In addition to clustering, which involves finding structure in the data by grouping similar points and separating dissimilar points, unsupervised learning also includes dimensionality reduction techniques, such as principal component analysis (PCA), autoencoders, singular-value decomposition, etc. Here, the objective is to reduce the dimension of the data without losing too much information.

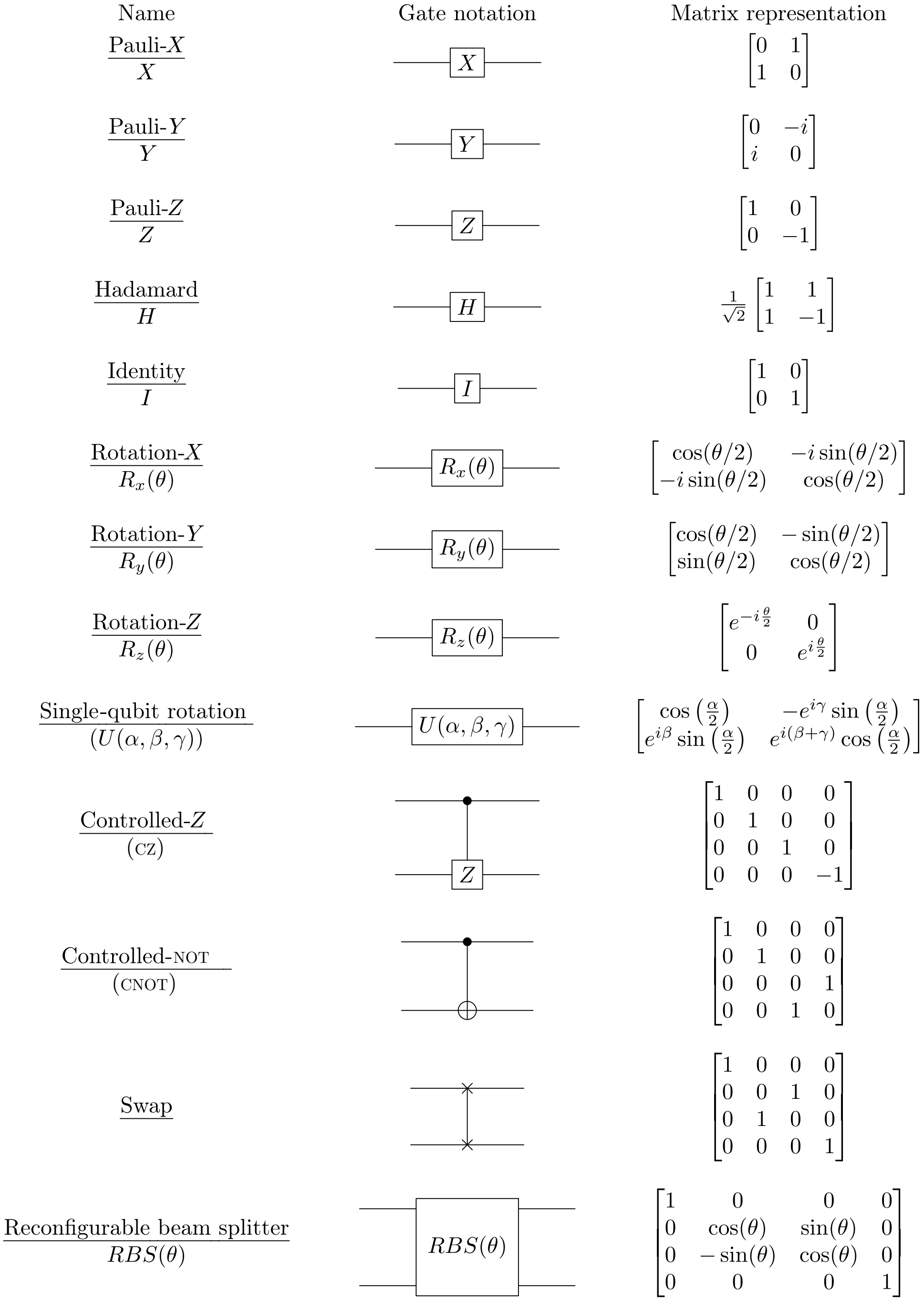
IV. QUANTUM COMPUTING

The phenomenon of quantum superposition and entanglement is what gives quantum computing an edge over classical computing. This can translate to significant speedup or reduced computational resources in terms of time and space. Here, we briefly discuss the basics of quantum computing. The basic unit of quantum computation is the qubit,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$ and $|0\rangle, |1\rangle$ represent the computational basis in the two-dimensional Hilbert space $\mathcal{H}$. The absolute squares of the amplitudes (i.e., $|\alpha|^2$ and $|\beta|^2$) are the probabilities to measure the qubit in either the 0 or 1 state, respectively, such that $|\alpha|^2 + |\beta|^2 = 1$. As such, $|\psi\rangle$ as can be rewritten as $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$, where $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$ are real numbers.

Unitary matrices (quantum gates) can be applied to quantum states to transform them into other quantum states to ensure that the condition on the amplitude-based probabilities is maintained even after the transformation. Through single-qubit quantum gates we can manipulate the basis state, amplitude, or phase of a qubit (for example, through the so-called X gate, the Z gate, and the Y gate, respectively), or put a qubit with $\beta = 0$ $(\alpha = 0)$ into an equal superposition: $\alpha = 1/\sqrt{2}$, $\beta = \pm 1/\sqrt{2}$ (the Hadamard or H gate). Multiqubit gates are often based on controlled operations that execute a single-qubit operation only if another (ancilla or control qubit) is in a certain state. One of the most important gates is the two-qubit controlled-NOT (CNOT or CX) gate, which flips the basis state of the target qubit when the control qubit is in state $|1\rangle$. A set of arbitrary one-qubit rotation gates and two-qubit CNOT gates is universal, which means that any quantum operation can be implemented using a combination of these basic gates. We list typical quantum gates



Name	Gate notation	Matrix representation
Pauli-X X	X	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
Pauli-Y Y	Y	$\begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$
Pauli-Z Z	Z	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
Hadamard H	H	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
Identity I	I	$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$
Rotation-X $R_x(\theta)$	$R_x(\theta)$	$\begin{bmatrix}\cos(\theta/2) & -i\sin(\theta/2)\\-i\sin(\theta/2) & \cos(\theta/2)\end{bmatrix}$
Rotation-Y $R_y(\theta)$	$R_y(\theta)$	$\begin{bmatrix}\cos(\theta/2) & -\sin(\theta/2)\\\sin(\theta/2) & \cos(\theta/2)\end{bmatrix}$
Rotation-Z $R_z(\theta)$	$R_z(\theta)$	$\begin{bmatrix}e^{-i\frac{\theta}{2}} & 0\\0 & e^{i\frac{\theta}{2}}\end{bmatrix}$
Single-qubit rotation $(U(\alpha,\beta,\gamma))$	$U(\alpha,\beta,\gamma)$	$\begin{bmatrix}\cos\left(\frac{\alpha}{2}\right) & -e^{i\gamma}\sin\left(\frac{\alpha}{2}\right)\\e^{i\beta}\sin\left(\frac{\alpha}{2}\right) & e^{i(\beta+\gamma)}\cos\left(\frac{\alpha}{2}\right)\end{bmatrix}$
Controlled-Z (CZ)	Z	$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & -1\end{bmatrix}$
Controlled-NOT (CNOT)		$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$
Swap		$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 1 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}$
Reconfigurable beam splitter $RBS(\theta)$	$RBS(\theta)$	$\begin{bmatrix}1 & 0 & 0 & 0\\0 & \cos(\theta) & \sin(\theta) & 0\\0 & -\sin(\theta) & \cos(\theta) & 0\\0 & 0 & 0 & 1\end{bmatrix}$

FIG. 2. Description of quantum gates in the order of single and multiple qubits.

(along with their symbols, and their matrix forms) used in quantum circuits for quantum machine learning in Fig. 2.

V. QUANTUM MACHINE LEARNING

A timeline trend and coverage distribution of works related to quantum machine learning on arXiv are shown in Fig. 3. In this study, we focus on two widely used QML algorithms based on variational quantum circuits and quantum kernel methods. In both of these approaches, we start by encoding d-dimensional classical data so that they are embedded as a quantum state vector in the Hilbert space. By doing so, we can exploit the exponential dimensionality of the Hilbert space, which grows with the number of qubits, giving it a stronger representational power over the classical feature space, which may help capture strong correlation between variables. Both the models involve data

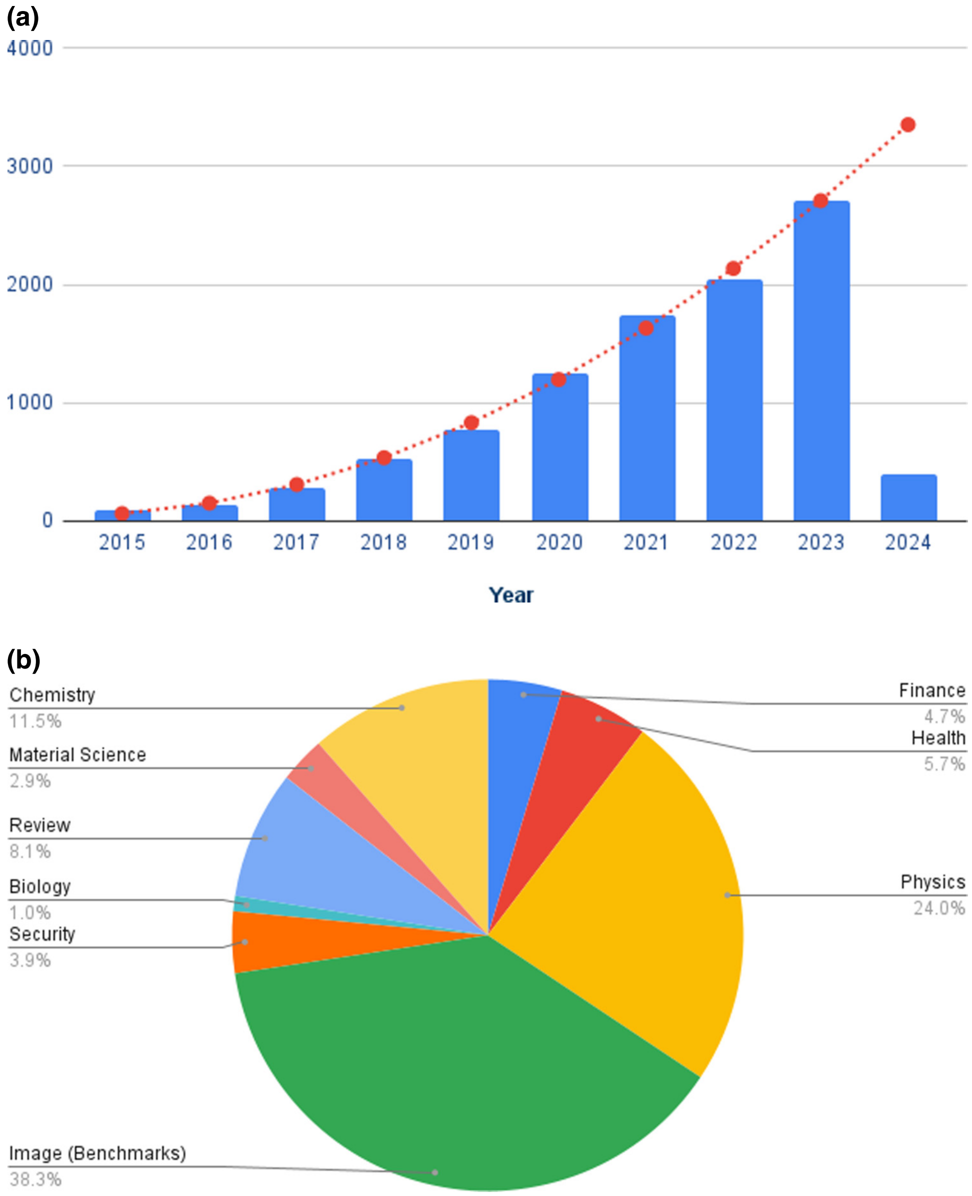


FIG. 3. (a) Yearly plot displaying the frequency of works on arXiv, utilizing the search term “quantum machine learning” in the title and/or abstract [138]. (b) The distribution of search terms related to various real-world applications in the titles and/or abstracts of around 1000 papers sampled that were obtained using the arXiv API under the “quant-ph” category in the last five years. For papers belonging to multiple domains, we include them in each of the respective categories to calculate the final distribution. We see that the collection of physics, finance, and healthcare constitutes approximately 34% of the applications queried using the API.

encoding but differ in the way the quantum state is handled. We look at some of the most commonly used techniques for encoding classical data along with the different QML models.

A. Encoding datasets

Quantum machine learning involves learning from either classical or quantum data. It is more likely to obtain exponential quantum advantage in machine learning when data come from quantum-mechanical processes [139]. Classical data are encoded in bits (0s and 1s), such as images, text, medical records, etc. Quantum data, on the other hand, are encoded in quantum bits called qubits, which can represent states beyond 0 and 1. Qubits can contain information from physical processes like quantum sensing or quantum control. While classical data can be efficiently encoded in qubits, the reverse is not true. In QML, quantum data refer to data already in a quantum state, while classical data need to be encoded into a quantum system.

A requisite for obtaining quantum advantage in both VQC and quantum kernel estimation (QKE) [45] on classical datasets is that embedding or encoding the datasets has to be efficiently implementable on quantum circuits to avoid the so-called data-loading problem [140]. The quantum embeddings help represent classical data as quantum states in the Hilbert space, thereby allowing us to truly harness the power of quantum systems. Some desirable properties of an encoder are that the number of gates required to implement the encoder must be at most polynomial in the number of qubits, and the intractability by any classical operation to simulate it is preferred. Additionally, it is ideal to have a bijective encoding such that there is a unique quantum state ρ_{x_i} for each sample x_i. Finally, the single- and two-qubit gates required to implement the encoder should be compatible with the native gate set of the near-term quantum devices so that the compilation of the circuit is hardware-efficient [141]. Thus, data encoding plays an important role as it determines the features that quantum models represent [43,47], the decision boundaries learnt [141], and measurements that optimally distinguish between data classes [142].

1. Basis encoding

This is the simplest and one of the most common encodings [53,143], which maps a binary string of classical data $x = x_1 \ldots x_n$ into the computational basis $|x\rangle = |x_1 \ldots x_n\rangle$. It requires n qubits to encode n bits of classical data, and is useful to feed one sample classical bit at a time to a QML model. The power of the quantum resource comes when the batches of classical samples are represented as superpositions of basis states [144]. Quantum bits can be used to create quantum states that are a superposition of classical datasets, i.e., *quantum batches*.

In the case of supervised learning, as pointed out in Ref. [53], one can create quantum states $|+1\rangle$ and $|-1\rangle$, each of which is a superposition of the basis encoding of samples with label $l(x)$ as $+1$ and -1, respectively, as below (omitting ancilla and working qubits), and use them to train a QML model on the superposition states of real-world data:

$$|+1\rangle = \frac{1}{\sqrt{N_+}} \sum_{x:\, l(x)=+1} |x\rangle ,$$

$$|-1\rangle = \frac{1}{\sqrt{N_-}} \sum_{x:\, l(x)=-1} |x\rangle ,$$

where N_+ and N_- are, respectively, the number of samples with label $+1$ and -1. It is argued in Ref. [53] that the above quantum batches can result in training a QML model with smoother loss fluctuation and can be more efficient in the sample complexity for better generalization error than individual samples.

2. Amplitude encoding

The classical data x, which is a d-dimensional vector, is encoded into the amplitude of the quantum state [81,145–147]. Namely, for $x = (x_1, \ldots, x_d)$ such that $\sum_i |x_i|^2 = 1$, the corresponding encoding is the quantum state

$$|\psi_x\rangle = \sum_{i=1}^{d} x_i \, |i\rangle ,$$

which requires only $\log d$ qubits to store x. The advantage of this encoding is in the exponential memory saving and, if one can design a QML model that runs in polynomial time in the size of the number of qubits, then there are hopes for exponential quantum advantage. In fact, many QML models that promise quantum advantages use this encoding combined with quantum basic linear algebras, such as HHL [148] and others (see, e.g., [149–152]). The main drawback is that quantum circuits that generate $|\psi_x\rangle$ can require quantum circuits with an exponential number of native gates [153–156], and hence the data-loading problem [140].

To avoid exponential circuit complexity, recent works [157,158] propose the use of *unary amplitude encoding* to encode x using a d-qubit quantum state (i.e., a qubit per feature) as

$$|\phi_x\rangle = \sum_{i=1}^{d} x_i \, |e_i\rangle ,$$

where $|e_i\rangle$ is the ith unary computational basis $|0 \ldots 010 \ldots 0\rangle$ with "1" only at the ith qubit. It is shown that the depth of the circuit to generate unary encoding is logarithmic in d [158], and linear using a cascade of reconfigurable beam-splitter (RBS) gates [157].

