Applied Physics Reviews REVIEW pubs.aip.org/aip/are

Quantum graph learning and algorithms applied in quantum computer sciences and image classification

Cite as: Appl. Phys. Rev. **12**, 021327 (2025); doi: 10.1063/5.0237599
Submitted: 5 September 2024 · Accepted: 29 May 2025 ·
Published Online: 12 June 2025

Farzaneh Shayeganfar,[1,2] Ali Ramazani,[3,a)] Veera Sundararaghavan,[1] and Yuhua Duan[4,a)]

AFFILIATIONS

[1]Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan 48109, USA
[2]Department of Physics and Energy Engineering, Amirkabir University of Technology, Tehran, Iran
[3]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
[4]National Energy Technology Laboratory, United States Department of Energy, 626 Cochran Mill Road, Pittsburgh, Pennsylvania 15236, USA

[a)]**Authors to whom correspondence should be addressed:** ramazani@mit.edu and Yuhua.duan@netl.doe.gov

ABSTRACT

Graph and network theory play a fundamental role in quantum computer sciences, including quantum information and computation. Random graphs and complex network theory are pivotal in predicting novel quantum phenomena, where entangled links are represented by edges. Quantum algorithms have been developed to enhance solutions for various network problems, giving rise to quantum graph computing and quantum graph learning (QGL). In this review, we explore graph theory and graph learning methods as powerful tools for quantum computers to generate efficient solutions to problems beyond the reach of classical systems. We delve into the development of quantum complex network theory and its applications in quantum computation, materials discovery, and research. We also discuss quantum machine learning (QML) methodologies for effective image classification using qubits, quantum gates, and quantum circuits. Additionally, the paper addresses the challenges of QGL and algorithms, emphasizing the steps needed to develop flexible QGL solvers. This review presents a comprehensive overview of the fields of QGL and QML, highlights recent advancements, and identifies opportunities for future research.

TABLE OF CONTENTS

I. INTRODUCTION

Quantum computing (QC) and quantum learning are rapidly expanding research fields that are creating new paradigms for solving complex quantum computational problems. The concepts of quantum phenomena such as entanglement and superposition were first introduced by pioneers like Richard Feynman[1] and Paul Benioff[2,3] in the early 1980s. Since then, both quantum algorithms and hardware have evolved significantly. Using graph theory to study complex phenomena in quantum mechanics represents a burgeoning area in mathematical physics, where physical phenomena are modeled through graph applications.

Data science and machine learning (ML), often considered the fourth pillar of scientific inquiry alongside experiment, theory, and simulation, are revolutionizing how we process and interpret graph-structured data. Techniques such as graph neural networks (GNNs) and geometric deep learning play crucial roles in this transformation.[4–6] Quantum graph learning (QGL) is an emerging field that promises to revolutionize graph learning by harnessing quantum computation and theory to encode graph data into quantum states, using qubits to overcome the limitations of traditional models, such as lack of interpretability and difficulty handling complex data.[7,8]

While implementing QGL on large-scale graphs presents challenges, it opens promising avenues for future quantum computing research. Quantum annealing (QA), for example, is a technique that leverages superposition and entanglement to explore vast solution spaces in parallel, potentially achieving significant speedups over classical computing methods. Despite current hardware limitations and coherence issues, ongoing research and advancements suggest that quantum annealing will play a vital role in developing new optimization techniques.

Graph theory algorithms stand out as one of the most promising applications of quantum computing due to their adjacency matrix structure, which is efficiently represented and manipulated by quantum circuits. This paper discusses two main categories of graph algorithms: graph coloring and techniques for solving differential equations using quantum annealing. These algorithms have demonstrated promising results and provide fertile ground for further research. However, challenges remain in scaling these algorithms for larger graphs and developing efficient methods for encoding graphs onto quantum circuits.

Quantum graph coloring is a promising method for solving graph coloring and other combinatorial optimization problems using quantum computing. Although quantum computers are still in their early stages, research in this area is growing, and quantum graph coloring is expected to become a standard tool for these problems in the future.

Recently, the integration of quantum computing with graph neural networks has led to the development of quantum graph neural network (QGNN) models for materials search. These quantum models require fewer trainable variables than classical models and achieve faster convergence during training.[9,10]

Image processing techniques and classification have had a profound impact on various scientific fields, including medical diagnosis and autonomous vehicles.[11–15] By extracting information and features from images, machines can interpret data using deep learning, neural networks, and quantum machine learning (QML) methods. Within the framework of QML, quantum image processing (QIP) and classification have become dominant practical applications.[16–18] The fields of quantum graph computing, learning, and image processing have been highly active in recent years, and we anticipate more exciting theoretical advancements and discoveries in the near future.

II. GRAPH THEORY ALGORITHM FOR QUANTUM COMPUTING

Graph theory algorithms have become a focal point in quantum computing due to their potential to tackle complex problems that are intractable for classical computers. Graph theory involves the study of graphs, which are mathematical structures composed of vertices (nodes) and edges (links). The vertices represent entities of interest, while the edges represent the relationships between them. Quantum computing offers a new paradigm for exploring graph theory algorithms, enabling the development of efficient solutions to computational challenges that classical methods struggle to address.

Graph theory algorithms for quantum computing can be broadly categorized into two main types: Hamiltonian encoding-based solvers and quantum random walk-based solvers. Hamiltonian encoding-based solvers involve mapping the graph onto a quantum Hamiltonian, which mathematically represents the energy levels of a quantum system.[19,20] The solution to the problem corresponds to the minimum energy state of the Hamiltonian. Quantum random walk-based solvers perform a quantum random walk on the graph, with the probability of visiting a vertex influenced by its degree and the edge weights.[21,22]

One of the most significant advantages of using quantum computing for graph theory algorithms is the potential for exponential speedup compared to classical algorithms. For instance, finding the chromatic number of a graph—the minimum number of colors needed to color the vertices without adjacent vertices sharing the same color—is a non-deterministic polynomial (NP)-hard problem in

classical computing. However, a quantum algorithm utilizing adiabatic evolution has been proposed to solve this problem in polynomial time on a quantum annealer. Another example is the maximum cut problem, which involves partitioning the graph's vertices into two sets to maximize the number of edges between them. While this is also an NP-hard problem, a quantum algorithm based on Grover's algorithm offers a quadratic speedup over classical solutions.

Graph theory algorithms for quantum computing have the potential to revolutionize computational science. The ability to solve complex problems with exponential speedup can profoundly impact diverse fields, including optimization and machine learning. Despite challenges such as error correction and scalability, the development of quantum graph theory algorithms is a promising area of research poised to achieve significant advances in the coming years. To develop universal quantum computing, adiabatic quantum computing (AQC) emerges as one approach, which implements an areal process such as quantum annealing to control the continuous-time evolution at finite temperature and capture the relaxation of the adiabatic conditions.

A. Quantum annealing

Quantum annealing (QA) is a powerful quantum computing approach for solving combinatorial optimization problems, which often become computationally infeasible for classical computers when scaled to larger sizes. The fundamental principle of quantum annealing leverages the quantum mechanics concepts of superposition and entanglement to explore and optimize a solution space more efficiently than classical methods.[23–25]

QA is rooted in the adiabatic theorem, which asserts that if a physical system evolves slowly enough, it will remain in its ground state. During the quantum annealing process, the system begins with a simple initial Hamiltonian and transitions to a final Hamiltonian that encodes the problem to be solved. By gradually transforming the Hamiltonian from the initial to the final state, the system can be maintained in its ground state, representing the optimal solution to the problem, provided the evolution is sufficiently slow (Fig. 1).[23–26]

Quantum annealing is a type of quantum computation that utilizes quantum fluctuations to find the global minimum of an optimization problem. It has been demonstrated that quantum annealing can address a wide array of optimization problems, including those in combinatorial optimization and machine learning. The core principle of quantum annealing is encoding the optimization problem into a Hamiltonian that describes a system of interacting quantum spins. The Hamiltonian is then gradually evolved from an easily prepared initial state to a final state that represents the solution to the optimization problem.[27–29]