3. Divide-and-conquer approach

This data-loading technique is a modified version of amplitude encoding and is introduced in Ref. [159] using controlled-SWAP gates and ancilla qubits. As the name suggests, this method is based on the divide-and-conquer approach and derives motivation from Ref. [160]. The d-dimensional input vector is loaded in the probability amplitudes of the computational basis state with entangled information in ancillary qubits. The results show exponential time advantage using a quantum circuit with polylogarithmic depth and $O(d)$ qubits. However, the reduced circuit depth comes at a cost of increasing the circuit width and creating additional entanglement between data register qubits and an ancillary system.

4. Angle encoding

While the aforementioned amplitude encodings require circuits with at least $O(\log d)$ depth, one can load x with constant-depth quantum circuits by embedding $x_i \in \mathbb{R}$, i.e., the ith element of x, as a parameter of Pauli rotational gates $R_X(x_i) \equiv e^{-ix_iX/2}$, or $R_Y(x_i) \equiv e^{-ix_iY/2}$, or $R_Z(x_i) = e^{-ix_iZ/2}$. The data also need to be normalized or scaled using min-max scaling in a suitable range to be evaluated as gate angles, and the choice of this range can influence the performance. For example, in Ref. [68], the use of angles in the range $[-1, +1]$ was found to be more optimal than $[-\pi, +\pi]$. For example, starting from the all-zero quantum state, one can create the following n-qubit quantum state (where $n = d$) representing x by applying $R_Y(x_i)$ to the ith qubit for $i = 0, \dots, d-1$:

$$|x\rangle \equiv \bigotimes_{i=0}^{d-1} R_Y(x_i)\,|0\rangle^d = \bigotimes_{i=0}^{d-1} \cos\left(\frac{x_i}{2}\right)|0\rangle + \sin\left(\frac{x_i}{2}\right)|1\rangle\,.$$

The above quantum state is a product state that can be represented classically in $O(d)$ computational space and time, but, when combined with entanglement layers and their block repetitions, the angle encoding can be used as a building block to generate sophisticated entangled states that are difficult to compute classically.

Also worth mentioning are the so-called *first-order encoding* (FOE) and *second-order encoding* (SOE) as defined in Ref. [47]. In FOE, to encode $x_k \in \mathbb{R}$, the single-qubit gates $R_Z(x_k)$ are used. This can be lifted to a higher encoding using SOE where more parameters are used along with entangling gates. For example, to encode $x_l, x_m \in \mathbb{R}$ along with their correlation in the lth and mth qubits, SOE utilizes the gate $e^{i(\pi-x_l)(\pi-x_m)Z_lZ_m}$.

When the classical data x are a bit string of length d, which is often used to represent discrete features, Yano *et al.* [87] propose to utilize the so-called *quantum random-access codes* (QRAC) to obtain a constant factor saving in the number of qubits. For example, the previous $|e_i\rangle$ is known as one-hot encoding in classical machine learning that requires d qubits. With the QRAC encoding, the bit string $x = x_0 \dots x_{d-1} \in \{0,1\}^d$ can be represented with $\lceil d/3 \rceil$-qubit quantum state ρ_x as below:

$$\rho_x \equiv |\psi_x\rangle\langle\psi_x| = \bigotimes_{i=0}^{d/3-1} \frac{1}{2}\Bigg(I + \frac{1}{\sqrt{3}}((-1)^{x_{3i}}X + (-1)^{x_{3i+1}}Y + (-1)^{x_{3i+2}}Z)\Bigg),$$

where for simplicity $d > 0$ is assumed to be divisible by 3. Notice that the value of x_{3i+j} can be retrieved by measuring the ith qubit of ρ_x in X, Y, or Z bases for $j = 0,\ 1,\ 2$, respectively. The QRAC encoding can be run with a single-qubit gate for each qubit.

a. Data reuploading. Angle encoding applies a Pauli rotation gate whose degree of freedom is one, say for $x_j \in \mathbb{R}$, the $R_Z(x_j)$ at the jth qubit. Meanwhile, it is known that a general single-qubit rotation gate $U(\cdot)$ has three degrees of freedom and is represented by the matrix form in Fig. 2.

First proposed in Ref. [161], the data reuploading technique utilizes the above $U(\cdot)$ to encode three elements of x in a qubit. By repeating the application of $U(\cdot)$ each with different three elements of x for $j \in \{0, \dots, d/3 - 1\}$, hence the reuploading, the whole data point x can be encoded in a single qubit. We can easily see that the data reuploading is the angle encoding repeated with different parameters x_j because the above $U(\cdot)$ can be decomposed into a sequence of Pauli rotation gates as below:

$$U(x_{3j}, x_{3j+1}, x_{3j+2}) = R_Z(x_{3j+1} + \pi)\sqrt{X}\,R_Z(x_{3j} + \pi)\sqrt{X}\,R_Z(x_{3j+2}).$$

The parameters of data reuploading can be linearly transformed before being used in $U(\cdot)$ or trained to fit the prediction [161]. This method has been used in a variety of applications, including drug discovery [162,163], image classification of MNIST dataset [164], and variational quantum eigensolver [165]. Because of the structure of single-qubit unitary gates, this encoding is particularly suited for data with rotational symmetry.

5. Hamiltonian encoding

While the encoding quantum state from angle encoding is obtained by transforming the all-zero quantum state with single-qubit rotational gates which are classically computable, Hamiltonian encoding evolves the all-zero quantum state according to the Hamiltonian parameterized by x to generate highly entangled states. Namely, let the Hamiltonian be $H(x) = \sum_i f_i(x) h_i$, where $f_i(x) \in \mathbb{R}$ is the weight function and $h_i = \bigotimes_{j=1}^{n} \sigma_i^j$ for $\sigma_i^j \in \{I, X, Y, Z\}$. For a fixed t, the quantum state $|\psi_t(x)\rangle$ that encodes x is obtained from the time evolution

$$|\psi_t(x)\rangle = e^{-iH(x)t/\hbar}\,|0\rangle^{\otimes n},$$

which can be run on gate-based quantum hardware using techniques such as Trotterization [166], variational approaches [167–169], and linear combination of unitaries [170,171].

Another example of encoding involving the parameters of a tunable Hamiltonian is discussed in Ref. [172], where the authors present a quantum feature map for graph data on a neutral-atom quantum processor comprising up to 32 qubits. The results demonstrate the ability of the map to effectively capture local and global graph structures while applying this quantum graph kernel to predict toxicity on a real-world dataset of molecules and compare its performance against various classical kernels.

B. Models

1. Quantum kernel estimation

The kernel trick enables one to process higher-dimensional data without explicitly computing the feature vector. This method is most commonly employed in classification using the support vector machine [55,56]. By means of the kernel, every feature map corresponds to a distance measure in input space by means of the inner product of feature vectors [43,173]. The key highlight of kernel tricks with quantum states, or quantum kernels, comes from its ability to compute similarities from the encoding of the classical data into the quantum state space through entanglement (Fig. 8) and interference so as to generate correlations between variables that are classically intractable [42]. This is expected to give more expressive feature embeddings, leading to better performance in pattern recognition and classification tasks compared to the classical counterparts. However, the true advantage does not come from the high-dimensional space (which is also possible using classical kernels) but rather from being able to construct complex circuits which are hard to calculate classically. Even so, while the classical kernels can be computed exactly, the quantum kernels are subject to small additive noise in each kernel entry due to finite sampling, while classical kernels can be computed exactly. To tackle this, error-mitigation techniques have been developed [44,174–176] for cases when the feature map circuit is sufficiently shallow.

The following steps are key components involved in QKE [45]:

- *Quantum feature map.* A feature map ϕ is employed to encode the classical data x to the quantum state space using unitary operations. For any two data points $x^i, x^j \in \mathcal{D}$, the encoded data are represented as $\Phi(x^i)$ and $\Phi(x^j)$, respectively.
- *Inner product.* The kernel entry can be obtained as the inner product between two data-encoded feature vectors $\Phi(x^i)$ and $\Phi(x^j)$, i.e.,

$$\kappa(x^i, x^j) = |\langle\Phi(x^j)|\Phi(x^i)\rangle|^2. \tag{1}$$

  The kernel entry can be estimated by recording the frequency of the all-zero outcome 0^n. This procedure is referred to as quantum kernel estimation. Different methods [177,178] can be employed to estimate the fidelity between general quantum states, one of which is the swap test.

Quantum support vector machines use the kernel built using QKE with a classical SVM. It was first introduced in Ref. [144] while the proof of principle was first demonstrated for classifying handwritten characters in Ref. [105].

The advantage of using quantum kernels is not so apparent when we have large datasets where the quantum cost scales quadratically with the training dataset size [142]. Efficient data encoding and generating useful quantum kernels are constrained by the limited number of qubits and heuristic characterization [45,47,68,161]. Additionally, fewer measurements and large system noise necessitate error-mitigation techniques requiring significant additional quantum resources [174,179]. In Ref. [180], an indefinite kernel learning-based method is implemented to demonstrate the advantage of kernel methods for near-term quantum devices by suppressing the estimation error. Recently, the work in Ref. [112] introduced a novel approach for measuring quantum kernels using randomized measurements showing a linear scaling of features based on circuit depth. The method also incorporates a cost-free error mitigation and offers improved scalability, with the quantum computation time scaling linearly with the dataset size and quadratic scaling for classical postprocessing.

a. Different types of kernels. In the previous subsection, the quantum kernel is computed as the (non-negative) frequency of observing the all-zero bits of running the concatenation of the quantum circuit encoding x^i with the inverse of quantum circuit encoding x^j as in Eq. (1). This type of quantum kernel is quite powerful for classifying artificial data derived from discrete log problems [44], and for classifying group-structured data when the initial state $|0^n\rangle$ in Eq. (1) is replaced with optimized fiducial quantum states computed from kernel alignment [130]. Regarding the latter, experimental results on a 27-qubit device, when the data are encoded with single-qubit rotational gates and the fiducial quantum state is matched with connectivity of qubits in the quantum device, have been demonstrated.

There are many other types of quantum kernels available whose elements are not necessarily restricted to be non-negative. For example, the Hadamard-test classifier (HTC), which encodes real-valued vectors with amplitude encoding, computes the weighted sum of the inner product between a test data vector with the superposition of training data vectors for binary classification [146]. The compact version of HTC has been given by Ref. [110]. While the full quantum space in Eq. (1) seems to be powerful, it is pointed out in Ref. [42] that it can fail to learn a simple function. To overcome this, the *projected quantum kernel*, which projects the quantum kernel into the classical one and computes the elements of kernels from the functions of reduced density matrices, was introduced in Ref. [42] to obtain better quantum kernels that can also learn the data derived from the discrete log problems in Ref. [44].

2. Swap-test classifier

The swap-test classifier (see Fig. 4) as proposed in Ref. [41] is implemented as a distance-based quantum

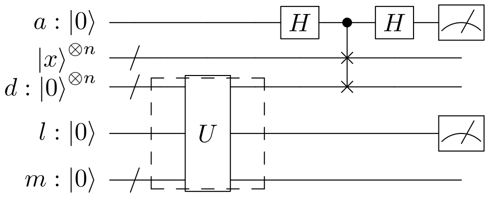


FIG. 4. Swap-test classifier. The first register is the ancilla qubit (a), the second contains n copies of the test datum (x), the third are the data qubits (d), the fourth is the label qubit (l), and the final register corresponds to the index qubits (m). An operator U creates the input state necessary for the classification protocol. The swap test and the two-qubit measurement statistics yield the classification outcome.

classifier where the kernel is based on the quantum state fidelity raised to a certain power at the cost of using multiple copies of training and test data. The choice of the quantum feature map plays a pivotal role in defining the kernel and the overall efficiency of the classifier. The training and test data are encoded in a specific format, following which the classifier is realized by means of the swap test [177].

The swap test measures the similarity between the input quantum state and the reference quantum states for each class using measurements to compute a similarity score that indicates the overlap between the input state and the reference states.

3. Variational quantum circuits

These algorithms primarily focus on optimizing the parameters of the PQC and are known to provide a general framework that is compatible with different classes of problems leading to different structures and grades of complexity. The optimization is performed classically while allowing the circuit to remain shallow, making it a versatile tool for near-term quantum devices.

This basic structure of VQC involves the following three steps:

- *Quantum feature map.* A nonlinear feature map ϕ is employed to encode the classical data x to the quantum state space (see Fig. 5). This is done by applying the circuit $U_{\phi(x)}$ to the initial state $|0\rangle^{\otimes n}$:

$$|\Phi(x)\rangle = U_{\phi(x)}|0\rangle^{\otimes n}.$$

The initial state $|0\rangle^{\otimes n}$ can be replaced by any fiducial quantum state as shown in Ref. [130]. The encoding circuit $U_{\phi(x)}$ can also be applied more than once and/or interleaved with the model circuit described later.

- *Model circuit.* A short-depth parameterized quantum circuit $W(\theta)$ is applied on the obtained quantum state with layers that are parameterized by the rotational angles for the gates that need to be optimized during training (see Fig. 6). The optimization is performed over a cost function.
- *Measurement and preprocessing.* The outcome of the measurement results in a bit string $z \in \{0,1\}^n$ that is mapped to a label. This circuit is rerun multiple times and sampled to estimate the probability of observation z, which can be obtained as

$$\langle\Phi(x)|W^{\dagger}(\theta)M_y W(\theta)|\Phi(x)\rangle,$$

which is calculated for each of the different classes y using the measurement operator M_y.

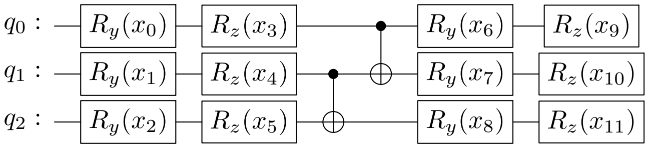


FIG. 5. The feature map ($\Phi(x)$) depicted utilizes three qubits to encode 12 parameters of the data point x. It applies rotation-Y and rotation-Z operations to the feature values while employing CNOT gates to establish entanglement.

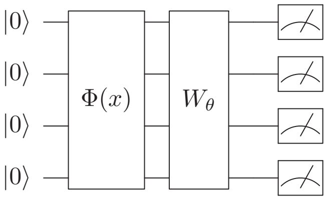


FIG. 6. Variational quantum classifier. The state preparation $\Phi(x)$ is followed by $W(\theta)$, which is the parameterized circuit with parameters θ, and is then followed by measurement in the Z basis.

At the aforementioned quantum feature map and the model circuit, the CZ and CNOT gates (along with Hadamard gate) are commonly used to create entanglement (Fig. 8). A common strategy to optimize the subcircuit for entangling qubits is to entangle adjacent qubits, namely, we first entangle the $2i$th qubit with the $(2i+1)$th qubit, and then after this, we only entangle the $(2i+1)$th qubit with the $(2i+2)$th qubit for $i = 0, \ldots, n$. By doing so, we can parallelize the entanglement operation and reduce the execution dependence [181]. The circuit for this is shown later in Fig. 8(a). Based on the depth of the circuit chosen, we can repeat the quantum feature map with entangling subcircuit, or the model circuit with entangling subcircuit.

a. Ansatz. The choice of the ansatz also plays a pivotal role as the parameters θ of the circuit represented by W are

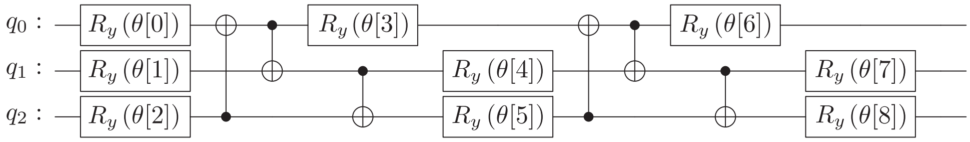


FIG. 7. The representation illustrates $W(\theta)$ as a hardware-efficient ansatz for a three-qubit system, employing a parameterized quantum circuit with nine parameters.

optimized during the training (see Fig. 7). For example, the experiments in Ref. [76] showed better performance for the weighted ansatz (Sec. V A 4 a) in comparison to the Fourier ansatz (inspired from Ref. [182]), which introduces linear and logarithmic dependences to the same gate without using tunable weights. The authors speculate that using weights allows better representability especially for smaller layers. The layered ansatz is another common technique that comprises layers such that each of the layers is composed of a set of entangling gates preceded by two alternating single-qubit rotation gates. The number of layers is a hyperparameter.

b. Modified layerwise learning. As the name suggests, this strategy involves training the circuit layer by layer such that only a small set of parameters are optimized in a single update [183]. Initially, only a small circuit with a few starting layers is chosen such that all parameters are set to 0. This circuit is optimized by running it for a few epochs. The parameters are now frozen and a new set of layers is added. Now, the new layers' parameters are optimized with the previous layers' frozen parameters until no more improvement is obtained in the cost function or until the desired depth is reached. Then, the circuit depth is fixed and a larger set of parameters is trained again. This strategy can help avoid the barren plateau due to the small number of layers and also maintains a favorable signal-to-noise ratio [64,183].

c. Optimizers. The work in Ref. [184] demonstrated that gradient-free optimizers, simultaneous perturbation stochastic approximation (SPSA) and Powell's method, and gradient-based optimizers, AMSGrad and BFGS, performed the best in a noisy simulation, and appeared to be less affected by noise than the rest of the methods. SPSA appeared to be the best-performing method while COBYLA, Nelder-Mead, and conjugate-gradient methods were the most heavily affected by noise even with the slightest noise levels.