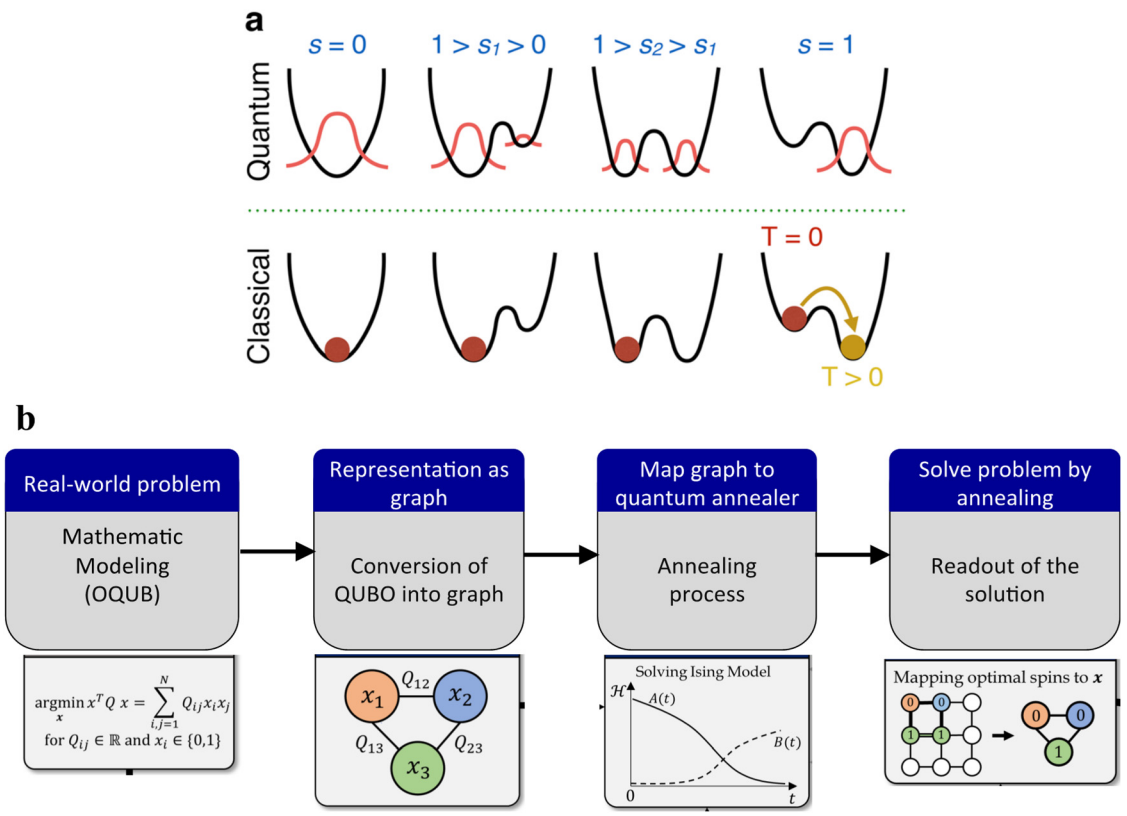

FIG. 1. (a) Quantum adiabatic annealing (QA) vs classical annealing. The system for classical annealing by thermal fluctuations (temperature $T > 0$) overcomes barriers, while for quantum annealing, to arrive at the ground state, the system state (red) tunnels through a changing barrier (black).[32] (b) The workflow of a typical quantum algorithm using a quantum annealer. Reproduced with permission from Li *et al.*, npj Quantum Inf. **4**(1), 14 (2018). Copyright 2018 Authors, licensed under a Creative Commons Attribution (CC BY 4.0) License.

Maintaining the system in its ground state during quantum annealing requires the Hamiltonian's dynamics to change at a pace dictated by the minimum energy gap between the ground state and the first excited state.[26,30] However, in practice, adhering to the adiabaticity condition can be challenging due to background noise and thermal fluctuations in open quantum systems, which can cause the system to deviate from its ground state. The presence of low-energy states and gaps in the spectrum of the time-dependent Hamiltonian H(t) can indicate the system's sensitivity to these changes.[31]

To address these challenges, relaxation of adiabatic quantum computing (AQC) heuristically determines the annealing time required to transition from the initial Hamiltonian (H_i) to the final Hamiltonian (H_f). A heuristic optimization algorithm is employed instead of a deterministic quantum algorithm, maintaining a nonzero probability of staying in the ground state throughout the annealing or total time evolution.[33] The AQC algorithms start by preparing the system in the ground state of H_i, which slowly evolves, decreasing over time, while the influence of H_f increases with the parameter t

$$\mathrm{H(t)} = A(t)\mathrm{H_i} + B(t)\mathrm{H_f}, \tag{1}$$

here, time t varies between zero and annealing time (T_a), and the monotonic functions $A(t)$ and $B(t)$ represent the annealing schedule, with initial conditions of $A(0) = 1$, $B(0) = 0$ and $A(T_a) = 0$, $B(T_a) = 1$. Transition from H_i to H_f evolves the initial ground state $|\Psi(t = 0)\rangle$ to the final ground state $|\Psi(t = T_a)\rangle$ of the H_f.[33,34] Usually, the H_i is a transverse field in the x-direction, defined as

$$H_i = \sum\nolimits_{i \in V} \sigma_i^x, \tag{2}$$

here, σ_i^x represents the x-Pauli matrix, which acts on the ith qubit; for instance, the eigenstates of σ^x is the superposition of the eigenstates of σ^z namely, $|x \pm\rangle = \frac{1}{\sqrt{2}}(|+\rangle \pm |-\rangle)$, where σ^x acts on eigenstates of σ^z ($|\pm\rangle$); flipping the eigenstates of σ^z; ($\sigma^x |\pm\rangle = |\mp\rangle$). Also, in Eq. (2), V indicates the lattice sites of qubits and is the set of vertices of the graph G(V, E). The system is placed in the ground state of H_i, which is $|\Psi(t = 0)\rangle = |x\text{-}, \ldots, x\text{-}\rangle$ and evolves to H_f

$$H_f = \sum\nolimits_{i \in V} h_i \sigma_i^z + \sum\nolimits_{(I,J) \in E} J_{ij} \sigma_i^z \sigma_j^z, \tag{3}$$

where E represents the set of edges of the graph G(V, E), indicating the connections between qubits. The parameter J_{ij} denotes the symmetric interaction strength between connected qubits i and j namely, $J_{ij} = J_{ji}$, while h_i represents the local field or on-site energy of qubit i. As discussed extensively in the literature, the Hamiltonian H(t) at any given time is expressed through this framework.[33,35]

One of the primary advantages of quantum annealing is its capability to explore vast solution spaces in parallel. Classical computing often struggles with the exploration of large solution spaces, making the process extremely time-consuming and computationally expensive. Quantum annealing, leveraging the principles of superposition and entanglement, can simultaneously explore multiple potential solutions. This capability has the potential to dramatically accelerate the discovery of optimal solutions for a wide range of optimization problems.[24,25]

QA's potential applications span various fields, including finance, cryptography, drug discovery, and logistics. In machine learning, quantum annealing has been employed to optimize the training of deep neural networks. In drug discovery, it aids in the optimization of designing new drug compounds, a process that is traditionally both time-consuming and costly when approached with classical methods.[25]

Despite its promising advantages, quantum annealing faces several limitations. A major challenge is maintaining coherence, which is crucial for the effective functioning of the quantum annealing process. Coherence refers to a quantum system's ability to sustain its quantum state over time, a condition that is often difficult to maintain in practical systems. Moreover, current hardware limitations constrain the size and complexity of problems that can be addressed through QA.[23–25] Standard approaches for data preprocessing and NP-hard graph problems such as graph partitioning and graph coloring are based on implementing quantum computing on graph data.

B. Quantum graph coloring method

Graph coloring is a fundamental problem in computer science, combinatorial optimization, and operations research, with applications ranging from scheduling and map coloring to register allocation and frequency assignment. The basic graph coloring problem is to assign colors to the vertices of a graph so that no two adjacent vertices share the same color, using the fewest possible colors.[36,37] Quantum computing offers a promising approach for addressing graph coloring problems, leveraging its inherent parallelism to accelerate the computation process.[36,37]

In general, the graph coloring problem is posed in terms of minimizing the sum of two cost functions: the first is a data cost of assigning a particular color to a pixel, and the other is a smoothing cost of assigning different colors to adjacent pixels. Using this generalization, a novel approach based on Potts energy formulation has been developed for materials mesh generation, where a generic finite element mesh of a representative volume element can be labeled using experimental data.[38] Potts energy optimizes the dual of the finite element mesh by decomposing it into field energy cost, which represents the likelihood of a grain label on the experimental voxel element data, and the interaction energy cost, which encodes a prior on this labeling and smooths the phase boundary.[38]

The graph-cuts method is used for energy minimization of multiscale (macro/micro scale) design of structural metallic materials used in the turbine blades of the aerospace industry (as depicted in Fig. 2), which this optimization is critical for tailoring the mechanical properties of materials.[39,40] For instance, stress variation at the macroscopic and microscopic scales of turbine blades necessitates microstructural mesh generation (3D structure meshing), capturing grain size, boundaries, and shape features, which play vital roles in fracture and localization processes.[41–44]

Various methods for quantum graph coloring exist, including the adiabatic quantum algorithm and the quantum walk algorithm. The adiabatic quantum algorithm relies on the adiabatic theorem of quantum mechanics, which states that a quantum system remains in its ground state if it evolves slowly enough. This algorithm maps the graph coloring problem onto an Ising Hamiltonian, gradually transitioning from the initial Hamiltonian to the final one, where the final Hamiltonian represents the graph coloring solution.[45,46]

The quantum walk algorithm, on the other hand, is based on quantum random walks, a quantum counterpart of classical random walks. This algorithm employs quantum walks to traverse the graph and assign colors to vertices. In certain cases, the quantum walk