In Ref. [185], the authors explored hardware-efficient ways to optimize the ansatz using algorithms based on tensor methods. One of the introduced methods, called coordinatewise optimization [186–189], does not require the use of gradient or Hessian computations. Additionally, the authors of Ref. [190] presented a non-gradient-based variational algorithm by introducing an unconventional hybrid quantum-classical algorithm that utilizes the quantum part only once after optimizing the circuits entirely on a classical computer. This is in contrast to traditional methods that involve multiple uses of the quantum computer during circuit optimization.

More recently, the work in Ref. [191] proposed a novel approach that combines the approximated gradient from SPSA with classical optimizers. This hybrid approach outperformed standard SPSA and the parameter-shift rule in regression tasks, demonstrating enhanced convergence rate and error reduction, especially when considering noise.

Even the choice of batch size for training affects the convergence rate. In principle, quantum computing allows one to encode a batch of training inputs into a quantum state in superposition and feed it into the classifier, which can be used to extract gradients for the updates from the quantum device. However, this would extend the time complexity of the state preparation routine for general cases, and even worse for more sophisticated feature maps. Single-batch stochastic gradient descent, where only one randomly sampled training input is considered in each iteration, can have favorable convergence properties, especially in cases where a lot of data are available [192]. However in Ref. [80], training using single data per update led to slow convergence with volatile validation loss per epoch, which was avoided by increasing the batch size to 64.

Some of the major limitations associated with classical optimizers are repeated measurements and the complexity of gradient calculation [193]. Classical optimizers often require repeated measurement of the outputs of a quantum circuit and feeding them into the classical computer. This process can lead to slower convergence rates for the optimization algorithm. The complexity of calculating gradients can impact the convergence of the optimization algorithm, particularly as the feature size (d) of the input increases. For instance, gradient-based methods like gradient descent have a complexity of $O(d)$ [194], which can become a scalability bottleneck. To address these limitations, researchers have proposed quantum gradient methods [195–197] in the recent past as potential alternatives. These methods aim to leverage the benefits of quantum computation to overcome the challenges associated with classical optimization. However, their practical implementation still faces challenges related to applicability and complexity.

d. Parameter-shift rule. To optimize the objective, it is useful to have access to exact gradients of quantum circuits with respect to gate parameters. The parameter update requires computing $\nabla L(\theta)$, which in turns requires computing the gradient of the quantum circuit output f due to the chain rule, since the loss function is a function of the output of the quantum circuits. The gradient of the quantum circuit output is calculated using the parameter-shift rules [198,199] by varying the value of the gate parameters θ slightly.

For the gates used in angle encoding, the parameter shift can be applied as

$$\frac{\partial f}{\partial \theta} = \frac{1}{2}\left[f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right)\right].$$

In other cases, different strategies can be applied, as discussed in Ref. [199]. When the ansatz consists of single-qubit rotation gates $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$, as in Fig. 2, the loss function can be optimized with gradient-free optimizers using coordinate descent [186,200]. While the gradient-based optimizers can be parallelized [201], the gradient-free coordinate descent methods are sequential but have been shown to converge to local optima faster [201], and can be combined with *data parallelism* to run several small quantum circuits on a device with a larger number of qubits as shown in Ref. [111]. The generalization of gradient-free sequential single-qubit gate optimizers is derived in Ref. [202,203]. Nevertheless, within the existing training framework for quantum neural networks (QNNs), it is necessary to compute gradients with respect to the objective function directly on the quantum device. However, this computation faces significant scalability challenges and is susceptible to hardware limitations and sampling noise inherent in the current generation of quantum hardware [204].

In a recent study, the authors of Ref. [205] presented an alternative training algorithm that circumvents the need for gradient information. They introduced a novel metaoptimization algorithm, which involves training a metaoptimizer network to generate optimal parameters for the quantum circuit. These parameters are carefully chosen to minimize the objective function without relying on traditional gradient-based approaches.

e. Quantum natural gradient. The geometry of the parameter space plays a huge role in the efficient optimization of the VQC parameters [206]. In Ref. [65], the authors expect that a smaller network structure of the VQC can lead to significant advantage, as it allows using a computationally more expensive optimization algorithm resulting in a faster learning rate. This is also advantageous when the training data are limited.

In vanilla gradient descent, the loss function $L(\theta)$ is minimized in the l^2 vector space by updating the network parameter $\theta^{(t)}$ at time t to $\theta^{(t+1)}$ in the direction of the steepest slope as

$$\theta^{(t+1)} = \theta^{(t)} - \eta\nabla L(\theta).$$

Since each of the model parameters is updated by the same Euclidean distance, there is a possibility of getting stuck in a local minimum since the value of $f(\theta)$ varies at different rates with respect to each parameter. This is tackled in natural gradient descent where the parameter space corresponds to Riemannian geometry, which is defined by the Fisher information matrix [207,208] and is invariant under reparameterization. The parameters are updated as

$$\theta^{(t+1)} = \theta^{(t)} - \eta F^{-1}\nabla L(\theta),$$

where F is the Fisher information index. The calculation of F^{-1} in general is computationally expensive. However, this leads to faster convergence, and can help avoid getting stuck in local minima [209].

For VQC parameter optimization, it has been shown that using the standard Euclidean geometry is suboptimal [210]. Quantum gradient descent is the quantum version of natural gradient descent [211] which uses the Fubini-Study metric g [212,213]. This Fubini-Study metric tensor is an invariant metric tensor unique to the space of quantum states and exploits the geometric structure of the VQC's parameter space. The parameters are updated as

$$\theta^{(t+1)} = \theta^{(t)} - \eta g^{+}\nabla L(\theta),$$

where g^{+} is the pseudoinverse of the Fubini-Study metric g. Faster convergence has been observed for quantum gradient descent compared to the vanilla gradient descent with a similar number of trainable parameters [65].

f. Quantum natural SPSA. The large computational costs associated with calculating the quantum Fisher information, which scales quadratically in the number of ansatz parameters, limits the advantage of using quantum gradient descent over standard gradients. To counter this, a new approach is introduced in Ref. [214], called quantum natural-simultaneous perturbation stochastic approximation (QN-SPSA), which inherits the fast convergence and robustness of quantum natural gradient with respect to the initial parameters, while having the computational cost benefits of SPSA [215].

Additionally, it is worth mentioning some recent works, such as the pure quantum gradient descent algorithm, which was proposed in a recent study [193]. This innovative quantum-based method for gradient calculation claims to provide a theoretical computational complexity of $O(1)$, in contrast to the $O(d)$ complexity of the classical algorithm [194].

4. Quantum principal component analysis

Principal component analysis has been used for the optimal low-rank approximation of a matrix through spectral decomposition by setting a threshold on the eigenvalues. By doing so, we only retain the principal components of the spectral decomposition while discarding those with smaller eigenvalues. However, when the size of the matrix is large, the computational costs increase, which is why we look at quantum algorithms.

The implementation of QPCA in Ref. [216] helps construct the eigenvectors and eigenvalues of the unknown density matrix, thereby discovering their properties. The authors assume that the matrix can be represented by a quantum state, i.e., it is a non-negative matrix with trace equal to 1, which covers a wide range of interesting cases. It uses multiple copies of an unknown density matrix to construct the eigenvectors corresponding to the large eigenvalues of the state (the principal components) in time $O(\log N)$, where N is the dimension of the Hilbert space, resulting in an exponential speedup over existing algorithms. They provide novel methods of state discrimination and cluster assignment.

5. Quantum orthogonal neural networks

Orthogonal neural networks are neural networks with orthogonal trained weight matrices which provide the advantage of avoiding vanishing gradients and improved accuracies [217]. The parameterized quantum circuit for implementing the orthogonal neural networks was first introduced in Ref. [157] using unary amplitude encoding and a pyramidal structure using only RBS gates. The orthogonality of the weight matrix is preserved by performing gradient descent on the parameters of the quantum circuit. This works because a quantum circuit with real-valued unitary gates is an orthogonal matrix, hence the gradient descent is equivalent to updating the weight matrix. Another feature of the circuit is one-to-one mapping between the parameters of the orthogonal matrix and the quantum gates of the circuit. The circuit architecture benefits from linear circuit depth and error mitigation due to unary encoding along with nearest-neighbor connectivity due to the distribution of the RBS gates. In Refs. [89,157], the results show linear scaling of the training runtime with respect to the number of parameters.

6. Quantum generative adversarial networks

The primary goal of a classical generative adversarial network (GAN) [218] is to generate data by studying a collection of training examples and learning the underlying probability distribution. It typically involves an iterative adversarial training procedure between two neural networks, the discriminator and the generator model. The generator creates fake data with the goal of generating data as close as possible to the real training dataset while the discriminator tries to separate these fake data from the real data.

The quantum variant of GAN (i.e., quantum generative adversarial network) was proposed independently in Ref. [219,220], where a QNN is used as the discriminator or generator or both. In Ref. [77], faster convergence was noted for a classical discriminator in comparison to other architectures. For more details refer to Ref. [221].

The research described in Ref. [222] showcased the first proof-of-concept experimental demonstration of a QGAN on a superconducting processor. The authors successfully trained a quantum-state generator through adversarial learning to replicate quantum data with 98.8% fidelity. However, the scalability of QGANs to noisy intermediate-scale quantum devices was first shown in Ref. [133] by implementing the QGAN using a programmable superconducting processor.

a. Quantum adversarial learning. Adversarial machine learning involves assessing the vulnerabilities of machine learning in adversarial settings and consequently implementing techniques to make the models more robust to such manipulations. In the quantum setting, Ref. [223] shows that a quantum classifier that performs with nearly state-of-the-art accuracy can be deceived by adding unnoticeable perturbations to the original samples. For more information and discussions on the latest advancements and key challenges in the field of quantum adversarial machine learning, we refer the reader to Ref. [224].

7. Tensor networks

Tensor networks (TNs) are a popular method in the field of quantum many-body problems due to their ability to represent many-body localized systems and are already known for their performance in the classical setting for supervised and unsupervised learning tasks. TNs can represent both quantum states and circuits [225–227] using VQCs with rules described in Ref. [228]. They can also simulate strongly entangled quantum systems [229–231]. Depending on the architecture, the number of physical qubits scales only logarithmically with, or independently of, the input or output data sizes, which can be implemented on small, near-term quantum devices using fewer physical qubits. The work in Ref. [69] shows that classical TNs require exponentially more trainable parameters and higher Hilbert-space mapping to perform on par with the quantum counterparts, which makes them vulnerable to a highly flat loss landscape. The work in Ref. [232] explores the trainability of randomly initialized quantum tensor networks with a focus on the different architectures. The conjecture suggests that classical gradient computation for quantum tensor networks could be more efficient than their quantum counterparts. A review can be found in Ref. [233].

8. Quantum autoencoder

The task of a classical autoencoder is to obtain a low-level representation of a given input such that the original data can be recovered. This has applications in dimensionality reduction and generative data modeling. The quantum version of the classical encoder was first implemented in Ref. [48] where an ansatz is trained to obtain a compressed version of an ensemble of pure quantum states. Different variants are explored in Refs. [234–236]. The learning task involves finding unitaries that preserve the quantum information of the input through a smaller intermediate latent space. The PQC initially encodes the input state into an intermediate latent space. Following this, the decoder acts with the goal of being able to reconstruct the input. A cost function is used to estimate the fidelity (distance) between the input and output states. Recently, quantum autoencoders have found application in quantum error correction [237], thereby expanding the possibilities for further research and development in the field.

VI. APPLICATIONS

The performance metric is usually measured using AUC-ROC, which stands for “area under the receiver operating characteristic curve” [238,239]. The ROC curve is a commonly used graph that summarizes the performance of a classifier over all possible probability thresholds. The AUC-ROC provides intuition about the capability of the model to distinguish accurately between true positives and false positives [true positive rate (TPR) on the y axis and false positive rate (FPR) on the x axis]. The score varies from 0 to 1, where higher score implies better distinction or performance and score 0.5 corresponds to random guessing.

A. High-energy physics

Most of the studies focus on obtaining better performance with limited data [240]. Among recent works, VQC has been used widely for HEP-based applications with data agnostic techniques for feature encoding like single-qubit rotation gate or *ZZ* gate [240]. However, these methods are not suitable for HEP, as they end up incurring large overhead on the number of qubits or gates for multidimensional data. Here we review the applications of variational quantum circuit and quantum support vector machine techniques in high-energy physics.

1. Classification

A prominent use case under this category is event classification, which involves discriminating signal events from the background events in the context of the Standard Model of particle physics [64,66].

In Ref. [66], the data are encoded using first-order encoding and a variational part based on a simplified version of Ref. [47] with depth 1. A combination of three variables is determined using a deep neural network with AUC-ROC and run on an IBM Quantum device [241] with three qubits. The cross-entropy cost function is optimized using COBYLA. Results showed a higher cost function with more fluctuations for the real device compared to the simulator, but both showed consistent AUC (around 0.80) within the standard deviation. Additionally, second-order encoding was employed, but no improvement was observed on a real quantum computer. This may be attributed to the 60% increase in single- and two-qubit gates when transitioning from FOE to SOE, resulting in increased hardware noise due to gate errors.

In Ref. [64], an improvement over this method is shown using data reuploading and modified layerwise learning with only one qubit and five layers. However, training is performed on the PennyLane simulator, optimizing the MSE cost function with the Adam optimizer [243]. Inference tests on Rigetti’s 32-qubit superconducting quantum processor obtained an AUC of 0.830, surpassing that of Ref. [66] while using fewer qubits. Training and testing AUC using 2000 samples demonstrate that data reuploading generalizes well without overfitting or underfitting.

To compare the performance of variational circuit-based and kernel-based methods on the same dataset, we refer to Refs. [67,68], which use VQC and QSVM, respectively. Both works employ PCA [243,244] for data preprocessing, matching the number of encoded variables with the available qubits of the IBM Quantum device [241], followed by angle encoding.

The ansatz in Ref. [67] uses parallelized entangling CZ gates with linear qubit connectivity [Fig. 8(a)]. For training on real hardware, the feature map and variational circuit depth were set to 1. SPSA was used for optimization, and the results were benchmarked against classical SVM [55] and binary decision tree [245]. The simulator performance was comparable to classical methods (AUC around 0.82). To minimize readout errors, only half the qubits were observed after pairing them with CZ gates. The performance on real quantum hardware was similar (AUC > 0.80) to that on the simulator. However, the authors note the long training time on quantum hardware (200 h) for 500 training iterations on 100 events.

For the QSVM-based approach in Ref. [68], a parallel entangling circuit was constructed using 15 qubits of the IBM Quantum device, similar to Ref. [67], to obtain a short-depth circuit for execution on real quantum hardware. To reduce statistical uncertainties, 8192 measurement shots were performed for each kernel entry. The hardware performance approached that of the noiseless simulator for small training samples of size 100 (average AUC 0.831).

Testing the model on an IBM Quantum device showed good performance, with an AUC of approximately 0.78. The authors attribute the faster learning rate, despite the computationally expensive optimization algorithm, to the small structure of the VQC.

In Ref. [128], the authors note that data obtained from quantum-enhanced experiments can achieve quantum advantage for learning tasks of a physical state and a physical process from a perspective of sampling complexity. Quantum-enhanced experiments consist of quantum sensors, quantum memory, and quantum computers. In quantum-enhanced experiments, quantum information is directly stored in quantum memory, while classical experiments require measurements to store classical data in classical memory. Quantum-enhanced experiments preserve the quantumness of quantum data until performing entanglement measurements on pairs of copies of data in quantum memory. It is expected that research utilizing the power of quantum data will become increasingly active in the future.

2. Regression

a. Simulation. The use of different variants of QGAN architectures in HEP can be seen in Refs. [77,78] for simulation. For example, in Ref. [78], the proposed QGAN contains a classical discriminator and two parameterized quantum circuit generators for generating images. The performance was measured via relative entropy and individual relative entropy. The model was trained using a simulator, while the inference results of the pretrained model on superconducting chips and ion-trap machines showed low standard deviation and error rates indicating the feasibility of dual-PQC training for superconducting chips. However, the authors note the vulnerability of the training process to fall into mode collapse [218] where the model reproduces only a low variety of samples. They also suggest techniques such as increasing the training set size and adding an additional term to the loss function as possible solutions to ameliorate the problem.