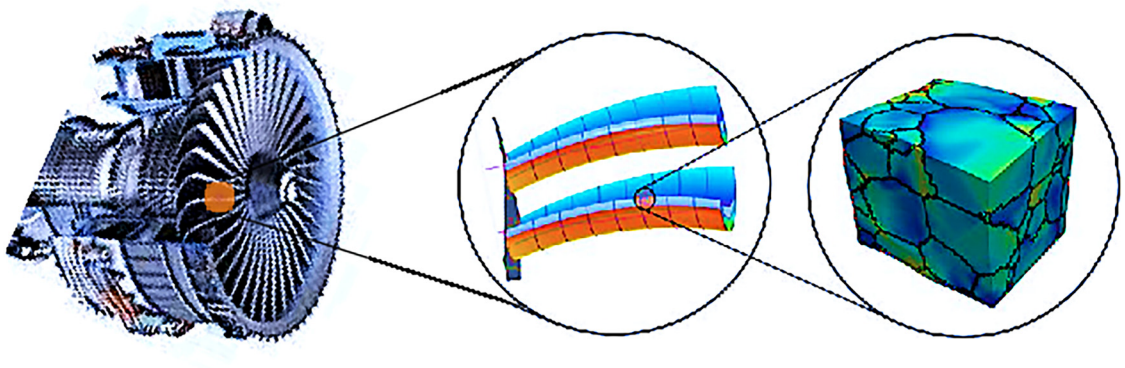

FIG. 2. Multiscale (macroscale and microscale) design of stress variation of turbine blades for aircraft engines.[38] Reproduced with authors' permission from S. Srivastava, "Graph theoretic algorithms adaptable to quantum computing," Ph.D. thesis (University of Michigan, 2021).

algorithm has demonstrated superior performance to classical graph coloring algorithms.[46]

Quantum graph coloring offers several advantages over classical methods. It can find exact solutions to graph coloring problems more quickly and solve more complex problems than classical algorithms. Moreover, quantum graph coloring can be applied to other combinatorial optimization problems, such as vertex cover, maximum independent set, and minimum dominating set.[46] Physical phenomena models such as deflection in elastic bars and heat flow include differential equations, which can be solved by quantum graph techniques based on graph coloring and annealing methods, as discussed in Sec. II C.

C. Graph techniques for solving differential equations on a quantum annealer

Differential equations, which describe the relationship between a function and its derivatives, are fundamental in fields such as physics, engineering, and other sciences. Solving these equations is often computationally intensive, particularly for complex problems. Quantum annealing offers a promising approach to solving differential equations more efficiently. Here, we focus on reviews of graph techniques for solving differential equations on a quantum annealer.[27–29] A quantum annealer is a specialized device or a type of designed quantum computer tailored to solve complex optimization problems in a similar simulated process of annealing in classical computing. It is supposed to find the global minimum/maximum of the cost/lost function by efficient solution space due to importing quantum mechanics.

Graph techniques can be effectively employed to solve differential equations on a quantum annealer. This approach involves discretizing the differential equation on a graph, where nodes represent unknowns and edges denote the connections between them. The resulting system of linear equations is then mapped onto the Hamiltonian of a quantum annealer. By annealing the quantum system and measuring the expectation values of the observables corresponding to the graph nodes, the solution to the differential equation can be obtained.[47,48]

Several algorithms have been proposed for solving differential equations using graph techniques on a quantum annealer. One promising approach is the adiabatic evolution method, which leverages the adiabatic theorem of quantum mechanics to ensure the system remains in the ground state throughout the annealing process. Another approach is the spectral method, which employs the eigenvalues and eigenvectors of the graph Laplacian matrix to solve the differential equation.[49]

To model some physical phenomena, such as deflection in elastic bars and steady-state analysis of heat flow, second-order differential equations are ubiquitous. Srivastava and Sundararaghavan[50] reformulated a finite element model in the form of an Ising Hamiltonian by using a quantum annealer to solve these differential equations. Revealing complications in the discrete variables in the Ising model by defining differential quantities leads to a graph coloring based methodology. This method introduce a "box algorithm," which searches iteratively for solutions in a subspace of weak solutions defined over a graph.[50] The box algorithm by solving the truss mechanics problem is demonstrated on the D-Wave quantum computer.[50]

An energy minimization formulation is also applied for solving certain differential equations. Quantum annealers map this minimization strategy to an Ising Hamiltonian. In this context, finite elements can be used with sparse graphs to develop Ising Hamiltonians, facilitating the embedding of larger problems on noisy intermediate-scale quantum (NISQ)-era hardware.[50] Figure 3 illustrates this procedure, demonstrating how a discretized differential equation can be solved using energy minimization on a graph. Additionally, qubits must encode rational numbers for representing real-valued functions, such as Ising lattice points, which carry up/down spins as two discrete levels in the ground states.[38] While encoding similar binary (0/1) data in classical computers requires significant memory, current quantum annealers have a limited number of physical qubits, making double-precision representation costly compared to classical methods.[50]

Despite promising results obtained using graph techniques to solve differential equations on a quantum annealer, several challenges remain. A primary challenge is the limited size of current quantum annealing devices, which constrains the size of usable graphs. Furthermore, the accuracy of solutions obtained through quantum annealing must be improved, particularly for problems requiring high precision.[48]

Graph techniques for solving differential equations on a quantum annealer hold the potential for significant speedup over classical methods. However, further research is essential to overcome existing limitations and develop more efficient and accurate algorithms.

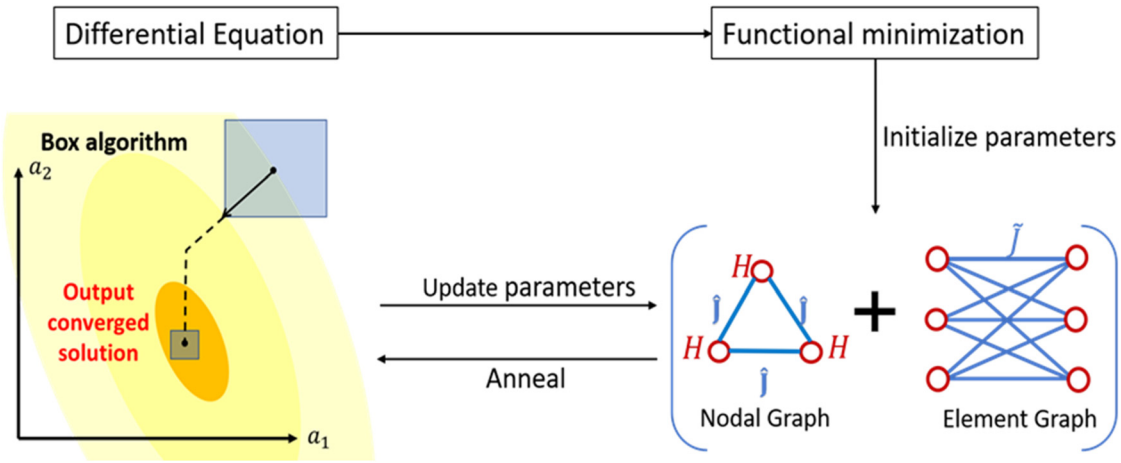

FIG. 3. Illustration of procedure for solving differential equation.[50] Reproduced with permission from Srivastava and Sundararaghavan, Phys. Rev. A **99**(5), 052355 (2019). Copyright 2019 American Physical Society.

III. GRAPH LEARNING AND QUANTUM MECHANICS

The integration of quantum theory with graph learning creates a powerful analytical tool for processing and interpreting graph data. High-performance quantum computing introduces a new paradigm for graph learning by renewing traditional methods and providing enhanced capabilities for accessing, storing, and processing complex graph-structured data. This synergy between quantum theory and graph learning effectively addresses many bottlenecks inherent in traditional graph learning methods. In the following, the characteristics of graph-structured data for graph data processing and access for graph learning will be discussed.

A. The characteristics of graph-structured data

The development of graph learning is motivated by the complexity inherent in graph computing and the limitations of traditional computing resources in handling graph-structured data.[51,52] Three traditional data structures are commonly used to represent graph-structured data: adjacency lists, adjacency matrices, and incidence matrices. An adjacency list uses a linked table to represent graph data, while a two-dimensional adjacency matrix indicates the existence of edges between specific nodes. In contrast, an incidence matrix captures relationships between each node and every edge. Despite advancements in storage solutions, handling graph-structured data presents significant challenges in both academia and industry.[53] While tools like sparse matrices in PyTorch and dense_hash_map in Google's SparseHash facilitate graph structure handling, managing graph dynamics remains challenging compared to other data structures.[54] Efficient space and speed are critical for handling graph data, necessitating a smaller footprint to store significant amounts of data.

B. The bottleneck of graph learning

Graph learning, despite its significant progress, faces several bottlenecks. Aggregating information over long paths poses a challenge, as data growth must be compressed into fixed-size vectors.[55] Consequently, information from distant nodes cannot be propagated effectively when tasks rely on remote interactions. Although some graph learning methods can integrate global information, they often require compensatory approaches.[56] Explainability and interpretability are additional bottlenecks; while interpretable models (white box) provide human-understandable explanations, black box models rely on *post hoc* explanation techniques.[57] Quantum theory offers solutions to these issues, addressing both explainability and interpretability. Another bottleneck is the limited ability of graph learning to provide practical solutions for large-scale graphs, such as brain networks.

C. The reciprocity of quantum theory and graph learning

Quantum theory underpins quantum computing, a new computational paradigm with potential superiority over traditional methods due to its exponential growth capabilities. Quantum computers excel in tasks like integer factorization and discrete logarithms[58] and outperform traditional methods in searching and retrieving values from unstructured databases.[59] Quantum theory's nonlocal effects, such as quantum teleportation, allow transmission of quantum states over vast distances with minimal communication costs.[60] Similarly, global information in graph-structured data can be transferred via short paths.[61,62] These advantages drive researchers to merge quantum computing with graph learning, given the critical role of global information in graph learning due to the non-Euclidean data structure.[63] However, current graph learning techniques, such as message-passing, struggle to integrate global information precisely at low computational costs. QGL leverages coherence and entanglement to facilitate fast, long-distance information transfer.

QGL offers potential solutions to large-scale problems with quantum-friendly hardware, where greater information capacity via qubit encoding reduces computational requirements. Employing quantum theory resolves issues in black box graph learning models, providing a theoretical basis for large-scale graph learning with real-world applications such as cognitive functions. This enhances the accuracy of downstream tasks in quantum computing and graph representation learning.

Quantum theory has demonstrated significant potential to reform and enhance graph learning. Its integration with graph learning methods enhances interpretability and transparency, improving graph data storage, access, and processing. The mutual relationship between quantum theory and graph learning offers a promising future for tackling complex graph problems.

IV. A TAXONOMY OF QUANTUM GRAPH LEARNING

A. Quantum data embedding and encoding schemes

A quantum data encoding (embedding) plays a crucial role in designing quantum algorithms as well as computational powers. To quantum encode data, we must convert classical data to quantum states in a Hilbert space upon a quantum feature map. To transform input data in a different space, where processing data is easier, a feature map is implemented. In a quantum feature map, quantum states act as the feature vectors in a Hilbert space as a feature space. Therefore, a set of classical data points will translate into a quantum state $|\psi_x\rangle$; as a set of gate parameters in a quantum circuit. There are various embedding techniques, some of which are basis embedding, amplitude encoding,[64] angle, and superposition encoding.[65] Some brief details can be found in Sec. VI A.

In this review, we categorize QGL methods into three distinct groups: quantum computing on graphs, quantum graph representation, and quantum circuits for graph neural networks (GNNs), as summarized in Table I. Most of these architecture models for quantum graph learning employ topology embedding, incorporating structural information into the quantum graph representation. A topological space embedding maps a homeomorphism onto its image by applying a continuous function. Embedding of a graph in a topological graph theory represents the association of homomorphic images (simple arcs) and vertices with points on the surface as a graph. The input data can be quantum (Q), classical (C), or synthetic (Syn.), which means the data processing is based on quantum computers, classical computers, and quantum computers assisted by classical computing

TABLE I. Comparison of quantum graph learning for different architecture models and their application. The input data can be quantum (Q), classical (C), or synthetic (Syn.).

Category (type)	Method	Input	Layer	Readout	Application
Quantum computing on graphs	Space-efficient quantum optimization (SEQO)[66]	Q	Q	Estimation	Graph coloring
	Quantum annealing-based decomposition (QAD)[67]	Q	Q and C	Estimation	Maximum clique; vertex cover
	Quantum annealing with integer slack variables (QAISV)[68]	Q	Q and C	Tomography	Graph partitioning
Quantum graph representation	Quantum-based subgraph convolutional NN (QS-CNN)[69]	C	Q and C	No tomography	Node classification
	Quantum graph recurrent embedding (QGRE)[70]	Q	Q and C	No tomography	Classification
	Quantum spatial graph convolutional NN (QSGCNN)[71]	C	Q and C	No tomography	Graph classification
	Quantum superposition-based graph kernel (QSGK)[20]	C	Q and C	Tomography	Graph classification
Quantum kernel based	Quantum Jensen–Shannon graph kernel (QJSK)[72]	C	Q and C	Tomography	Graph classification
	Gaussian boson samplers kernel (GBSK)[73]	C	Q and C	Estimation	Graph classification
	Specific feature graph kernel (SFGK)[20]	C	Q and C	Swap test	Graph classification
	Quantum evolution kernel (QEK)[19]	C	Q and C	Tomography	Graph classification
Quantum circuit for graph neural networks	Dynamic quantum graph NN (DQGNN)[74]	Q	Q and C	Tomography	Graph classification
	Quantum graph convolutional layer (QGCL)[75]	Syn.	Q	Tomography	Node classification
	Equivariant quantum graph circuit (EQGC)[76]	Q	Q and C	Estimation	Graph classification
	Quantum graph neural network (QGNN)[9]	Q	Q	Tomography	Graph classification
	Quantum graph convolution network (QGCN)[77]	Syn.	Q and C	Estimation	Graph classification
	Quantum neural network (QNN)[78]	Q	Q	Estimation	Network embedding
Hybrid deep	Hierarchical neural network based on QRWs (QWNN)[79]	C	Q and C	Tomography	Node classification
	Quantum subgraph graph convolutional NN (QSGCNN)[71]	C	Q and C	Tomography	Graph classification
	Quantum subgraph convolutional NN (QS-CNN)[69]	C	Q and C	Tomography	Node classification
	Quantum graph convolutional NN (QGCNN)[80]	C	Q and C	Estimation	Graph classification
	Hybrid quantum graph NN (HQGNN)[81]	C	Q and C	Estimation	Link prediction

modules, respectively. As shown in Table I, some models use the classical or synthetic inputs, which introduce classical layers to preprocess the graph data, which are inevitable for assisting the quantum computer to update the model parameters. To transform the quantum information into the classical expression, several readout operations have been implemented, such as estimation of the probability outcomes and swap test, which requires a small number of measurements. However, the tomography needs an exponentially large number of measurements.

Most of the quantum graph learning methods, as represented in Table I, apply the quantum circuit model to handle graph data, where the size of current quantum devices is insufficient to process the entire graph data as the input. To resolve this limitation, a Graph Quantum Neural Tangent Kernel (GraphQNTK)[82] has been developed by Tang and Yan based on quantum kernel graph classification of an infinite-width GNN with attention. However, their model cannot map Euclidean data into a quantum Hilbert space besides lacking of graph isomorphism classification theoretical proof. Hancock *et al.*[83] constructed a scale-free graph-structured data by combining classical and quantum machine learning methods based on QGNN, which called egoQGNN. This method can handle real-world datasets and apply to graph isomorphism with hierarchy architecture.[83]

B. Quantum computing on graphs

One of the standard approaches for addressing graph-theoretical problems and data preprocessing is the application of quantum computing to graph data, particularly in tackling NP-hard problems such as vertex cover, graph coloring, and graph partitioning. To capture the topological information of graphs, quantum evolution is typically performed on the underlying graph structure.[84]

Many NP-hard problems related to graphs can be formulated as Quadratic Unconstrained Binary Optimization (QUBO) problems, which serve as input for algorithms like the Quantum Approximate Optimization Algorithm (QAOA) and Quantum Annealing (QA).[85] Figure 4 illustrates examples of quantum computing-based algorithms used to solve NP-hard graph problems.[66] In QUBO problems, the objective is to minimize the cost function "f," as defined in the following equation:

$$\begin{aligned} &minimize\, f(x) = \sum_{i,j=1}^{N} S_{i,j} x_i x_j, \\ &y = argmin f(x)\, x \in \{0,1\}^N, \end{aligned} \tag{4}$$

where f(x) denotes the cost function, SS is a real symmetric matrix, and "y" indicates a global minimizer of $f(x)$. Solving QUBO problems, as explained above, involves finding the minimum energy of the Ising Hamiltonian of an N-qubit system, which considers the dynamics of quantum systems. The Hamiltonian is defined as follows:

$$H = \sum_{i,j=1}^{N} S_{i,j}(1 - M_i)(1 - M_j). \tag{5}$$

Here, M_z is the Pauli operator acting as the Pauli-M gate on the zth qubit. NP-hard graph problems have been formulated into QUBO problems, making them solvable using quantum computing. For

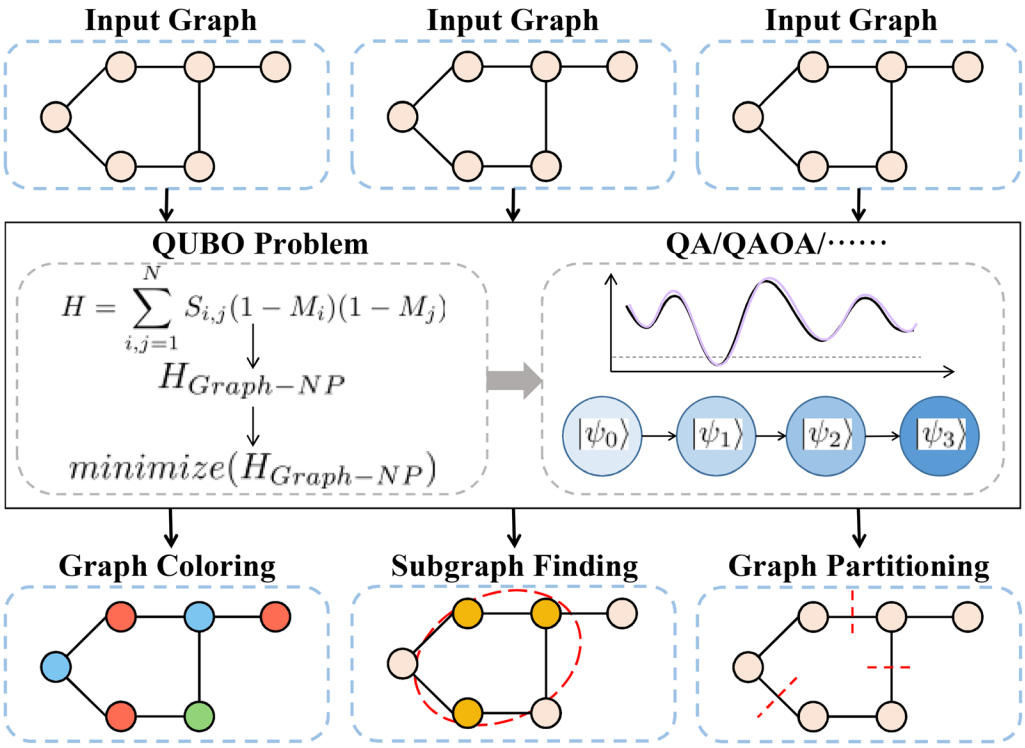

FIG. 4. Representation of quantum computing applied to graph problems.[8] Reproduced with authors' permission from Yu *et al.*, arXiv:2302.00892 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) License.

instance, Tabi *et al.*[66] tackled the graph coloring problem by implementing a space-efficient quantum optimization algorithm (SEQO) based on quantum annealing (QA). They transformed the graph coloring problem into a QUBO problem, which was then used as input for the Quantum Approximate Optimization Algorithm (QAOA). This approach developed a flexible framework that reduces the number of qubits needed, providing a space-efficient solution for graph coloring.