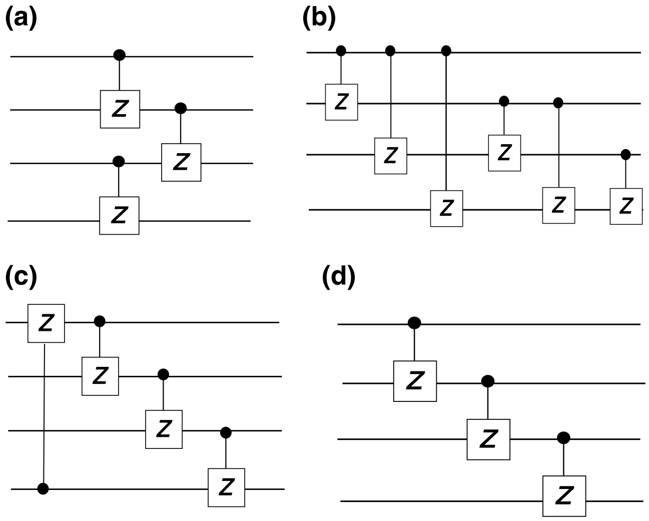


FIG. 8. Different types of entanglement: (a) parallel, (b) all-to-all, (c) circular, and (d) linear.

In contrast, the QGAN in Ref. [77], named style-QGAN, was implemented using a QNN generator and a classical NN discriminator. The data were encoded using angle encoding, and the cross-entropy loss function was optimized for both using Adadelta [246]. While earlier implementations of QGAN provided the prior noise distribution to the generator via the first input gates, the work in Ref. [77] embeds it on every layer of single qubit and entangling gate in the circuit. The results showed an improvement over the state of the art with shallow circuits on both the three-qubit superconducting and ion-trapped architectures, implying potential hardware-independent viability. Additionally, both quantum hardware variants are able to capture the correlations even on small sample set.

B. Healthcare

A few applications of quantum machine learning in healthcare include healthcare diagnostics and treatment, cancer detection, prediction of different stages of diabetes, and even the security of sensitive information such as healthcare data.

Given the sensitivity of these applications, the cost of any incorrect predictions may have huge negative consequences and hence requires utmost carefulness. In this regard, binary classification of magnetic resonance images using a VQC is performed in Ref. [90] to check the vulnerability of quantum learning systems in healthcare diagnostics. To prepare highly entangled multiqubit quantum state, interleaved block encoding [161,247,248] was used on 10 qubits. The variational parameters are fixed for adversarial perturbations, and the results show that the originals differ from the adversarial images by a small amount of perturbations. The results show that the quantum classifier predicts the legitimate states accurately while mispredicting all (half) of the adversarial examples, highlighting the vulnerability aspect. Additionally, experiments were performed on quantum data as well with the quantum classifier, reaching perfect accuracy in about 30 epochs on both training and test datasets.

In Ref. [89], QNN and quantum orthogonal neural networks are used for healthcare image classification on the RetinaMNIST dataset and PneumoniaMNIST [249]. The images were preprocessed using PCA and followed by unary amplitude encoding. A series of experiments was performed on the real hardware using five and nine qubits. The results show comparable accuracies for the majority of the classification experiments performed on real quantum hardware to those of their classical counterparts. However, the hardware limitations come into the picture for more difficult tasks. Additionally, circuit optimization based on the hardware and translation of the RBS gates into native hardware gates was performed to reduce the overall gate count. The results show better performance for five-qubit results in contrast to the nine-qubit experiments, where the hardware performance seems to diverge from the simulator performance. The authors note the unstable performance of the quantum hardware due to the randomness in the training or inference making it incapable of performing healthcare image classification on par with the classical models.

To analyze the advantage of using quantum machine learning in terms of sample complexity, the authors of Ref. [92] conducted experiments using QSVM on a small dataset of size 200–300 training samples with kernel techniques for prediction of six-month persistence of rheumatoid arthritis. The experiments were conducted on different configurations of features and data sizes to identify cases where quantum kernels could provide advantage. A new metric, empirical quantum advantage, is proposed to quantitatively estimate the accuracy of the model performance as a function of the number of features and sample size. The estimation of the custom kernel turns out to be the most computationally expensive task. The authors claim to be the first to use geometric difference to analyze the relative separation between classical and quantum features. They note that kernels are noisy and that quantum advantage expressed in terms of the generalization error vanishes with large datasets, fewer measurements, and increased system noise.

C. Finance

Some applications of ML operations applicable to finance include regression for asset pricing [250–252], classification for portfolio optimization [253–258], clustering for portfolio risk analysis and stock selection [259–263], generative modeling for market regime identification [257,264–266], feature extraction for fraud detection [267–273], reinforcement learning for algorithmic trading [274–277], and natural language processing for risk assessment [278,279], financial forecasting [280–285], and accounting and auditing [279,286, 287].

In a similar vein, QML has been used for different applications, such as feature selection for fraud detection in Ref. [86], where a PQC was trained on a subset of good features selected based on their performance using a predefined metric. The use of QN-SPSA showed good convergence for training on 20 qubits with potential for deeper circuits. The results on hardware were comparable to state-of-the-art classical methods in certain aspects, while in others it showed the potential to find better subsets. The authors of Ref. [86] note that a model run on an IBM Quantum device was able to outperform traditional methods without using error mitigation.

A recent study [288] introduced an innovative method for quantum reservoir computing (QRC) and applied it to the foreign exchange market. The approach accurately depicted the stochastic evolution of exchange rates compared to classical methods. In QRC, input signals are transformed into a complex quantum superposition in a high-dimensional space, after which the transformed signals are connected to the desired output through a basic neural network. The study highlights the learning performance and potential of QRC to be run on near-term quantum devices.

Another application of QML was explored in Ref. [85] to reduce the number of noisy factors for pricing interest-rate financial derivatives using a QPCA. The experiments were performed on a five-qubit IBM Quantum device for 2×2 and 3×3 cross-correlation matrices based on historical data for two and three time-maturing forward rates. However, this method showed difficulty in scaling to larger datasets.

We direct the interested reader to Ref. [289] for a more comprehensive summary of the state of the art of quantum computing for financial applications.

VII. LIMITATIONS

The current quantum hardware is susceptible to noise resulting in a very low qubit coherence time of the order of a few hundred microseconds. Common sources of noise include: (1) crosstalk due to simultaneous gate execution in quantum algorithms that allow parallel operations; (2) quantum decoherence; (3) single-qubit rotation and two-qubit gate error rate due to imperfect implementation; and (4) shot noise from measurements on quantum states. Additional limitations due to qubit count and gate fidelity prevent the use of quantum error correction. The use of VQCs provides a framework to enable practical applications of noisy quantum hardware. Here, we briefly look at some of the limitations associated with the current QML approaches.

A. Hardware limitations

The common causes of error rates are the state preparation and measurement error rate (SPAM) and gate errors. The SPAM measures the correctness of the initial calibration settings and the final readout measurement and is indispensable for scaling to hundreds or thousands of qubits. A general strategy to counter the noise in quantum hardware is to increase the number of measurements to help reduce the generalization error [180]. However, this may also be counterproductive due to the readout error during measurement. For example, the prediction accuracy dropped on increasing the number of shots from 500 to 1000 in Ref. [120]. The authors note that the experiment was already dominated by systematic noise, which was prone to change every time the system was calibrated, indicating the variability in the calibration of the system. Other options include using shot-frugal optimizers [290–293], which use a stochastic gradient-descent-based approach while adapting the number of shots (or measurements) needed at each iteration.

A popular noise mitigation technique is zero-noise extrapolation to first order for gate-error mitigation described in Ref. [174,294], which can be implemented in software without requiring any prior knowledge of the quantum computer noise parameters. Factors such as qubit lifetime and coherence time are affected by decoherence. Decoherence, characterized by uncontrolled interactions between a quantum system and its environment, poses a significant challenge in quantum computing, resulting in the loss of quantum behavior within the quantum processor, nullifying any potential advantages offered by quantum algorithms. The decoherence time limitation significantly restricts the number of operations that can be performed in a quantum algorithm. Additionally, the development of high-fidelity qubits poses another critical hardware challenge. To tackle these issues, an effective approach is to treat qubits as part of an open environment and leverage classical simulation software packages during the design phase.

Superconducting QPUs have a coherence time of around 100 μs while certain trapped ions have extended that to 50 s. The gate speed along with decoherence need to make sure that the gates are applied before the system decoheres. Superconducting and photonic systems generally have the fastest gate speeds. The qubit connectivity, which is the general layout of the qubits, dictates the interaction between a given qubit and its neighbors. Because of limited connectivity, SWAP gates can be inserted but can result in additional overhead and subsequent error rates. While some devices offer all-to-all connectivity, long-range gates are generally more noisy. The delay between submitting a circuit to the Cloud and receiving a result, without clarity on the calibration timings, can lead to significant statistical errors [295], as it is unclear as to how these errors influence circuit performance between runs on all systems. The lack of information on aspects such as qubit assignment, compiler and/or transpiler methods, component drift rate, and time since last calibration also affect the analysis, as noted in Ref. [295].

B. Long running time

Often, studies require a large number of samples and qubits (20 qubits or more), which necessitates a large amount of computational power for quantum computer simulations. Long running times have been noted in Refs. [67–69,92,296] on current quantum hardware, even when using small data samples, likely due to the initialization, queuing, execution, and measurement time in the current quantum hardware. For example, the study in Ref. [67] took around 200 h to run 500 training iterations on 100 events on quantum hardware. This poses a serious limitation for real-world applications such as HEP, which generally require large training data. In terms of the model performance, using small sample size often leads to significant variance and poor performance. Furthermore, the limited access to QPU resources makes it infeasible to conduct validation on multiple sets [92].

In Ref. [297], the authors propose measuring the speed using circuit layer operations per second (CLOPS) by considering the interaction between classical and quantum computing. The CLOPS benchmark consists of 100 parameterized templated circuits and takes into account various factors such as data transfer, runtime compilation, latencies, gate times, measurements, qubit reset time, delays, parameter updates, and result processing. However, CLOPS focuses mainly on the quantum computing aspect and considers classical computation as an auxiliary to quantum computing. Furthermore, factors such as qubit quality and gate operations are not captured in the metric. Experimental results indicate that the execution time of quantum circuits constitutes a small proportion (less than 1%) of the total execution time [297,298].

Another proposed solution to improve the training time is quantum federated learning (QFL), which uses distributed training across several quantum computers. Federated learning consists of several clients or local nodes learning on their own data and a central node to aggregate the models collected from those local nodes. A framework for federated training was presented in Ref. [299] using hybrid quantum-classical machine learning models. Their simulation results show faster convergence compared to the non-federated training and the same level of trained model accuracies. Other works include Ref. [300] where the authors introduce slimmable QFL (SlimQFL), a dynamic QFL framework which has been shown to achieve higher classification accuracy than the standard QFL.

In contrast, ensemble learning involves the combination of multiple individual models, referred to as base models or weak learners, to create a more accurate and robust predictive model. These base models can be of the same type or different types, and their predictions are aggregated using methods such as voting, averaging, or weighted averaging. Ensemble learning aims to improve overall performance and accuracy by leveraging the strengths of multiple models.

On the other hand, federated learning is distinct from ensemble learning in that it enables collaborative training across distributed entities without sharing raw data, ensuring privacy and security. While ensemble learning focuses on model aggregation, federated learning emphasizes the distributed nature of training.

Some works that explore ensemble learning in the context of quantum machine learning include Refs. [301–306].

C. Inefficient data loader

Being able to load classical data as quantum states efficiently is a bottleneck that has often been sidelined in works that discuss speedup using QML algorithms. Given a classical data point, the job of a data loader is to read the data once and output a PQC that prepares an appropriate quantum representation. The encoding part of input data generally consumes a significant portion of the coherence time, often leaving little time for the actual algorithm to process the data [240]. Several proposals for more efficient data loading have been made in this regard. For example, the work in Ref. [120] tries to tackle this by describing ways to load a classical data point with logarithmic-depth quantum circuits while using the same number of qubits as the feature dimension. Another technique is described in Ref. [89], where a shallow parallel data loader is implemented for d-dimensional data points using d qubits, $d - 1$ RBS gates, and circuits of depth only $\log d$. However, the viability of this approach is limited by connectivity requirements beyond those supported by the hardware.

The idea of quantum random-access memory (QRAM) [144,308–310] has been proposed for the long-term storage of the state of quantum registers and can be considered to be a specific hardware device that can access classical data in superposition natively, thus having the ability to create quantum states in logarithmic time. Despite challenges in implementation, alternative proposals with similar functionality have emerged. In Ref. [311], a circuit with $O(d)$ qubits and $O(d)$ depth was described to perform the bucket brigade architecture with proven robustness to a certain level of noise.

D. Barren plateau

Flat optimization landscapes, where the gradient variance diminishes exponentially with the number of qubits, are commonly encountered in variational quantum algorithms. Similar to classical machine learning, quantum loss landscapes are susceptible to numerous local minima. Recent studies [312,313] have demonstrated that overparameterization can help alleviate barren plateaus by utilizing more parameters than necessary for a given problem. This allows the quantum neural network to explore all relevant directions in the state space. However, factors such as ansatz architecture [314,315], cost function [316], and parameter initialization contribute to encountering barren plateaus [86]. For instance, a highly expressive ansatz [317] or an ansatz with exhaustive entanglement [318–320] can result in exponentially flat landscapes as the number of qubits increases [321].

Addressing these challenges involves employing adaptive initialization methods or informed parameter initialization, problem-dependent ansatz design, circuit pruning, utilizing density matrices and random features for distribution estimation, concurrent optimization of parameters and rotation generators, as well as the incorporation of global optimization techniques like genetic algorithms for enhancing gate or structural optimization. The work in Ref. [322] focuses on supervised learning with quantum feature maps optimized using a genetic algorithm which designs feature map circuits with high accuracy, generalization, and minimal size, demonstrated through diverse benchmarks, suggesting potential for hybrid quantum-inspired machine learning strategies.

Limiting entanglement in the ansatz [323] can help overcome exhaustive entanglement-induced barren plateaus [318–320]. The choice of observables to define the loss function also influences the presence of barren plateaus. Using global observables that require measuring all n qubits simultaneously [316] can lead to barren plateaus, whereas employing local observables that compare quantum states at the single-qubit level [316,324] can avoid this issue. Recent research [325] has shown that local cost functions encounter barren plateaus when learning random unitary properties. Furthermore, local noise in the hardware [326] can affect the optimization process. Techniques such as error mitigation [174,327] can help reduce the impact of local noise. Different ansatz designs, including variable ansatz [178,328], Hamiltonian variational ansatz [329–331], or hardware-efficient ansatz, which aim to reduce gate overhead [332,333], can be utilized and optimized using quantum-specific optimizers [291,334] for training.

Gradient-based methods are generally preferred for large parameter spaces [198]. However, gradient-free methods have also been utilized for optimization, as shown in Ref. [335], where Nelder-Mead was employed for quantum variational eigensolver optimization. However, scaling results in Ref. [321] indicate that deep versions of randomly initialized hardware-efficient ansatzes suffer from exponentially vanishing gradients. As an alternative, one can opt for barren-plateaus-immune ansatzes [316,336–338] instead of hardware-efficient ansatzes. Additionally, using shallow circuits with local cost functions [316,328] can help mitigate the presence of barren plateaus. In Ref. [316], an alternating layered ansatz is proposed, which was later proven to have sufficient expressibility [339]. The results in Ref. [316] demonstrate that the barren plateau phenomenon extends to variational quantum algorithms with randomly initialized shallow alternating layered ansatzes and establish a relationship between locality and trainability of VQCs. They also show that, despite using a shallow circuit, defining a cost function using global observables leads to exponentially vanishing gradients. Among other techniques, the initialization strategy using identity blocks described in Ref. [340] and layerwise training can be employed to mitigate barren plateaus.

The researchers in Ref. [341] developed a scalable method to calculate the gradient and its variance by proving that randomly initialized circuits can be exactly mapped to a set of simpler circuits that can be efficiently simulated on a classical computer.

The study in Ref. [342] highlights the potential of geometric quantum machine learning for addressing barren plateaus and overfitting where quantum models are customized to reflect image symmetry. The results show an improvement in accuracy compared to generic models while using amplitude encoding. The issue of overfitting in VQCs has also been addressed in Ref. [301], where the authors implemented boosted ensembles of quantum support vector machines on HEP datasets. The ensemble classifier was found to double the efficiency of a single QSVM, which the authors claim is highly susceptible to overfitting.