More recently, Pelofske *et al.*[67] addressed the maximum clique and minimum vertex cover problems by developing a QA-based decomposition algorithm (QAD). This algorithm recursively splits the problem into smaller subproblems, which can be effectively solved using QA. Their primary objective was to minimize the Ising model to obtain an optimal set of vertices.

Wang *et al.*[68] proposed a model based on quantum annealing with integer slack variables (QAISV) to handle inequality constraints in the grid partitioning optimization model, viewed as a type of graph. To convert this problem into a QUBO problem, they employed integer slack and binary expansion methods. Furthermore, Tang *et al.*[84] utilized quantum evolution through QUBO to solve graph isomorphism, another challenging NP-hard graph problem.

C. Quantum graph representation

Graph representation has emerged as a powerful analytical tool in quantum computing, enabling various downstream tasks such as link prediction and combinatorial optimization by mapping graphs into an embedding vector space. The significance of quantum graph representation algorithms lies in their ability to extract atypical or unseen patterns in graphs.[86] The core concept involves embedding graphs into quantum states, where superposition and entanglement can effectively characterize graph features in a Hilbert space.[36,84] Two representative approaches are quantum random walks and quantum graph kernels, illustrated in Fig. 4, which will be discussed in detail in Secs. IV C 1 and IV C 2.

1. Quantum random walks

In quantum random walks, the amplitudes of quantum states determine the initial distribution of the walker, allowing it to evolve in a quantum-mechanical manner. A graph G(V, E) consists of a vertex set V and an edge set E, where the basic state of the walker at vertex $v \in V$ is η_v in the Hilbert space H. The quantum state of the walker $|\psi(t)\rangle$ at time t is given by a linear combination of η at time t

$$|\psi(t)\rangle = \sum\nolimits_{v \in V} \alpha_v \eta_v, \tag{6}$$

where α_v is the complex amplitude. Quantum random walks are categorized into two types: discrete-time and continuous-time.[69] In discrete-time (or coin) quantum random walks, two Hilbert spaces are defined: the position Hilbert space H_p and the coin Hilbert space H_c, such that $H = H_p \otimes H_c$. H_p captures the superposition of nodes, while H_c captures the multi-directional superposition of the walker on each node. In contrast, continuous-time quantum random walks consider only the position Hilbert space H_p.

Quantum random walks represent the topological information of graphs in quantum information in real or complex space as a collection of nodes and correlation between nodes as edges, attracting significant research interest due to their ability to capture different patterns of vertex connectivity through destructive and constructive interference. In fact, the quantum walker characterizes its time evolution by the amplitudes of quantum states of its initial distribution. Adopting quantum random walks for graph decomposition introduced by Zhang *et al.*,[69] which called a quantum-based subgraph convolutional neural network (QS-CNN).

2. Quantum graph kernels

Graph kernels create a feature space within the Hilbert space of quantum states, giving rise to quantum graph kernel-based graph representation methods. The core idea is to represent different graphs in a Hilbert space and compare their similarity based on quantum representations.[70,73] To measure graph similarity, an inner product functions as a quantum graph kernel. A kernel K measures the similarity between two graphs, G_1 and G_2 in a Hilbert space H

$$K(G_1, G_2) = \langle \varphi(G_1) \,|\, \varphi(G_2) \rangle. \tag{7}$$

The field of quantum graph kernel-based graph representation has seen an increasing number of studies. For instance, to characterize graphs, Henry *et al.*[19] proposed a quantum evolution kernel (QEK), where Hamiltonian encoding-based quantum evolution applies for the realization of a graph kernel. Similarly, a quantum superposition-based graph kernel (QSGK) by Kishi *et al.*[20] implemented to measure subgraph similarity and extract features. They concentrated on graph classification by mapping many subgraphs into a quantum state in the Hilbert space, accomplishing high performance in subsequent tasks. A schematic of quantum graph learning is shown in Fig. 5.

D. General concept of classical graph neural networks

To develop quantum graph neural networks (QGNNs), it is essential to start with an understanding of classical graph neural networks (GNNs) as discussed in references.[87–89] GNNs are used to represent complex systems of interacting elements, such as identifying phase transitions in many-body physics, compressing many-body systems, and analyzing social networks. Graphs serve as structured data, where a pair $\vec{G} = (V, \vec{E})$ is called a simple directed graph (digraph). Here, V is a set of nodes, and E is a finite set of edges that connect pairs of nodes, defined as $\vec{E} \subseteq \{(v, w) : v, w \in V, v \neq w\}$.

Graphs are employed to model relational structures such as proteins, users, and webpages. In these models, nodes represent entities (e.g., proteins, users, and webpages), and edges represent interactions or relationships between them (e.g., molecular bonds, friendships, and hyperlinks). Node information is represented as a real-valued feature vector, denoted by matrix $X \in R^{N \times C}$ for a graph with $N = |V|$ nodes. The row vector x_u in X corresponds to the C-dimensional feature vector of node u. The adjacency matrix $A \in R^{N \times N}$ defines the graph's connectivity, where $a_{uv} = 1$auv $= 1$ if there is an edge between nodes u and v; otherwise, $a_{uv} = 0$.

GNNs are a category of machine learning models that act on graph structures (X, A), with permutation equivariance being a defining property. If $P \in \{0, 1\}^{N \times N}$ is a permutation matrix, a GNN layer function F(X, A) is permutation equivariant if:[90]

$$F(PX, PAP^T) = PF(X, A). \tag{8}$$

Permutation equivariance ensures that the GNN output is invariant to node reordering, serving as a desirable inductive bias for graph representation learning.

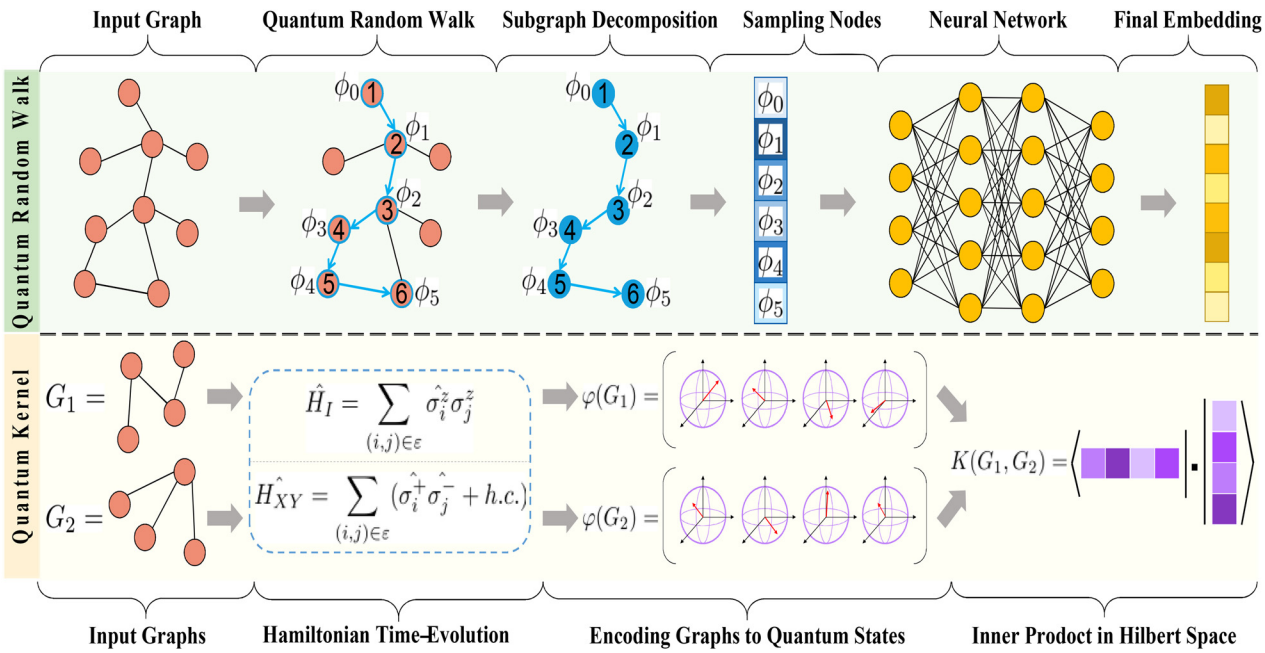

FIG. 5. Representation of quantum graph learning with quantum kernels and random walks.[8] Reproduced with authors' permission from Yu *et al.*, arXiv:2302.00892 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) License.