VIII. OPEN QUESTIONS

The key objective in the field of quantum machine learning is to demonstrate quantum advantage, surpassing classical methods in data science applications in terms of either sample complexity or time complexity. This requires a flexible and exploratory approach to identify the areas where QML can have the greatest impact. Although there are claims of polynomial and exponential speedups in QML, empirical evidence establishing a clear advantage over classical algorithms is still limited. Furthermore, providing a robust theoretical foundation for quantum advantage poses significant challenges in the field. It remains unclear whether the observed performance improvements are solely attributed to careful hyperparameter selection, benchmarks, and comparisons, or if there is a fundamental structural advantage [63]. It can be observed that QML as a field is moving toward becoming an empirical science. The theoretical aspect of proving concepts is anticipated to be challenging, and the emphasis is increasingly placed on practical demonstrations. This trend is particularly notable as the number of qubits and circuit depth surpasses 100×100. There is a possibility that an efficient classical algorithm exists for a given learning problem that can achieve comparable results to quantum learning algorithms. This is exemplified in Ref. [47], where the variational circuits can be replaced by a classical support vector machine if the encoding is classically tractable.

Furthermore, due to finite sampling noise, none of the heuristic quantum learning algorithms have been proven to solve a classically hard learning problem [44]. These inherent limitations imply that the current benefits of quantum algorithms can only be realized under certain circumstances. Specifically, only a few variational quantum-based algorithms have shown an apparent advantage in a constrained situation [343]. Recently, the authors in Ref. [134] investigated the impact of finite sampling noise and subsequently introduced a technique called variance regularization based on the expressivity of QNNs to reduce the variance of the output.

To shed light on the current state of quantum machine learning, several research directions and areas of investigation are now identified.

A. Establishing standardized benchmarks

In order to effectively evaluate the superiority of QML algorithms compared to classical ones, it is crucial to establish standardized benchmarks. Currently, standard classical data benchmarks such as MNIST, Iris, etc. are used (as shown earlier in Table IV). The lack of standardized quantum datasets highlights the need for easily preparable quantum states to serve as benchmarks for evaluating QML models [139]. We note that open-source software and standard datasets for benchmarking standard QML models for binary classification tasks addressing the claims of QML superiority over its classical counterparts are provided in Ref. [344]. Furthermore, a proposal to develop practically meaningful quantum datasets using quantum circuits or states where quantum methods are expected to excel compared to classical methods has been addressed in Ref. [129].

B. Quantum data preparation

While achieving a quantum advantage with classical data is challenging, QML models utilizing quantum data show more promise. Finding the most optimal encoding technique for a given dataset is another crucial challenge that needs to be addressed. These embedding techniques necessitate having features that are classically hard to simulate with practical usefulness. Identifying datasets that can take advantage of quantum computing for computing kernels is an important avenue of research. Some recent work [345] shows that, when the classical data follow certain patterns, the quantum states can be represented using an efficient approximation leading to quantum circuits where the number of gates grows linearly with the number of qubits rather than exponentially, making them more feasible. Currently, there is a lack of efficient QRAM capable of encoding and reliably storing information as a quantum state. This presents a significant hardware challenge in quantum computing.

C. Error mitigation and quantum error correction

Error mitigation and error correction are crucial for the long-term viability of fault-tolerant quantum computers. However, the implementation of quantum error correction introduces overhead that can reduce the speedup of quantum computations [139]. Therefore, finding efficient quantum error-correcting codes and developing methods to generate ground states using QML models are important areas of research.

Recently, studies have investigated the potential of QML, particularly quantum autoencoders (QAEs), for error correction in quantum memory [237]. QAEs offer promise in autonomously correcting errors and extending logical qubit lifetimes, potentially streamlining error-correction processes. Future research should explore the fault tolerance of QAEs and their integration into broader error-correction frameworks. Various error-mitigation techniques, including ensemble-learning approaches that combine multiple variational quantum circuits (VQCs), are employed to enhance the precision of classifiers for both classical and quantum datasets. One such study [346] proposes two ensemble-learning error-mitigation methods for VQCs: bootstrap aggregating and adaptive boosting. These methods can be applied to classification, kernel learning, and regression, and can even be extended to QSVM. Importantly, their ensemble-learning VQCs are designed to be compatible with near-term quantum devices, distinguishing them from other ensemble-learning proposals that rely on resource-intensive hardware implementations involving multiqubit controlled unitaries and complex quantum subroutines such as quantum phase estimation [152,347], Grover search [348,349], and quantum mean estimation [350–352].

D. Ansatz selection and scalability

Ansatz selection plays a crucial role in preventing barren plateaus and achieving efficient scalability. Despite the theoretical work done to demonstrate provable advantage on synthetic datasets [353], more research is needed to understand the impact of entanglement in the model ansatz. Developing efficient methods to adjust parameter values and train quantum circuits to minimize specific loss functions in VQCs is an active area of research. Parameter initialization strategies for large-scale QNNs need to be explored to improve their scalability. To comprehend the scalability of QML methods for large problems, analyzing trainability and prediction error is necessary. Access to reliable quantum hardware is also crucial. In QML, training the model involves minimizing a loss function to find the optimal set of parameters. Quantum landscape theory explores the properties of this loss function landscape, focusing on challenges like local minima and barren plateaus [139]. Recently, there has been an active line of research, with open-source software, on quantum circuit selection tailored to underlying quantum devices [115,117] that partially addresses this question.

E. Backpropagation and scalability

Backpropagation plays a crucial role in the success of deep neural networks by efficiently computing gradients using the computational graph. This computational advantage allows for the training of deep networks. Recent applications like ChatGPT [354] utilize backpropagation during training for the efficient calculation of gradients for batches of input-output pairs which enables scalability in handling large datasets. This technique allows for parallel computation and parameter updates, contributing to the model's ability to handle increased complexity. However, when it comes to parameterized quantum circuits, backpropagation is significantly less efficient compared to classical circuits. This inefficiency directly impacts the trainability of quantum models. The existing gradient methods used in parameterized quantum models lack the scaling properties of backpropagation, raising questions about their computational complexity.

Addressing this issue, a recent study [204] highlights the need to explore alternative architectures and optimization methods to improve the scalability of quantum models. The authors suggest that backpropagation may not be the appropriate optimization method for quantum models and propose an alternative while emphasizing the importance of finding optimization methods that can effectively handle the computational complexity of parameterized quantum circuits to enhance the trainability and scalability of quantum models.

F. QML model security

The current state of QML lacks privacy-preserving features, raising concerns about the potential exposure of sensitive information in machine learning datasets [355–358]. To address this issue, it is crucial to implement privacy-preserving algorithms in QML, such as differential privacy, which minimizes the influence of individual data points on the training process. However, the application of differential privacy in the context of QML requires further study and exploration to ensure effective privacy protection in machine learning models. Recently, Watkins *et al.* [356] demonstrated the first proof of principle of privacy-preserving QML.

Additionally, another study [359] highlights the robustness of VQCs against adversarial attacks, even outperforming classical neural networks in this regard. The authors propose that combining the outcomes of both quantum and classical networks can have significant implications for enhancing security and reliability in applications like autonomous vehicles, cyber security, and surveillance robotic systems, thereby opening up new possibilities for addressing security by leveraging the power of quantum and classical models together. Recent work in Ref. [96] investigates quantum federated learning in the context of secure data handling and sharing in distributed settings and cooperative learning.

G. Toward explainable QML models

Realizing explainable AI (XAI) is a challenging yet crucial research field that provides insights into the decision-making process of machine learning models, addressing aspects such as fairness and security, especially in domains like medical research [360,361]. An organic extension to QML also necessitates studying the fundamental aspects of QML [63,131,362,365]. Given this context, exploring explainability in QML, called explainable QML (XQML), aims to provide humanly understandable interpretations of QML systems, similar to classical ML. Currently, the field of XQML remains relatively unexplored; however, it holds great potential for yielding fundamental insights, particularly given that QML is still in its early stages. Certain aspects of XQML, such as intuitively explaining quantum feature spaces and understanding the behavior of QML models in relation to QPUs through transformations and operations, go beyond the scope of classical XAI. Addressing these aspects may require the development of entirely new approaches to explainability or interpretability. The exploration of XQML, in conjunction with the prospect of improved hardware, may be considered more promising than solely focusing on identifying quantum advantages [63,131].

H. Hyperparameter choices and transparency

The lack of extensive discussions on hyperparameter choices in current quantum machine learning studies poses challenges to transparency, interpretability, and progress in the field. Many studies that demonstrate promising results on benchmark datasets often fail to provide open-source reference implementations of their competitive algorithms. This lack of accessibility hinders the reproducibility of results and raises concerns about potential positive bias, where only a selected set of experiments showing favorable model performance are reported, while others are disregarded.

Furthermore, reproducibility can be challenging when working with open-source projects like Qiskit [366], which undergo regular updates and improvements to enhance functionality, address bugs, and introduce new features. These updates can lead to deprecated features and code incompatibility, affecting the ability to reproduce results.

To address this issue, researchers should prioritize providing comprehensive documentation that includes detailed information on hyperparameter selection. This documentation should offer insights into the decision-making processes behind choosing specific hyperparameters and discuss the potential implications of different selections. By sharing this information, researchers can enhance transparency and enable others to replicate and build upon their work effectively. Furthermore, the provision of open-source reference implementations is crucial for fostering collaboration, promoting rigorous evaluation, and advancing the field collectively. Accessible and reproducible code allows researchers to validate and compare different approaches, facilitating the identification of strengths and weaknesses in quantum machine learning algorithms.

Reproducibility in experiments on quantum hardware can be challenging due to noise, limited access, calibration issues, and algorithmic variability. To address these challenges, researchers should thoroughly document the experimental setup, share the source code, use standardized benchmarks, and promote collaboration and open science practices.

I. Federated learning and quantum boosting

Thoroughly studying the use of federated learning to distribute computational tasks among limited-capability quantum machines, coupled with investigating the potential of quantum boosting classifiers, can significantly enhance the scalability and utilization of available near-term noisy devices [299,365,366]. These research directions hold promise for leveraging the collective power of distributed quantum resources and improving the overall performance of quantum machine learning systems.

The challenge of scalability in quantum algorithms and its impact on real-world applications is a critical issue that requires further investigation. Recent findings [367] demonstrate the successful measurement of accurate expectation values for large circuit volumes using a noisy 127-qubit quantum processor, highlighting the potential of quantum computing in a pre-fault-tolerant era. However, it is important to acknowledge that the error mitigation techniques discussed in Ref. [367] suffer from exponential computational time as the number of qubits increases. Moreover, comparing these techniques to “brute-force” classical methods may not be entirely fair, as it fails to acknowledge the significant advancements made by classical methods in simulating quantum dynamics.

To advance the field, it is crucial to establish a shared community consensus on identifying problems that are both interesting for practical applications and genuinely challenging to simulate classically. This requires acknowledging the progress made by classical methods and not solely equating high entanglement with classical simulation difficulty. It is necessary to continue the development and exploration of both quantum and classical approximation methods, as they provide valuable benchmarks for each other's capabilities. By addressing these challenges and fostering collaboration between quantum and classical approaches, we can drive the field forward and unlock the full potential of quantum computing.

ACKNOWLEDGMENTS

We want to emphasize that the list of papers presented in this draft has been compiled to include as many relevant papers as possible. However, we acknowledge that, due to the rapid pace of development in this field, our coverage may not be exhaustive. Our thanks go to arXiv for use of its open-access interoperability that enabled us to gather almost all of the papers referred to in this paper. We also offer special thanks to Tamiya Onodera of IBM Quantum, IBM Research, Tokyo, and Shesha Raghunathan of IBM Quantum, IBM Research, India, for their valuable discussions and comments. This work is partially supported by Kakenhi Grants No. 23K28035, No. 23K18501, No. 20H05967, and No. 21H05052.

This paper was prepared for informational purposes by the Global Technology Applied Research Center of JPMorgan Chase & Co. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

[1] J. Preskill, in *Feynman Lectures on Computation* (CRC Press, 2023), p. 193.

[2] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[3] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. **41**, 303 (1999).

[4] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, Nature **414**, 883 (2001).

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

[6] R. P. Feynman, in *Feynman and Computation* (CRC Press, 2018), p. 133.

[7] S. Das Sarma, D.-L. Deng, and L.-M. Duan, Machine learning meets quantum physics, Phys. Today **72**, 48 (2019).

[8] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014).

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

[10] N. Meyer, C. Ufrecht, M. Periyasamy, D. D. Scherer, A. Plinge, and C. Mutschler, A survey on quantum reinforcement learning, arXiv:2211.03464.

[11] V. Dunjko, J. M. Taylor, and H. J. Briegel, in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE Press, Banff, AB, Canada, 2017), p. 282.

[12] D. Dong, C. Chen, H. Li, and T.-J. Tarn, Quantum reinforcement learning, IEEE Trans. Syst., Man, Cybern., B (Cybernetics) **38**, 1207 (2008).

[13] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Validating quantum computers using randomized model circuits, Phys. Rev. A **100**, 032328 (2019).

[14] A. Fedorov, N. Gisin, S. Beloussov, and A. Lvovsky, Quantum computing at the quantum advantage threshold: A down-to-business review, arXiv:2203.17181.

[15] A. Jadhav, A. Rasool, and M. Gyanchandani, Quantum machine learning: Scope for real-world problems, Procedia Comput. Sci. **218**, 2612 (2023).

[16] J. Singh and K. S. Bhangu, Contemporary quantum computing use cases: Taxonomy, review and challenges, Arch. Comput. Methods Eng. **30**, 615 (2023).

[17] F. Bova, A. Goldfarb, and R. G. Melko, Commercial applications of quantum computing, EPJ Quantum Technol. **8**, 2 (2021).

[18] V. Hassija, V. Chamola, V. Saxena, V. Chanana, P. Parashari, S. Mumtaz, and M. Guizani, Present landscape of quantum computing, IET Quantum Commun. **1**, 42 (2020).

[19] A. Bayerstadler *et al.*, Q. Technology, and A. C. QUTAC, Industry quantum computing applications, EPJ Quantum Technol. **8**, 25 (2021).

[20] J. Preskill, Quantum computing: Pro and con, Proc. R. Soc. London. Ser. A: Math., Phys. Eng. Sci. **454**, 469 (1998).

[21] S. Gupta, S. Modgil, P. C. Bhatt, C. J. Chiappetta Jabbour, and S. Kamble, Quantum computing led innovation for achieving a more sustainable Covid-19 healthcare industry, Technovation **120**, 102544 (2023).

[22] M. Coccia, Technological trajectories in quantum computing to design a quantum ecosystem for industrial change, Technol. Anal. Strateg. Manag. **0**, 1 (2022).

[23] J. Wurtz, P. Lopes, N. Gemelke, A. Keesling, and S. Wang, Industry applications of neutral-atom quantum computing solving independent set problems, arXiv:2205.08500.

[24] K. K. Sharma, Quantum machine learning and its supremacy in high energy physics, Mod. Phys. Lett. A **36**, 2030024 (2021).

[25] W. Guan, G. Perdue, A. Pesah, M. Schuld, K. Terashi, S. Vallecorsa, and J.-R. Vlimant, Quantum machine learning

in high energy physics, Mach. Learn.: Sci. Technol. **2,** 011003 (2021).

[26] T. S. Humble, A. Delgado, R. Pooser, C. Seck, R. Bennink, V. Leyton-Ortega, C.-C. J. Wang, E. Dumitrescu, T. Morris, and K. Hamilton *et al.*, Snowmass white paper: Quantum computing systems and software for high-energy physics research, arXiv:2203.07091.

[27] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, Quantum computing for finance: State-of-the-art and future prospects, IEEE Trans. Quantum Eng. **1,** 1 (2020).

[28] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, A survey of quantum computing for finance, arXiv:2201.02773.

[29] R. Orús, S. Mugel, and E. Lizaso, Quantum computing for finance: Overview and prospects, Rev. Phys. **4,** 100028 (2019).

[30] S. Wilkens and J. Moorhouse, Quantum computing for financial risk measurement, Quantum Inf. Process. **22,** 51 (2023).

[31] F. F. Flöther, The state of quantum computing applications in health and medicine, arXiv:2301.09106.

[32] L. Wei, H. Liu, J. Xu, L. Shi, Z. Shan, B. Zhao, and Y. Gao, Quantum machine learning in medical image analysis: A survey, Neurocomputing **525,** 42 (2023).

[33] R. Ur Rasool, H. F. Ahmad, W. Rafique, A. Qayyum, J. Qadir, and Z. Anwar, Quantum computing for healthcare: A review, Future Internet **15,** 94 (2023).

[34] M. G. Vakili, C. Gorgulla, A. Nigam, D. Bezrukov, D. Varoli, A. Aliper, D. Polykovsky, K. M. P. Das, J. Snider, and A. Lyakisheva *et al.*, Quantum computing-enhanced algorithm unveils novel inhibitors for KRAS, arXiv:2402.08210.

[35] O. A. Von Lilienfeld, Quantum machine learning in chemical compound space, Angew. Chem. Int. Ed. **57,** 4164 (2018).

[36] O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, Exploring chemical compound space with quantum-based machine learning, Nat. Rev. Chem. **4,** 347 (2020).