This property stems from the unordered nature of graphs, defined by connectivity rather than node ordering. In each GNN layer, nodes update features by aggregating information from local neighborhoods, defined as $N_u = \{v|(u, v) \in E \text{ or } (v, u) \in E\}$. Mathematically, a GNN layer, given input features XX, computes a new feature matrix $H \in R^{N\times C'}$ as[90]

$$H = F(X, A) = [\phi(x_1, X_{\mathcal{N}_1}), \phi(x_2, X_{\mathcal{N}_2}), \ldots, \phi(x_N, X_{\mathcal{N}_N})]^T, \quad (9)$$

where $X_{\mathcal{N}_u} = x_v \,|\, v \in \mathcal{N}_u$ is the multiset of neighborhood features, and ϕ is the local function, known as the message-passing or neighborhood aggregation function. The function ϕ depends on local neighbors and is shared across all nodes. I ϕ is permutation invariant in $X_{\mathcal{N}_u}$, then F will be permutation equivariant. Multiple GNN layers can be stacked to propagate information over longer distances, allowing the network to capture high-order interaction effects. Despite the simplicity of local neighborhood aggregation, many choices exist for the aggregation function ϕ.[90]

GNNs, a rapidly expanding field of deep learning, include several "flavors" such as convolutional, attentional, and message-passing (Fig. 6), which determine how the local learning ϕ transforms neighborhood features and interactions across the graph at varying complexity levels. In the convolutional flavor, neighboring node features are directly combined with fixed weights:[91]

$$h_u = \phi\left(x_u, \oplus_{v\in\mathcal{N}_u} c_{uv}\psi x_v\right). \quad (10)$$

Here, is the aggregation operator (often summation), and c_{uv} is a constant indicating the significance of node v to node u's representation ϕ and ψ are learnable transformations defined as

$$\phi(x, z) = Wx + Uz + b, \quad \psi(x) = Wx + b. \quad (11)$$

Classical GNNs model various graph-structured data tasks, including node classification, link prediction, and graph classification.[88,92] Node classification assigns labels to nodes based on attributes and graph structure, such as classifying protein functions in a protein–protein interaction network,[93] or categorizing social network users based on their connections and profile information. Link prediction determines the likelihood of an edge between two nodes. For graph classification, the objective is to classify entire graphs based on attributes and structures, such as classifying molecules by their quantum-mechanical properties.[94] Despite their success, classical GNNs face scalability challenges. Quantum computing offers solutions to these challenges, leading to the development of quantum neural network (QNN) architectures related to graph convolutional networks (GCNs). In Secs. V–VIII, we will devise and analyze QNN architectures corresponding to the three major types of classical GNNs (convolutional, attentional, and message-passing): quantum graph convolutional networks, quantum graph attention networks, and quantum message-passing GNNs. These architectures fall under the research area of quantum graph neural networks.

E. Quantum graph convolutional networks

The quantum algorithm for node classification using graph convolutional networks (GCNs) begins by redefining some key notations. Consider a graph G = (V, E) where V is the set of nodes, E is the set of edges, and the adjacency matrix is $A \in R^{N\times N}$ with N being the total number of nodes. The node attributes are represented by $X \in R^{N\times C}$, where C denotes the number of features for each node. The node representation at layer l is given b $H^{(l)} \in R^{N\times F_l}$, where F_l denotes the dimension of the node representation for each node. In a graph

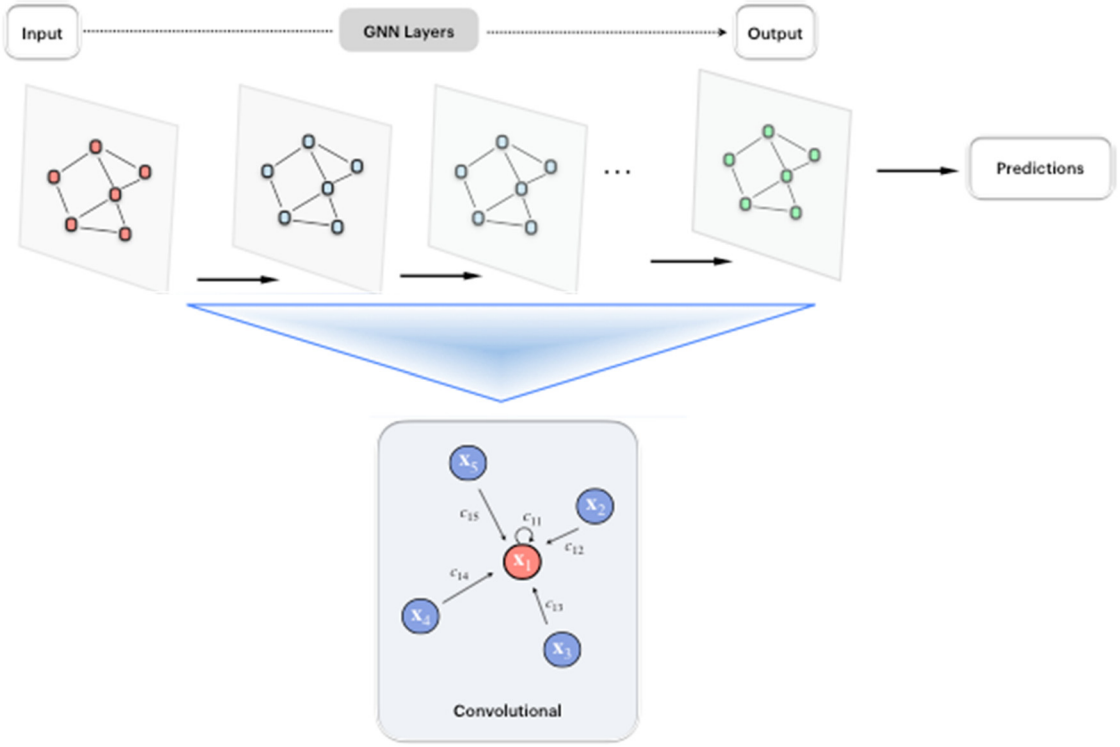

FIG. 6. Pipeline of GNN and convolutional GNN layers.[90] Permutation equivariant functions F(X,A) create GNN architectures by applying shared permutation invariant functions ϕ over local neighborhoods. This local function ϕ, often related to "diffusion" or "message-passing," is part of the overall computation of a "GNN layer." It transforms neighborhood features, enabling interaction modeling across the graph.[90] Reproduced from Liao *et al.*, arXiv:2405.17060v1 (2024). Copyright 2024 Authors, licensed under a Creative Commons Attribution 4.0 International (CC BY-NC-ND 4.0) License.

convolutional network, the GNN layer (or graph convolution) is defined as follows, as depicted in Fig. 7:[25]

$$H^{l+1} = \sigma\left(\hat{A}H^{(l)}W^{(l)}\right). \tag{12}$$

In this equation, the normalized adjacency matrix of graph G is denoted by $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, where $\tilde{A} = A + I_N$ includes added self-connections, I_N is the identity matrix, and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. The matrix $W^{(l)}$ is a layer-specific trainable weight matrix, and σ is an activation function. The softmax function, defined as $softmax(x_i) = \frac{1}{Z}\exp(x_i)$ with $Z = \sum_i \exp(x_i)$, is applied at the output of the last layer to the node feature matrix to produce the final output of the network

$$Z = softmax\left(\hat{A}H^{(K-1)}W^{(K-1)}\right). \tag{13a}$$

The loss function is given by

$$L = -\sum_{s\in Y_L}\sum_{f=1}^{F_K} Y_{sf} ln Z_{sf}. \tag{13b}$$

This approach integrates quantum mechanics into graph convolutional networks, offering potential improvements in computational efficiency and accuracy for complex node classification tasks. By leveraging the principles of quantum computing, quantum graph convolutional networks aim to enhance the capabilities of traditional GNNs, providing more robust solutions for a variety of applications.

F. Quantum circuits for graph neural networks

Noisy intermediate-scale quantum (NISQ) devices have spurred advancements in quantum graph neural networks (QGNNs), which integrate quantum modules with graph neural networks to enhance and optimize current models.[9] QGNNs offer several advantages, including reducing the complexity of learning models and the number of training parameters. There are two primary approaches to implementing QGNNs:

1. **Quantum algorithms with fault-tolerant quantum computers**: This approach focuses on accelerating the computational steps of classical graph neural network models using quantum algorithms.
2. **NISQ-based methods**: In this approach, the structure of graph neural networks is modified with quantum circuits.[75] Parameterized quantum circuits (PQCs) are a key component of this approach, employing both fixed quantum gates (e.g., Pauli-Z gates) and adjustable quantum gates (e.g., Ry(θ) gates).[95] These adjustable quantum gate parameters approximate the objective function of learning models. Notably, in PQC-modified graph neural networks, input graph data is encoded into quantum amplitudes.

Tüysüz *et al.*[81] developed a hybrid quantum–classical graph neural network (HQGNN) for particle track reconstruction problems.

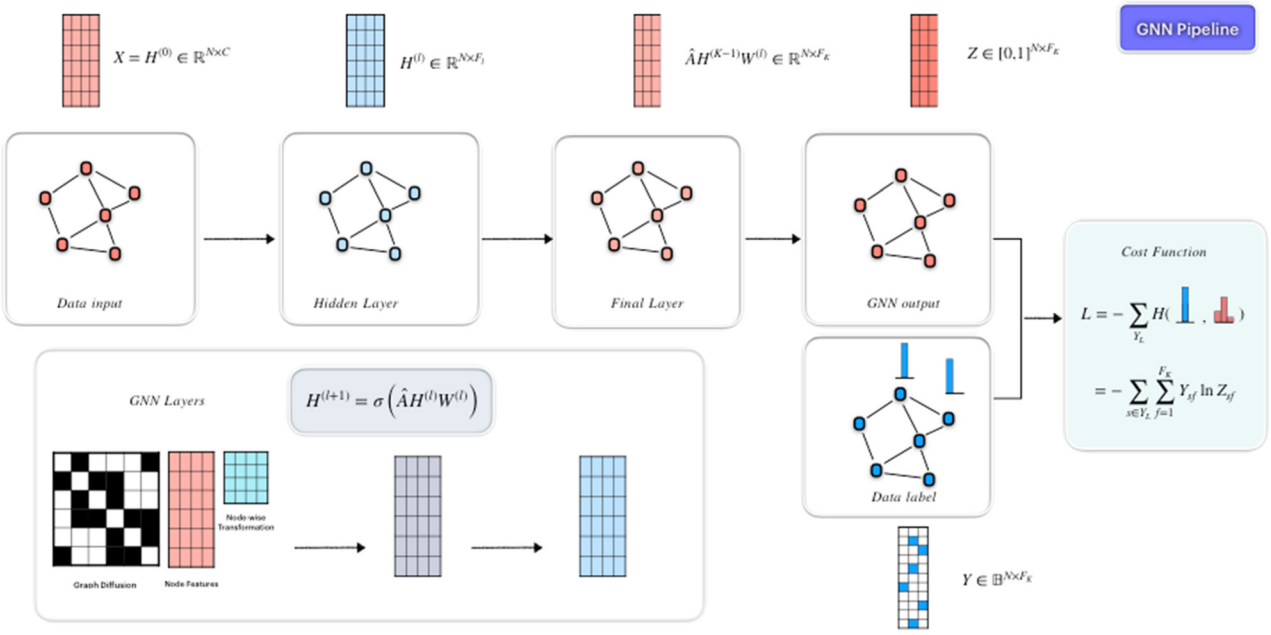

FIG. 7. Pipeline of graph convolution networks (GCNs) with a series of layers.[90] Nonlinear activation functions and convolution are applied to the node features. The softmax activation function is used at the output of the last layer.[90] Reproduced from Liao *et al.*, arXiv:2405.17060v1 (2024). Copyright 2024 Authors, licensed under a Creative Commons Attribution 4.0 International (CC BY-NC-ND 4.0) License.

To achieve optimal results with HQGNN, they utilized two types of PQCs with different repressibility and entangling capacities: one composed of parameterized gate layers and the other consisting of circuits with hierarchical architectures (containing multiple layers).

More recently, Ai *et al.*[74] introduced a decomposition quantum graph neural network (DQGNN) using a fixed-sized quantum device to handle larger graph data. They claimed that by using unitary matrix representation and tensor products, it is possible to reduce the number of parameters required, even with limited physical qubits.

Equivariant quantum graph circuits (EQGCs) have been proposed by Mernyei *et al.*[96] as a class of PQCs with two subclasses: equivariantly diagonalizable unitary quantum graph circuits (EDU-QGCs) and equivariant Hamiltonian quantum graph circuits (EH-QGCs). These provide a unifying framework for QGNNs, enabling efficient processing of graph data with quantum methods.

V. QUANTUM GRAPH NEURAL NETWORKS

Quantum graph neural networks (QGNNs) as a new class of quantum NN ansatz, where the interaction of qubits is model as the nodes connected by edges on a quantum network of a distributed quantum system, have been developed to predict the chemical and physical properties of molecules and materials by expressing entire graph theory by a quadratic Hamiltonian. The framework of QGNNs is based on the combination of quantum models with the graph neural to optimize the complexity of learning models and number of training parameters with a variational approach to learn the establishing dynamics of the system, such as variation quantum circuit architecture (VQC). In this realm, to establish discrete link features, several techniques have been developed, such as equivariantly diagonalizable unitary quantum graph circuit (EDU-QGC), which minimizes quantum circuit embedding. Results from QGNNs reveal that this architecture can be a powerful tool for training and analyzing complex structure variables and overcome to scalability and computational challenges inherent in classical GNNs.

A. Classical graph neural networks for materials research

In materials applications, the primary task of machine learning on graphs (composed of nodes and edges) is related to graph classification or the regression of graph-wide features. These tasks require consideration of various symmetries, such as translational and rotational symmetries, which can enhance the model's performance by incorporating these biases into the machine learning model. Graph neural networks are constructed to process graph data effectively. In the case of molecules, atomistic graphs are created with atoms as nodes and bonds as edges, modeled in GNNs using a message-passing neural network (MPNN) framework (Fig. 8). This framework applies a set-to-set readout function for various property feature regression tasks.[10]

Representing materials as graphs for crystals with periodic structures in a GNN framework is a nontrivial task. In this setup, nodes represent atoms, and a threshold distance between atoms determines links in an undirected multigraph, as utilized in crystal graph convolutional neural networks (CGCNN) in 2018.[97] GNNs are capable of studying structures beyond molecules and crystals, such as effectively predicting the magnetostriction of polycrystalline materials.[98] Generally, for unseen crystals, their atomic properties, stress fields, and energy distribution can be predicted by training on crystals with vacancies or grain boundaries.[99] GNNs are also applied in polymer property prediction[100–102] and the classification of amorphous materials as either glass or liquid.[103]

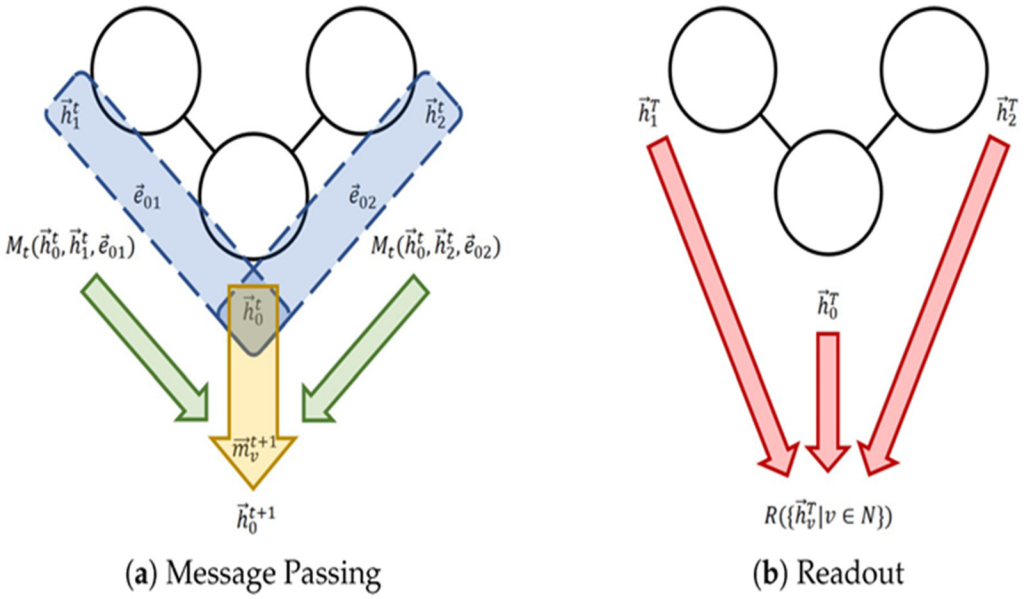

FIG. 8. The framework of the message-passing neural network (MPNN).[10] (a) One message-passing step for node 0. The calculation of individual messages is shown by the blue section; the aggregation is shown by green arrows; and the node feature update is shown by the yellow arrow. (b) The readout phase. Reproduced from Ryu *et al.*, Materials **16**(12), 4300 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) license.

Alternative approaches, such as generative models, can screen for molecules and materials with desired properties. In materials and chemistry applications, generative adversarial networks (GANs)[104] and variational autoencoders (VAEs)[105] are significant and widely studied models.

B. Variational quantum machine learning

Quantum computing has been applied to machine learning through various algorithms, including quantum kernel estimation,[106,107] quantum basic linear algebra subroutine-based algorithms,[108–111] and variational quantum machine learning (QML).[106,107,112] This section focuses on variational QML due to its potential applications with noisy intermediate-scale quantum (NISQ) computers.[113]

Hybrid algorithms, such as variational quantum algorithms, leverage both quantum and classical computers to iteratively solve problems.[114] In this approach, a quantum computer begins calculations with an initial state, undergoes time evolution, and measures expectation values using a parameterized quantum circuit (PQC).[10] Classical parameters are used to rotate the quantum state of quantum gates in a circuit diagram. PQCs are crucial for mapping classical input data samples to loss functions or output predictions. The trainable sections of the circuit are called the quantum neural network (QNN), while the data input sections are known as the quantum encoding circuit.[10]

A classical computer performs optimization by updating the parameter values based on the outputs, forming a feedback loop that repeats. In machine learning, quantum circuits serve as parameterized models.[10] Figure 9 illustrates a quantum circuit for a variational QGNN as a circuit diagram.