[37] P. Zaspel, B. Huang, H. Harbrecht, and O. A. von Lilienfeld, Boosting quantum machine learning models with a multilevel combination technique: Pople diagrams revisited, J. Chem. Theory. Comput. **15,** 1546 (2018).

[38] O. A. Von Lilienfeld, First principles view on chemical compound space: Gaining rigorous atomistic control of molecular properties, Int. J. Quantum. Chem. **113,** 1676 (2013).

[39] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Rev. Mod. Phys. **92,** 015003 (2020).

[40] M. Sajjan, J. Li, R. Selvarajan, S. H. Sureshbabu, S. S. Kale, R. Gupta, V. Singh, and S. Kais, Quantum machine learning for chemistry and physics, Chem. Soc. Rev. **51,** 6475 (2022).

[41] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, Quantum classifier with tailored quantum kernel, npj Quantum Inf. **6,** 1 (2020).

[42] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nat. Commun. **12,** 1 (2021).

[43] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122,** 040504 (2019).

[44] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nat. Phys. **17,** 1013 (2021).

[45] M. Schuld, Supervised quantum machine learning models are kernel methods, arXiv:2101.11020.

[46] K. Bartkiewicz, C. Gneiting, A. Černoch, K. Jiráková, K. Lemr, and F. Nori, Experimental kernel-based quantum machine learning in finite feature space, Sci. Rep. **10,** 1 (2020).

[47] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567,** 209 (2019).

[48] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, Quantum Sci. Technol. **2,** 045001 (2017).

[49] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nat. Comput. Sci. **1,** 403 (2021).

[50] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, and L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3,** 625 (2021).

[51] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, Training deep quantum neural networks, Nat. Commun. **11,** 1 (2020).

[52] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, Continuous-variable quantum neural networks, Phys. Rev. Res. **1,** 033063 (2019).

[53] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002.

[54] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, Quantum Inf. Process. **13,** 2567 (2014).

[55] B. E. Boser, I. M. Guyon, and V. N. Vapnik, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (Association for Computing Machinery, New York, NY, USA, 1992), p. 144.

[56] W. S. Noble, What is a support vector machine?, Nat. Biotechnol. **24,** 1565 (2006).

[57] W. Li and D.-L. Deng, Recent advances for quantum classifiers, Sci. China Phys., Mech. Astron. **65,** 220301 (2022).

[58] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, Perspectives of quantum annealing: Methods and implementations, Rep. Prog. Phys. **83,** 054401 (2020).

[59] A. Das and B. K. Chakrabarti, Colloquium: Quantum annealing and analog quantum computation, Rev. Mod. Phys. **80,** 1061 (2008).

[60] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, Quantum annealing: A new method for minimizing multidimensional functions, Chem. Phys. Lett. **219,** 343 (1994).

[61] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum

machine learning: A classical perspective, Proc. R. Soc. A: Math., Phys. Eng. Sci. **474**, 20170551 (2018).

[62] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).

[63] M. Schuld and N. Killoran, Is Quantum Advantage the Right Goal for Quantum Machine Learning?, PRX Quantum **3**, 030101 (2022).

[64] E. R. Muten, T. T. Yusuf, and A. V. Tomut, in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Broomfield, CO, USA, 2021), p. 82.

[65] A. Blance and M. Spannowsky, Quantum machine learning for particle physics using a variational quantum classifier, J. High Energy Phys. **2021**, 1 (2021).

[66] K. Terashi, M. Kaneda, T. Kishimoto, M. Saito, R. Sawada, and J. Tanaka, Event classification with quantum machine learning in high-energy physics, Comput. Softw. Big Sci. **5**, 1 (2021).

[67] S. L. Wu, J. Chan, W. Guan, S. Sun, A. Wang, C. Zhou, M. Livny, F. Carminati, A. Di Meglio, and A. C. Li *et al.*, Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM Quantum computer simulator and hardware with 10 qubits, J. Phys. G: Nucl. Part. Phys. **48**, 125003 (2021).

[68] S. L. Wu, S. Sun, W. Guan, C. Zhou, J. Chan, C. L. Cheng, T. Pham, Y. Qian, A. Z. Wang, and R. Zhang *et al.*, Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the LHC, Phys. Rev. Res. **3**, 033221 (2021).

[69] J. Y. Araz and M. Spannowsky, Classical versus quantum: Comparing tensor-network-based quantum circuits on Large Hadron Collider data, Phys. Rev. A **106**, 062423 (2022).

[70] K. A. Woźniak, V. Belis, E. Puljak, P. Barkoutsos, G. Dissertori, M. Grossi, M. Pierini, F. Reiter, I. Tavernelli, and S. Vallecorsa, Quantum anomaly detection in the latent space of proton collision events at the LHC, arXiv:2301.10780.

[71] T. Li, Z. Yao, X. Huang, J. Zou, T. Lin, and W. Li, in *Journal of Physics: Conference Series* (IOP Publishing, 2023), Vol. 2438, p. 012071.

[72] E. Bermot, C. Zoufal, M. Grossi, J. Schuhmacher, F. Tacchino, S. Vallecorsa, and I. Tavernelli, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2023), Vol. 1, p. 331.

[73] D. Cugini, D. Gerace, P. Govoni, A. Perego, and D. Valsecchi, Comparing quantum and classical machine learning for vector boson scattering background reduction at the Large Hadron Collider, Quantum Mach. Intell. **5**, 35 (2023).

[74] M. C. Peixoto, M. Crispim Romão, M. G. J. Oliveira, and I. Ochoa, Fitting a collider in a quantum computer: Tackling the challenges of quantum machine learning for big datasets, Front. Artif. Intell. **6**, 1268852 (2023).

[75] J. Lazar, S. G. Olavarrieta, G. Gatti, C. A. Argüelles, and M. Sanz, New pathways in neutrino physics via quantum-encoded data analysis, arXiv:2402.19306.

[76] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza, Determining the proton content with a quantum computer, Phys. Rev. D **103**, 034027 (2021).

[77] C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska, and S. Carrazza, Style-based quantum generative adversarial networks for Monte Carlo events, Quantum **6**, 777 (2022).

[78] S. Y. Chang, E. Agnew, E. Combarro, M. Grossi, S. Herbert, and S. Vallecorsa, in *Journal of Physics: Conference Series* (IOP Publishing, 2023), Vol. 2438, p. 012062.

[79] F. Rehm, S. Vallecorsa, K. Borras, D. Krücker, M. Grossi, and V. Varo, Precise image generation on current noisy quantum computing devices, Quantum Sci. Technol. **9**, 015009 (2023).

[80] V. S. Ngairangbam, M. Spannowsky, and M. Takeuchi, Anomaly detection in high-energy physics using a quantum autoencoder, Phys. Rev. D **105**, 095004 (2022).

[81] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, npj Quantum Inf. **5**, 1 (2019).

[82] A. Ray, S. S. Guddanti, V. Ajith, and D. Vinayagamurthy, Classical ensemble of quantum-classical ML algorithms for phishing detection in Ethereum transaction networks, arXiv:2211.00004.

[83] T. Suzuki, T. Hasebe, and T. Miyazaki, Quantum support vector machines for classification and regression on a trapped-ion quantum computer, arXiv:2307.02091.

[84] S. Thakkar, S. Kazdaghli, N. Mathur, I. Kerenidis, A. J. Ferreira-Martins, and S. Brito, Improved financial forecasting via quantum machine learning, arXiv:2306.12965.

[85] A. Martin, B. Candelas, Á. Rodríguez-Rozas, J. D. Martín-Guerrero, X. Chen, L. Lamata, R. Orús, E. Solano, and M. Sanz, Toward pricing financial derivatives with an IBM Quantum computer, Phys. Rev. Res. **3**, 013167 (2021).

[86] C. Zoufal, R. V. Mishmash, N. Sharma, N. Kumar, A. Sheshadri, A. Deshmukh, N. Ibrahim, J. Gacon, and S. Woerner, Variational quantum algorithm for unconstrained black box binary optimization: Application to feature selection, Quantum **7**, 909 (2023).

[87] H. Yano, Y. Suzuki, K. M. Itoh, R. Raymond, and N. Yamamoto, Efficient discrete feature encoding for variational quantum classifier, IEEE Trans. Quantum Eng. **2**, 1 (2021).

[88] E. Acar and I. Yilmaz, COVID-19 detection on IBM quantum computer with classical-quantum transfer learning, Turk. J. Electr. Eng. Comput. Sci. **29**, 46 (2021).

[89] N. Mathur, J. Landman, Y. Y. Li, M. Strahm, S. Kazdaghli, A. Prakash, and I. Kerenidis, Medical image classification via quantum neural networks, arXiv:2109.01831.

[90] W. Ren, W. Li, S. Xu, K. Wang, W. Jiang, F. Jin, X. Zhu, J. Chen, Z. Song, and P. Zhang *et al.*, Experimental quantum adversarial learning with programmable superconducting qubits, Nat. Comput. Sci. **2**, 711 (2022).

[91] S. Moradi, C. Brandner, C. Spielvogel, D. Krajnc, S. Hillmich, R. Wille, W. Drexler, and L. Papp, Clinical data classification with noisy intermediate scale quantum computers, Sci. Rep. **12**, 1851 (2022).

[92] Z. Krunic, F. F. Flöther, G. Seegan, N. D. Earnest-Noble, and O. Shehab, Quantum kernels for real-world

predictions based on electronic health records, IEEE Trans. Quantum Eng. **3**, 1 (2022).

[93] V. Azevedo, C. Silva, and I. Dutra, Quantum transfer learning for breast cancer detection, Quantum Mach. Intell. **4**, 5 (2022).

[94] X. Vasques, H. Paik, and L. Cif, Application of quantum machine learning using quantum kernel algorithms on multiclass neuron M-type classification, Sci. Rep. **13**, 11541 (2023).

[95] S. Mensa, E. Sahin, F. Tacchino, P. K. Barkoutsos, and I. Tavernelli, Quantum machine learning framework for virtual screening in drug discovery: A prospective quantum advantage, Mach. Learn.: Sci. Technol. **4**, 015023 (2023).

[96] N. Innan, M. A.-Z. Khan, A. Marchisio, M. Shafique, and M. Bennai, FedQNN: Federated learning using quantum neural networks, arXiv:2403.10861.

[97] M. Benedetti, B. Coyle, M. Fiorentini, M. Lubasch, and M. Rosenkranz, Variational inference with a quantum computer, Phys. Rev. Appl. **16**, 044057 (2021).

[98] R. A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. **7**, 179 (1936).

[99] E. Anderson, The species problem in Iris, Ann. Missouri Bot. Gard. **23**, 457 (1936).

[100] L. Deng, The MNIST database of handwritten digit images for machine learning research, IEEE Signal Process. Mag. **29**, 141 (2012).

[101] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, arXiv:1708.07747.

[102] F. E. Harrell Jr., Titanic dataset, https://www.openml.org/d/40945 (2017).

[103] T. Allam Jr, A. Bahmanyar, R. Biswas, M. Dai, L. Galbany, R. Hložek, E. E. Ishida, S. W. Jha, D. O. Jones, and R. Kessler *et al.*, The Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC): Data set, arXiv:1810.00001.

[104] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. **47**, 547 (2009).

[105] Z. Li, X. Liu, N. Xu, and J. Du, Experimental Realization of a Quantum Support Vector Machine, Phys. Rev. Lett. **114**, 140504 (2015).

[106] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, Hierarchical quantum classifiers, npj Quantum Inf. **4**, 1 (2018).

[107] W. Cappelletti, R. Erbanni, and J. Keller, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2020), p. 22.

[108] N. Thumwanit, C. Lortaraprasert, H. Yano, and R. Raymond, Trainable discrete feature embeddings for variational quantum classifier, arXiv:2106.09415.

[109] E. Peters, J. Caldeira, A. Ho, S. Leichenauer, M. Mohseni, H. Neven, P. Spentzouris, D. Strain, and G. N. Perdue, Machine learning of high dimensional data on a noisy quantum processor, npj Quantum Inf. **7**, 1 (2021).

[110] C. Blank, A. J. da Silva, L. P. de Albuquerque, F. Petruccione, and D. K. Park, Compact quantum kernel-based binary classifier, Quantum Sci. Technol. **7**, 045007 (2022).

[111] I. Koyasu, R. Raymond, and H. Imai, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2023), Vol. 01, p. 457.

[112] T. Haug, C. N. Self, and M. Kim, Quantum machine learning of large datasets using randomized measurements, Mach. Learn.: Sci. Technol. **4**, 015005 (2023).

[113] R. D. M. Simões, P. Huber, N. Meier, N. Smailov, R. M. Füchslin, and K. Stockinger, Experimental evaluation of quantum machine learning algorithms, IEEE Access **11**, 6197 (2023).

[114] scikit-learn developers, scikit-learn: Machine Learning in Python, https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_multilabel_classification.html?highlight=make_multilabel_classification (2022).

[115] T. Chen, Z. Zhang, H. Wang, J. Gu, Z. Li, D. Z. Pan, F. T. Chong, S. Han, and Z. Wang, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2023), Vol. 1, p. 51.

[116] A. Melo, N. Earnest-Noble, and F. Tacchino, Pulse-efficient quantum machine learning, Quantum **7**, 1130 (2023).

[117] S. Anagolum, N. Alavisamani, P. Das, M. Qureshi, E. Kessler, and Y. Shi, Élivágar: Efficient quantum circuit search for classification, arXiv:2401.09393.

[118] K. Shen, B. Jobst, E. Shishenina, and F. Pollmann, Classification of the Fashion-MNIST dataset on a quantum computer, arXiv:2403.02405.

[119] S. U. Khan, A. J. Awan, and G. Vall-Llosera, K-means clustering on noisy intermediate scale quantum computers, arXiv:1909.12183.

[120] S. Johri, S. Debnath, A. Mocherla, A. Singk, A. Prakash, J. Kim, and I. Kerenidis, Nearest centroid classification on a trapped ion quantum computer, npj Quantum Inf. **7**, 1 (2021).

[121] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, and Y. Xu *et al.*, Experimental quantum generative adversarial networks for image generation, Phys. Rev. Appl. **16**, 024051 (2021).

[122] M. S. Rudolph, N. B. Toussaint, A. Katabarwa, S. Johri, B. Peropadre, and A. Perdomo-Ortiz, Generation of High-Resolution Handwritten Digits with an Ion-Trap Quantum Computer, Phys. Rev. X **12**, 031010 (2022).

[123] J. Gibbs, Z. Holmes, M. C. Caro, N. Ezzell, H.-Y. Huang, L. Cincio, A. T. Sornborger, and P. J. Coles, Dynamical simulation via quantum machine learning with provable generalization, Phys. Rev. Res. **6**, 013241 (2024).

[124] K. Bartkiewicz, P. Tulewicz, J. Roik, and K. Lemr, Synergic quantum generative machine learning, Sci. Rep. **13**, 12893 (2023).

[125] J. Herrmann, S. M. Llima, A. Remm, P. Zapletal, N. A. McMahon, C. Scarato, F. Swiadek, C. K. Andersen, C. Hellings, and S. Krinner *et al.*, Realizing quantum convolutional neural networks on a superconducting quantum processor to recognize quantum phases, Nat. Commun. **13**, 1 (2022).

[126] M. Gong, H.-L. Huang, S. Wang, C. Guo, S. Li, Y. Wu, Q. Zhu, Y. Zhao, S. Guo, and H. Qian *et al.*, Quantum neuronal sensing of quantum many-body states on a 61-qubit programmable superconducting processor, Sci. Bull. **68**, 906 (2023).

[127] X. Pan, Z. Lu, W. Wang, Z. Hua, Y. Xu, W. Li, W. Cai, X. Li, H. Wang, and Y.-P. Song *et al.*, Deep quantum neural networks on a superconducting processor, Nat. Commun. **14**, 4006 (2023).

[128] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, and J. Preskill, Quantum advantage in learning from experiments, Science **376**, 1182 (2022).

[129] A. Nakayama, K. Mitarai, L. Placidi, T. Sugimoto, and K. Fujii, VQE-generated quantum circuit dataset for machine learning, arXiv:2302.09751.

[130] J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, Covariant quantum kernels for data with group structure, Nat. Phys. **20**, 1 (2024).

[131] R. Heese, T. Gerlach, S. Mücke, S. Müller, M. Jakobs, and N. Piatkowski, Explainable quantum machine learning, arXiv:2301.09138.

[132] G. Gentinetta, D. Sutter, C. Zoufal, B. Fuller, and S. Woerner, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2023), Vol. 1, p. 256.

[133] K. Huang, Z.-A. Wang, C. Song, K. Xu, H. Li, Z. Wang, Q. Guo, Z. Song, Z.-B. Liu, and D. Zheng *et al.*, Quantum generative adversarial networks with multiple superconducting qubits, npj Quantum Inf. **7**, 165 (2021).

[134] D. Kreplin and M. Roth, Reduction of finite sampling noise in quantum neural networks, arXiv:2306.01639.

[135] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

[136] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning* (Springer, Springer New York, NY, 2006).

[137] M. I. Jordan and T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science **349**, 255 (2015).

[138] D. W. Hook, S. J. Porter, and C. Herzog, Dimensions: Building context for search and evaluation, Front. Res. Metr. Anal. **3**, 23 (2018). https://www.frontiersin.org/articles/10.3389/frma.2018.00023/pdf.

[139] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, Nat. Comput. Sci. **2**, 567 (2022).

[140] S. Aaronson, Read the fine print, Nat. Phys. **11**, 291 (2015).

[141] R. LaRose and B. Coyle, Robust data encodings for quantum classifiers, Phys. Rev. A **102**, 032420 (2020).

[142] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning, arXiv:2001.03622.

[143] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (Springer, 2018), Vol. 17.

[144] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).

[145] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[146] M. Schuld, M. Fingerhuth, and F. Petruccione, Implementing a distance-based classifier with a quantum interference circuit, EPL (Europhys. Lett.) **119**, 60002 (2017).

[147] A. C. Vazquez and S. Woerner, Efficient state preparation for quantum amplitude estimation, Phys. Rev. Appl. **15**, 034027 (2021).

[148] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[149] A. Gilliam, S. Woerner, and C. Gonciulea, Grover adaptive search for constrained polynomial binary optimization, Quantum **5**, 428 (2021).

[150] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Contemp. Math. **305**, 53 (2002).

[151] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, USA, 1996), p. 212.

[152] G. D'Ariano, C. Macchiavello, and M. Sacchi, On the general problem of quantum phase estimation, Phys. Lett. A **248**, 103 (1998).

[153] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, Phys. Rev. A **83**, 032302 (2011).

[154] Y. R. Sanders, G. H. Low, A. Scherer, and D. W. Berry, Black-Box Quantum State Preparation without Arithmetic, Phys. Rev. Lett. **122**, 020502 (2019).

[155] V. V. Shende, S. S. Bullock, and I. L. Markov, Synthesis of quantum-logic circuits, IEEE TCAD **25**, 1000 (2006).

[156] L. K. Grover, Synthesis of Quantum Superpositions by Quantum Computation, Phys. Rev. Lett. **85**, 1334 (2000).

[157] I. Kerenidis, J. Landman, and N. Mathur, Classical and quantum algorithms for orthogonal neural networks, arXiv:2106.07198.

[158] I. Kerenidis, A method for loading classical data into quantum states for applications in machine learning and optimization (2020).

[159] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, A divide-and-conquer algorithm for quantum state preparation, Sci. Rep. **11**, 1 (2021).

[160] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Transformation of quantum states using uniformly controlled rotations, arXiv:quant-ph/0407010.

[161] 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).

[162] J. Li, M. Alam, M. S. Congzhou, J. Wang, N. V. Dokholyan, and S. Ghosh, in *2021 58th ACM/IEEE Design Automation Conference (DAC)* (IEEE, San Francisco, CA, USA, 2021), p. 1356.

[163] K. Batra, K. M. Zorn, D. H. Foil, E. Minerali, V. O. Gawriljuk, T. R. Lane, and S. Ekins, Quantum machine learning algorithms for drug discovery applications, J. Chem. Inf. Model. **61**, 2641 (2021).

[164] P. Easom-McCaldin, A. Bouridane, A. Belatreche, and R. Jiang, On depth, robustness and performance using the data re-uploading single-qubit classifier, IEEE Access **9**, 65127 (2021).

[165] A. Cervera-Lierta, J. S. Kottmann, and A. Aspuru-Guzik, Meta-Variational Quantum Eigensolver: Learning Energy Profiles of Parameterized Hamiltonians for Quantum Simulation, PRX Quantum **2**, 020329 (2021).

[166] S. Lloyd, Universal quantum simulators, Science **273,** 1073 (1996).

[167] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Theory of variational quantum simulation, Quantum **3,** 191 (2019).

[168] Y.-X. Yao, N. Gomes, F. Zhang, C.-Z. Wang, K.-M. Ho, T. Iadecola, and P. P. Orth, Adaptive Variational Quantum Dynamics Simulations, PRX Quantum **2,** 030307 (2021).

[169] S. Barison, F. Vicentini, and G. Carleo, An efficient quantum algorithm for the time evolution of parameterized circuits, Quantum **5,** 512 (2021).

[170] G. H. Low, V. Kliuchnikov, and N. Wiebe, Well-conditioned multiproduct Hamiltonian simulation, arXiv:1907.11679.

[171] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, arXiv:1202.5822.

[172] B. Albrecht, C. Dalyac, L. Leclerc, L. Ortiz-Gutiérrez, S. Thabet, M. D'Arcangelo, J. R. Cline, V. E. Elfving, L. Lassablière, and H. Silvério *et al.*, Quantum feature maps for graph machine learning on a neutral atom quantum processor, Phys. Rev. A **107,** 042615 (2023).

[173] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, The complexity of quantum support vector machines, Quantum **8,** 1225 (2024).

[174] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, Phys. Rev. Lett. **119,** 180509 (2017).

[175] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Error mitigation extends the computational reach of a noisy quantum processor, Nature **567,** 491 (2019).

[176] Y. Li and S. C. Benjamin, Efficient Variational Quantum Simulator Incorporating Active Error Minimization, Phys. Rev. X **7,** 021050 (2017).

[177] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, Quantum Fingerprinting, Phys. Rev. Lett. **87,** 167902 (2001).

[178] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, New J. Phys. **20,** 113022 (2018).

[179] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, J. Phys. Soc. Jpn. **90,** 032001 (2021).

[180] X. Wang, Y. Du, Y. Luo, and D. Tao, Towards understanding the power of quantum kernels in the NISQ era, Quantum **5,** 531 (2021).

[181] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations, J. High Energy Phys. **2014,** 1 (2014).

[182] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103,** 032430 (2021).

[183] A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, Layerwise learning for quantum neural networks, Quantum Mach. Intell. **3,** 1 (2021).

[184] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, and F. Petruccione, A comparison of various classical optimizers for a variational quantum linear solver, Quantum Inf. Process. **20,** 1 (2021).

[185] M. Benedetti, M. Fiorentini, and M. Lubasch, Hardware-efficient variational quantum algorithms for time evolution, Phys. Rev. Res. **3,** 033083 (2021).

[186] M. Ostaszewski, E. Grant, and M. Benedetti, Structure optimization for parameterized quantum circuits, Quantum **5,** 391 (2021).

[187] R. M. Parrish, J. T. Iosue, A. Ozaeta, and P. L. McMahon, A Jacobi diagonalization and Anderson acceleration algorithm for variational quantum algorithm parameter optimization, arXiv:1904.03206.

[188] K. M. Nakanishi, K. Fujii, and S. Todo, Sequential minimal optimization for quantum-classical hybrid algorithms, Phys. Rev. Res. **2,** 043158 (2020).

[189] J. G. Vidal and D. O. Theis, Calculus on parameterized quantum circuits, arXiv:1812.06323.

[190] C. Mc Keever and M. Lubasch, Classically optimized Hamiltonian simulation, Phys. Rev. Res. **5,** 023146 (2023).

[191] M. Wiedmann, M. Hölle, M. Periyasamy, N. Meyer, C. Ufrecht, D. D. Scherer, A. Plinge, and C. Mutschler, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2023), Vol. 1, p. 450.

[192] L. Bottou, in *Proceedings of COMPSTAT'2010* (Physica-Verlag HD, Heidelberg, 2010), p. 177.

[193] R. Chen, Z. Guang, C. Guo, G. Feng, and S.-Y. Hou, Pure quantum gradient descent algorithm and full quantum variational eigensolver, Front. Phys. **19,** 21202 (2024).

[194] P. Gao, K. Li, S. Wei, J. Gao, and G. Long, Quantum gradient algorithm for general polynomials, Phys. Rev. A **103,** 042403 (2021).

[195] A. Gilyén, S. Arunachalam, and N. Wiebe, in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (Society for Industrial and Applied Mathematics, San Diego, California, 2019), p. 1425.

[196] R. Wiersema, D. Lewis, D. Wierichs, J. Carrasquilla, and N. Killoran, Here comes the SU(N): Multivariate quantum gates and gradients, Quantum **8,** 1275 (2024).

[197] S. P. Jordan, Fast Quantum Algorithm for Numerical Gradient Estimation, Phys. Rev. Lett. **95,** 050501 (2005).

[198] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98,** 032309 (2018).

[199] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99,** 032331 (2019).

[200] K. M. Nakanishi, K. Fujii, and S. Todo, Sequential minimal optimization for quantum-classical hybrid algorithms, Phys. Rev. Res. **2,** 043158 (2020).

[201] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, and G. H. Booth *et al.*, The variational quantum eigensolver: A review of methods and best practices, Phys. Rep. **986,** 1 (2022).

[202] H. C. Watanabe, R. Raymond, Y.-Y. Ohnishi, E. Kaminishi, and M. Sugawara, in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Broomfield, CO, USA, 2021), p. 100.

[203] K. Wada, R. Raymond, Y. Sato, and H. C. Watanabe, Full optimization of a single-qubit gate on the generalized sequential quantum optimizer (2022).

[204] A. Abbas, R. King, H.-Y. Huang, W. J. Huggins, R. Movassagh, D. Gilboa, and J. McClean, On quantum backpropagation, information reuse, and cheating measurement collapse, Adv. Neural Inf. Process. Syst. **36** (2006).

[205] A. Kulshrestha, X. Liu, H. Ushijima-Mwesigwa, and I. Safro, Learning to optimize quantum neural network without gradients, arXiv:2304.07442.

[206] B. Neyshabur, R. R. Salakhutdinov, and N. Srebro, Path-sgd: Path-normalized optimization in deep neural networks, Adv. Neural Inf. Process. Syst. **28** (2015).

[207] S.-I. Amari, R. Karakida, and M. Oizumi, in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, edited by K. Chaudhuri and M. Sugiyama (PMLR, 2019), p. 694, https://proceedings.mlr.press/v89/amari19a.html.

[208] S.-I. Amari, Natural gradient works efficiently in learning, Neural Comput. **10**, 251 (1998).

[209] N. Yamamoto, On the natural gradient for variational quantum eigensolver, arXiv:1909.05074.

[210] A. W. Harrow and J. C. Napp, Low-Depth Gradient Measurements Can Improve Convergence in Variational Hybrid Quantum-Classical Algorithms, Phys. Rev. Lett. **126**, 140502 (2021).

[211] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, Quantum **4**, 269 (2020).

[212] D. C. Brody and L. P. Hughston, Geometric quantum mechanics, J. Geom. Phys. **38**, 19 (2001).

[213] R. Cheng, Quantum geometric tensor (Fubini-Study metric) in simple quantum system: A pedagogical introduction, arXiv:1012.1337.

[214] J. Gacon, C. Zoufal, G. Carleo, and S. Woerner, Simultaneous perturbation stochastic approximation of the quantum Fisher information, Quantum **5**, 567 (2021).

[215] J. C. Spall, An overview of the simultaneous perturbation method for efficient optimization, Johns Hopkins APL Tech. Dig. **19**, 482 (1998).

[216] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[217] S. Li, K. Jia, Y. Wen, T. Liu, and D. Tao, Orthogonal deep neural networks, IEEE Trans. Pattern. Anal. Mach. Intell. **43**, 1352 (2019).

[218] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, Commun. ACM **63**, 139 (2020).

[219] S. Lloyd and C. Weedbrook, Quantum Generative Adversarial Learning, Phys. Rev. Lett. **121**, 040502 (2018).

[220] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, Phys. Rev. A **98**, 012324 (2018).

[221] T. Li, S. Zhang, and J. Xia, Quantum generative adversarial network: A survey, CMC-Comput. Mater. Contin. **64**, 401 (2020).

[222] L. Hu, S.-H. Wu, W. Cai, Y. Ma, X. Mu, Y. Xu, H. Wang, Y. Song, D.-L. Deng, and C.-L. Zou *et al.*, Quantum generative adversarial learning in a superconducting quantum circuit, Sci. Adv. **5**, eaav2761 (2019).

[223] S. Lu, L.-M. Duan, and D.-L. Deng, Quantum adversarial machine learning, Phys. Rev. Res. **2**, 033212 (2020).

[224] M. T. West, S.-L. Tsang, J. S. Low, C. D. Hill, C. Leckie, L. C. Hollenberg, S. M. Erfani, and M. Usman, Towards quantum enhanced adversarial robustness in machine learning, Nat. Mach. Intell. **5**, 1 (2023).

[225] F. Verstraete, V. Murg, and J. I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, Adv. Phys. **57**, 143 (2008).

[226] G. Vidal, Class of Quantum Many-Body States That Can Be Efficiently Simulated, Phys. Rev. Lett. **101**, 110501 (2008).

[227] Y.-Y. Shi, L.-M. Duan, and G. Vidal, Classical simulation of quantum many-body systems with a tree tensor network, Phys. Rev. A **74**, 022320 (2006).

[228] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Towards quantum machine learning with tensor networks, Quantum Sci. Technol. **4**, 024001 (2019).

[229] R. Orús, Tensor networks for complex quantum systems, Nat. Rev. Phys. **1**, 538 (2019).

[230] D. Perez-Garcia, F. Verstraete, M. Wolf, and J. Cirac, Matrix product state representations, Quantum Inf. Comput. **7**, 401 (2007).

[231] L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. I. Latorre, Scaling of entanglement support for matrix product states, Phys. Rev. B **78**, 024410 (2008).

[232] E. C. Martín, K. Plekhanov, and M. Lubasch, Barren plateaus in quantum tensor network optimization, Quantum **7**, 974 (2023).

[233] J. Liu, S. Li, J. Zhang, and P. Zhang, Tensor networks for unsupervised machine learning, Phys. Rev. E **107**, L012103 (2023).

[234] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. Kim, Quantum generalisation of feedforward neural networks, npj Quantum Inf. **3**, 1 (2017).

[235] G. Verdon, J. Pye, and M. Broughton, A universal training algorithm for quantum deep learning, arXiv:1806.09729.

[236] C. Cao and X. Wang, Noise-assisted quantum autoencoder, Phys. Rev. Appl. **15**, 054012 (2021).

[237] D. F. Locher, L. Cardarelli, and M. Müller, Quantum error correction with quantum autoencoders, Quantum **7**, 942 (2023).

[238] T. Fawcett, *Pattern Recognition Letters* (2006), Vol. 27.

[239] D. J. Hand and R. J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, Mach. Learn. **45**, 171 (2001).

[240] K. Terashi, S. Ganguly, Y. Iiyama, T. Inada, W. Jang, S. Mizuhara, L. Nagano, R. Okubo, R. Sawada, and J. Tanaka *et al.*, Machine learning and architectural perspectives for quantum HEP applications (2021).

[241] IBM Quantum, IBM Quantum, https://quantum-computing.ibm.com/ (2019).

[242] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[243] K. Pearson, LIII. On lines and planes of closest fit to systems of points in space, London, Edinburgh, Dublin Philos. Mag. J. Sci. **2**, 559 (1901).

[244] I. T. Jolliffe and J. Cadima, Principal component analysis: A review and recent developments, Philos. Trans. R. Soc. A: Math., Phys. Eng. Sci. **374**, 20150202 (2016).

[245] T. Chen and C. Guestrin, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, NY, USA, 2016), p. 785.

[246] M. D. Zeiler, Adadelta: An adaptive learning rate method, arXiv:1212.5701.

[247] T. Haug, C. N. Self, and M. Kim, Large-scale quantum machine learning, arXiv:2108.01039.

[248] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert, and R. Sweke, Encoding-dependent generalization bounds for parametrized quantum circuits, Quantum **5**, 582 (2021).

[249] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, and F. Yan *et al.*, Identifying medical diagnoses and treatable diseases by image-based deep learning, Cell **172**, 1122 (2018).

[250] S. Gu, B. Kelly, and D. Xiu, Empirical asset pricing via machine learning, Rev. Financ. Stud. **33**, 2223 (2020).

[251] S. Nagel, *Machine Learning in Asset Pricing* (Princeton University Press, 2021), Vol. 1.

[252] M. Bagnara, Asset pricing and machine learning: A critical review, J. Econ. Surv. **38**, 27 (2022).

[253] T. Conlon, J. Cotter, and I. Kynigakis, Machine learning and factor-based portfolio optimization, arXiv:2107.13866.

[254] V.-D. Ta, C.-M. Liu, and D. Addis, in *Proceedings of the 9th International Symposium on Information and Communication Technology* (Association for Computing Machinery, New York, NY, USA, 2018), p. 98.

[255] S. Perrin and T. Roncalli, in *Machine Learning for Asset Management: New Developments and Financial Applications* (Wiley Online Library, 2020), p. 261.

[256] W. Chen, H. Zhang, M. K. Mehlawat, and L. Jia, Mean–variance portfolio optimization using machine learning-based stock price prediction, Appl. Soft Comput. **100**, 106943 (2021).

[257] G.-Y. Ban, N. El Karoui, and A. E. Lim, Machine learning and portfolio optimization, Manage. Sci. **64**, 1136 (2018).

[258] Y. Ma, R. Han, and W. Wang, Portfolio optimization with return prediction using deep learning and machine learning, Expert Syst. Appl. **165**, 113973 (2021).

[259] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, Machine learning techniques and data for stock market forecasting: A literature review, Expert Syst. Appl. **197**, 116659 (2022).

[260] R. J. Yan and C. X. Ling, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2007), p. 1038.

[261] X. Yuan, J. Yuan, T. Jiang, and Q. U. Ain, Integrated long-term stock selection models based on feature selection and machine learning algorithms for China stock market, IEEE Access **8**, 22672 (2020).

[262] K. C. Rasekhschaffe and R. C. Jones, Machine learning for stock selection, Financ. Anal. J. **75**, 70 (2019).

[263] X. Fu, J. Du, Y. Guo, M. Liu, T. Dong, and X. Duan, A machine learning framework for stock selection, arXiv:1806.01743.

[264] L. Min, J. Dong, J. Liu, and X. Gong, Robust mean-risk portfolio optimization using machine learning-based trade-off parameter, Appl. Soft Comput. **113**, 107948 (2021).

[265] M. Pinelis and D. Ruppert, Machine learning portfolio allocation, JFDS **8**, 35 (2022).

[266] S. R. Dahal, Utilizing generative AI for real-time financial market analysis opportunities and challenges, J. Intell. Inf. Syst. **8**, 1 (2023).

[267] O. S. Yee, S. Sagadevan, and N. H. A. H. Malim, Credit card fraud detection using machine learning as data mining technique, J. Telecommun., Electron. Comput. Eng. (JTEC) **10**, 23 (2018).

[268] R. A. Bauder and T. M. Khoshgoftaar, in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (IEEE, Cancun, Mexico, 2017), p. 858.

[269] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (IEEE, Noida, India, 2019), p. 488.

[270] V. N. Dornadula and S. Geetha, Credit card fraud detection using machine learning algorithms, Procedia Comput. Sci. **165**, 631 (2019).

[271] J. Perols, Financial statement fraud detection: An analysis of statistical and machine learning algorithms, Auditing: J. Pract. Theory **30**, 19 (2011).

[272] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)* (IEEE, East Sarajevo, Bosnia and Herzegovina, 2019), p. 1.

[273] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, in *2017 International Conference on Computing Networking and Informatics (ICCNI)* (IEEE, Lagos, Nigeria, 2017), p. 1.

[274] J. Cumming, D. D. Alrajeh, and L. Dickens, *An Investigation Into the Use of Reinforcement Learning Techniques Within the Algorithmic Trading Domain* (Imperial College London, London, UK, 2015), Vol. 58.

[275] Y. Li, W. Zheng, and Z. Zheng, Deep robust reinforcement learning for practical algorithmic trading, IEEE Access **7**, 108014 (2019).

[276] T. Théate and D. Ernst, An application of deep reinforcement learning to algorithmic trading, Expert Syst. Appl. **173**, 114632 (2021).

[277] T.-V. Pricope, Deep reinforcement learning in quantitative algorithmic trading: A review, arXiv:2106.00123.

[278] R. Gao, Z. Zhang, Z. Shi, D. Xu, W. Zhang, and D. Zhu, in *International Symposium on Artificial Intelligence and Robotics 2021* (SPIE, 2021), Vol. 11884, p. 262.

[279] I. E. Fisher, M. R. Garnsey, and M. E. Hughes, Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research, Intell. Syst. Account., Finance Manag. **23**, 157 (2016).

[280] K. Kwong, Financial forecasting using neural network or machine learning techniques, (University of Queensland, 2001), Vol. 13, p. 221.

[281] B. Krollner, B. Vanstone, and G. Finnie, in *European Symposium on Artificial Neural Networks: Computational Intelligence and Machine Learning* (2010), p. 25.

[282] H. Wasserbacher and M. Spindler, Machine learning for financial forecasting, planning and analysis: Recent developments and pitfalls, Digit. Finance **4**, 63 (2022).

[283] M. Chawla and P. Sarangi, A literature review on machine learning applications in financial forecasting, J. Technol. Manag. Grow. Economies **11**, 23 (2020).

[284] F. Kamalov, I. Gurrib, and K. Rajab, Financial forecasting with machine learning: Price vs return, J. Comput. Sci. **17**, 251 (2021).

[285] L. Di Persio and O. Honchar, Multitask machine learning for financial forecasting, Int. J. Circuits, Syst. Signal Process. **12**, 444 (2018).

[286] P. Roszkowska, Fintech in financial reporting and audit for fraud prevention and safeguarding equity investments, J. Account. Organ. Change **17**, 164 (2021).

[287] A. Koshiyama, E. Kazim, P. Treleaven, P. Rai, L. Szpruch, G. Pavey, G. Ahamat, F. Leutner, R. Goebel, and A. Knight *et al.*, Towards algorithm auditing: A survey on managing legal, ethical and technological risks of AI, ML and associated algorithms, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3778998 (2021), sSRN Working Paper.

[288] W. Xia, J. Zou, X. Qiu, F. Chen, B. Zhu, C. Li, D.-L. Deng, and X. Li, Configured quantum reservoir computing for multi-task machine learning, Sci. Bull. **68**, 2321 (2023).

[289] D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev, Quantum computing for finance, Nat. Rev. Phys. **5**, 450 (2023).

[290] A. Gu, A. Lowe, P. A. Dub, P. J. Coles, and A. Arrasmith, Adaptive shot allocation for fast convergence in variational quantum algorithms, arXiv:2108.10434.

[291] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, An adaptive optimizer for measurement-frugal variational algorithms, Quantum **4**, 263 (2020).

[292] A. Arrasmith, L. Cincio, R. D. Somma, and P. J. Coles, Operator sampling for shot-frugal optimization in variational algorithms, arXiv:2004.06252.

[293] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, Stochastic gradient descent for hybrid quantum-classical optimization, Quantum **4**, 314 (2020).

[294] A. Kandala, K. Temme, A. D. Corcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Extending the computational reach of a noisy superconducting quantum processor, arXiv:1805.04492.

[295] S. Blinov, B. Wu, and C. Monroe, Comparison of cloud-based ion trap and superconducting quantum computer architectures, AVS Quantum Sci. **3**, 033801 (2021).

[296] S. L. Wu and S. Yoo, Challenges and opportunities in quantum machine learning for high-energy physics, Nat. Rev. Phys. **4**, 143 (2022).

[297] A. Wack, H. Paik, A. Javadi-Abhari, P. Jurcevic, I. Faro, J. M. Gambetta, and B. R. Johnson, Quality, speed, and scale: Three key attributes to measure the performance of near-term quantum computers, arXiv:2110.14108.

[298] J. Wang, G. Guo, and Z. Shan, SoK: Benchmarking the performance of a quantum computer, Entropy **24**, 1467 (2022).

[299] S. Y.-C. Chen and S. Yoo, Federated quantum machine learning, Entropy **23**, 460 (2021).

[300] W. J. Yun, J. P. Kim, S. Jung, J. Park, M. Bennis, and J. Kim, Slimmable quantum federated learning, arXiv:2207.10221.

[301] M. T. West, M. Sevior, and M. Usman, Boosted ensembles of qubit and continuous variable quantum support vector machines for B meson flavor tagging, Adv. Quantum Technol. **6**, 2300130 (2023).

[302] M. Schuld and F. Petruccione, Quantum ensembles of quantum classifiers, Sci. Rep. **8**, 1 (2018).

[303] X. Wang, Y. Ma, M.-H. Hsieh, and M.-H. Yung, Quantum speedup in adaptive boosting of binary classification, Sci. China Phys., Mech. Astron. **64**, 220311 (2021).

[304] S. Arunachalam and R. Maity, in *International Conference on Machine Learning* (PMLR, 2020), p. 377.

[305] A. Izdebski and R. de Wolf, Improved quantum boosting, arXiv:2009.08360.

[306] A. Macaluso, L. Clissa, S. Lodi, and C. Sartori, Quantum ensemble for classification, arXiv:2007.01028.

[307] A. Abbas, M. Schuld, and F. Petruccione, On quantum ensembles of quantum classifiers, Quantum Mach. Intell. **2**, 1 (2020).

[308] Z. Zhao, J. K. Fitzsimons, and J. F. Fitzsimons, Quantum-assisted Gaussian process regression, Phys. Rev. A **99**, 052331 (2019).

[309] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, Phys. Rev. Lett. **100**, 160501 (2008).

[310] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, Phys. Rev. A **78**, 052310 (2008).

[311] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, On the robustness of bucket brigade quantum RAM, New J. Phys. **17**, 123010 (2015).

[312] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks, Nat. Comput. Sci. **3**, 542 (2023).

[313] B. T. Kiani, S. Lloyd, and R. Maity, Learning unitaries by gradient descent, arXiv:2001.11897.

[314] H.-Y. Huang, K. Bharti, and P. Rebentrost, Near-term quantum algorithms for linear systems of equations with regression loss functions, New J. Phys. **23**, 113021 (2021).

[315] B. Coyle, M. Henderson, J. C. J. Le, N. Kumar, M. Paini, and E. Kashefi, Quantum versus classical generative modelling in finance, Quantum Sci. Technol. **6**, 024013 (2021).

[316] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nat. Commun. **12**, 1 (2021).

[317] A. Hayashi, T. Hashimoto, and M. Horibe, Reexamination of optimal quantum state estimation of pure states, Phys. Rev. A **72**, 032325 (2005).

[318] Z. Holmes, A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, Barren Plateaus Preclude Learning Scramblers, Phys. Rev. Lett. **126**, 190501 (2021).

[319] K. Sharma, M. Cerezo, L. Cincio, and P. J. Coles, Trainability of Dissipative Perceptron-Based Quantum Neural Networks, Phys. Rev. Lett. **128**, 180505 (2022).

[320] C. O. Marrero, M. Kieferová, and N. Wiebe, Entanglement-Induced Barren Plateaus, PRX Quantum **2**, 040316 (2021).

[321] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 1 (2018).

[322] S. Altares-López, A. Ribeiro, and J. J. García-Ripoll, Automatic design of quantum feature maps, Quantum Sci. Technol. **6**, 045015 (2021).

[323] T. L. Patti, K. Najafi, X. Gao, and S. F. Yelin, Entanglement devised barren plateau mitigation, Phys. Rev. Res. **3**, 033090 (2021).

[324] A. Uvarov and J. D. Biamonte, On barren plateaus and cost function locality in variational quantum algorithms, J. Phys. A: Math. Theor. **54**, 245301 (2021).

[325] R. J. Garcia, C. Zhao, K. Bu, and A. Jaffe, Barren plateaus from learning scramblers with local cost functions, J. High Energy Phys. **2023**, 1 (2023).

[326] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, Nat. Commun. **12**, 1 (2021).

[327] C. Piveteau, D. Sutter, S. Bravyi, J. M. Gambetta, and K. Temme, Error Mitigation for Universal Gates on Encoded Qubits, Phys. Rev. Lett. **127**, 200505 (2021).

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

[329] C.-Y. Park and N. Killoran, Hamiltonian variational ansatz without barren plateaus, Quantum **8**, 1239 (2024).

[330] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[331] S. Hadfield, Z. Wang, B. O'gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[332] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, and S. Boixo *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, Nat. Phys. **17**, 332 (2021).

[333] F. C. Arute *et al.*, Hartree-Fock on a superconducting qubit quantum computer, Science **369**, 6507 (2020).

[334] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. De Jong, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2020), p. 267.

[335] 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, Nat. Commun. **5**, 1 (2014).

[336] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao, Quantum circuit architecture search for variational quantum algorithms, npj Quantum Inf. **8**, 62 (2022).

[337] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao, Toward trainability of quantum neural networks, arXiv:2011.06258.

[338] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Absence of Barren Plateaus in Quantum Convolutional Neural Networks, Phys. Rev. X **11**, 041011 (2021).

[339] K. Nakaji and N. Yamamoto, Expressibility of the alternating layered ansatz for quantum computation, Quantum **5**, 434 (2021).

[340] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, Quantum **3**, 214 (2019).

[341] V. Heyraud, Z. Li, K. Donatella, A. Le Boité, and C. Ciuti, Efficient Estimation of Trainability for Variational Quantum Circuits, PRX Quantum **4**, 040335 (2023).

[342] M. T. West, M. Sevior, and M. Usman, Reflection equivariant quantum neural networks for enhanced image classification, Mach. Learn.: Sci. Technol. **4**, 035027 (2023).

[343] Y. Kwak, W. J. Yun, S. Jung, and J. Kim, in *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)* (IEEE, 2021), p. 413.

[344] J. Bowles, S. Ahmed, and M. Schuld, Better than classical? The subtle art of benchmarking quantum machine learning models, arXiv:2403.07059.

[345] B. Jobst, K. Shen, C. A. Riofrío, E. Shishenina, and F. Pollmann, Efficient MPS representations and quantum circuits from the Fourier modes of classical image data, arXiv:2311.07666.

[346] Q. Li, Y. Huang, X. Hou, Y. Li, X. Wang, and A. Bayat, Ensemble-learning variational shallow-circuit quantum classifiers, arXiv:2301.12707.

[347] U. Dorner, R. Demkowicz-Dobrzanski, B. J. Smith, J. S. Lundeen, W. Wasilewski, K. Banaszek, and I. A. Walmsley, Optimal Quantum Phase Estimation, Phys. Rev. Lett. **102**, 040403 (2009).

[348] J. Cui and H. Fan, Correlations in the Grover search, J. Phys. A: Math. Theor. **43**, 045305 (2010).

[349] D. Biron, O. Biham, E. Biham, M. Grassl, and D. A. Lidar, in *Quantum Computing and Quantum Communications*, edited by C. P. Williams (Springer Berlin Heidelberg, Berlin, Heidelberg, 1999), p. 140.

[350] R. Blume-Kohout, Optimal, reliable estimation of quantum states, New J. Phys. **12**, 043034 (2010).

[351] A. Carollo, B. Spagnolo, A. A. Dubkov, and D. Valenti, On quantumness in multi-parameter quantum estimation, J. Stat. Mech.: Theory Exp. **2019**, 094010 (2019).

[352] C. W. Helstrom, Quantum detection and estimation theory, J. Stat. Phys. **1**, 231 (1969).

[353] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, Generalization in quantum machine learning from few training data, Nat. Commun. **13**, 1 (2022).

[354] Chatgpt, ChatGPT, https://chat.openai.com/chat (2023), accessed: June 5, 2023.

[355] M. G. Tehrani, E. Sultanow, W. J. Buchanan, M. Amir, A. Jeschke, M. Houmani, R. Chow, and M. Lemoudden, Stabilized quantum-enhanced SIEM architecture and speed-up through Hoeffding tree algorithms enable quantum cybersecurity analytics in botnet detection, Sci. Rep. **14**, 1732 (2024).

[356] W. M. Watkins, S. Y.-C. Chen, and S. Yoo, Quantum machine learning with differential privacy, Sci. Rep. **13**, 2453 (2023).

[357] M. Akter, H. Shahriar, A. Iqbal, M. Hossain, M. Karim, V. Clincy, and R. Voicu, in *2023 IEEE International Conference on Software Services Engineering (SSE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2023).

[358] S. Kundu and S. Ghosh, in *Proceedings of the Great Lakes Symposium on VLSI 2022* (Association for Computing Machinery, New York, NY, USA, 2022), p. 463.

[359] M. T. West, S. M. Erfani, C. Leckie, M. Sevior, L. C. Hollenberg, and M. Usman, Benchmarking adversarially robust quantum machine learning at scale, Phys. Rev. Res. **5**, 023186 (2023).

[360] M. R. Islam, M. U. Ahmed, S. Barua, and S. Begum, A systematic review of explainable artificial intelligence in terms of different application domains and tasks, Appl. Sci. **12**, 1353 (2022).

[361] A. Saranya and R. Subhashini, A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends, Decis. Anal. J. **7**, 100230 (2023).

[362] P. Steinmüller, T. Schulz, F. Graf, and D. Herr, Explainable AI for quantum machine learning, arXiv:2211.01441.

[363] R. Wang, F. Baba-Yara, and F. Chen, in *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)* (IEEE, 2024), p. 121.

[364] G. Aleksandrowicz *et al.*, Qiskit: An Open-Source Framework for Quantum Computing (2019).

[365] W. Li, S. Lu, and D.-L. Deng, Quantum federated learning through blind quantum computing, Sci. China Phys., Mech. Astron. **64**, 100312 (2021).

[366] A. S. Bhatia, S. Kais, and M. A. Alam, Federated quanvolutional neural network: A new paradigm for collaborative quantum learning, Quantum Sci. Technol. **8**, 045032 (2023).

[367] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, Nature **618**, 500 (2023).