An essential aspect of working with classical data is representing the input as a quantum state. In variational quantum machine learning, quantum encoding of classical input, such as rotation angles, is a key concept. The circuit diagram in Fig. 10 demonstrates this concept, where the Hilbert space can be used as a feature space. Quantum encoding circuits implement different types of functions that quantum machine learning can learn. The variational model involves repeated quantum encoding algorithms, known as re-uploading, which increase the model's repressibility.[115] Constructing a learning problem with one qubit is feasible using re-uploading, whereas multiple qubits would be necessary without it.[116] A supervised learning approach has been developed to separate two ensembles of states by training the encoding through optimal measurement.[117]

Different QNN architectures depend on the structure of the gates, such as quantum convolutional neural networks (QCNN),[118] dissipative quantum neural networks,[119] and quantum graph neural networks.[9,78,81,96] These architectural designs are summarized in Fig. 11. Models are also categorized by their task, including quantum classifiers,[120] quantum autoencoders,[81,121] and quantum generative adversarial networks.[122,123]

A hybrid classical–quantum neural network is an interesting variant of QNNs,[124–129] combining classical and quantum layers by calculating gradients for the QNN layers.[130,131] Quantum encoding

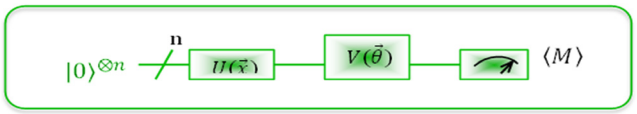

FIG. 9. Schematic of a quantum variational model circuit. The quantum encoding circuit, represented by U, maps input data into a quantum state, while V represents the trainable quantum neural network (QNN). The expectation value of an observable M is taken as the model output.[10] Reproduced from Ryu *et al.*, Materials **16**(12), 4300 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) License.

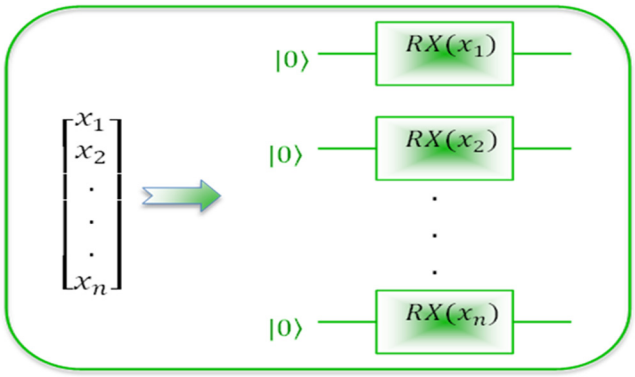

FIG. 10. Representation of quantum encoding example. Rotation angles of rotation Pauli X (RX) gates are used as individual elements of an input classical vector, creating a quantum state.[10] Reproduced from Ryu *et al.*, Materials **16**(12), 4300 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) License.

transforms classical data into quantum states, while measurements convert quantum states back to classical data. This hybrid scheme helps address limitations posed by the limited number of qubits in NISQ computers when QNNs handle high-dimensional data. Hybrid classical–quantum neural networks reduce data dimensionality for the QNN layers, making them more efficient. Gradients needed for optimization in training quantum circuits are one challenge called barren plateau, in which gradients decrease as the number of qubits increases. Consequently, several strategies such as quantum graph neural network have been developed to overcome this barren plateau for the success of VQAs.

C. Quantum graph neural network models for materials search

To learn and make predictions about the basal graph by utilizing a quantum circuit's ansatz, define quantum graph neural network (QGNN) as a state-of-the-art approach in quantum machine learning (QML). The elemental idea is to embed data by leveraging quantum dynamics, which leads to a richer feature map relative to classical method characteristics.[1] QGNN encodes the graph topology in the dynamics within the system Hamiltonian, and as a result, one can imply quantum circuits with graph-theoretic properties, where QGNN uses a variational approach as VQC to learn the basal dynamics of the system. Several models of quantum variational machine learning have been developed to address graph-structured data. QGNNs were first proposed in 2019 by Verdon *et al.*[9] using rotation-generating Hamiltonian operators, taking into account the topology of the problem graph. The most general QGNN ansatz, PQC as proposed in Ref. 9 consists of a Q sequence of Hamiltonian evolutions with the sequence repeated P times

$$U_{QGNN}(\eta,\theta)=\prod_{p=1}^{P}\prod_{q=1}^{Q}e^{-i\eta_{pq}\hat{H}_q(\theta)}, \tag{14}$$

where the η and θ are trainable (variational) parameters, the product is time-ordered, and the $\hat{H}_q(\theta)$ is parameterized Hamiltonians. A single layer of QGNN for input graph data, in which topology of interactions as each node's state in a Hilbert space with interactions between nodes connected via node-local (v) and links (E) is expressed using $\hat{H}_q(\theta)$ with operators $\hat{O}$, $\hat{P}$, $\hat{R}$ in the following equation:

$$\hat{H}_q(\theta)=\sum_{(j,k)\in E}\sum_{r\in\mathfrak{T}_{jk}}W_{qrjk}\hat{O}_j^{qr}\otimes\hat{P}_k^{(qr)} +\sum_{v\in V}\sum_{r\in\mathcal{T}_v}B_{qrv}\hat{R}_j^{(qv)}. \tag{15}$$

Here, θ is trainable variables, operators $\hat{O}_j^{qr}$, $\hat{P}_j^{(qr)}$, $\hat{R}_j^{(qv)}$ are Hermitian operators, and acting on the jth node of the graph on the Hilbert space, the W_{qjk} and B_{qrv} are real-valued coefficients and independent trainable parameters. Index sets of $\mathfrak{T}_{jk}$ and $\mathcal{T}_v$ indicate the corresponding links (graph edges) and graph nodes, respectively.

To avoid the barren plateaus problem and create the ansatz more tunable training, one must add some specificity and constraints. For instance, for tiding parameters spatially, convolutional architecture has been proposed, or for tiding over exponential mapping of the sequential iterations, recurrent architecture has been proposed.[3] Different computational models with various properties as subcategories of

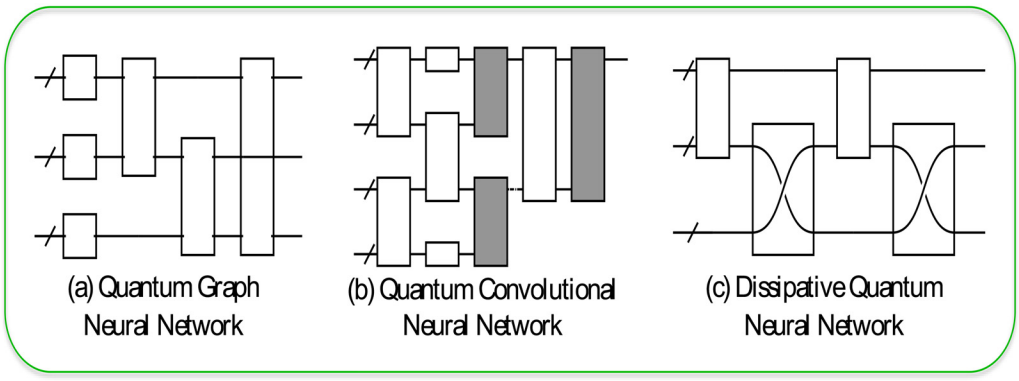

FIG. 11. Representation of different quantum neural networks (QNNs) architecture designs having trainable parameters in empty gates. (a) The circuit structure of QGNN is designed for graph-structured data processing, depending on the input graph data. (b) The circuit structure for quantum convolution NN, in which white gates are convolution operators and gray gates are pooling operators. (c) This circuit represents each neuron as a group of qubits, where unitary operators transform one layer to another for dissipative quantum. The top group of qubits as the input layer of one perceptron is mapped to a layer of two perceptrons as the bottom two groups of qubits.[10] Reproduced from Ryu *et al.*, Materials **16**(12), 4300 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) License.

QGNNs, such as QRecGNNs (quantum recurrent graph neural networks), QCGNNs (quantum convolutional graph neural networks), and EQGC (equivariant graph quantum circuits), have been utilized for toy problems like Hamiltonian dynamics learning and graph clustering.

1. Quantum recurrent graph neural networks

Subclass of the more general QGNN ansatz and classical RecGNNs creates QRecGNNs, which is defined as ansatz explained in Eq. (14) with binding temporal parameters between iterations of outer sequence by considering $\eta_{pq} \rightarrow \eta_q$, i.e., for each iteration, where the temporal parameters are similar to classical RecGNNs by sharing parameters over sequential application of the recurrent map. The QGRNN ansatz can be considered as a Trotter-based quantum simulation of the time parameter η_q for Hamiltonian evolution under $e^{-i\Delta \hat{H}_{eff}}$, where the effective Hamiltonian $\hat{H}_{eff} = \Delta^{-1} \sum_q \eta_q \hat{H}_q$ with time step of $\Delta = \sum_q |\eta_q|$. Herein, to learn effective quantum dynamics raised on a graph system, this ansatz is tailored.[9]

In other words, QRecGNN is defined as a PQC to map the established graph structure into Q different Hamiltonian evolution sequences of quantum circuits, where the whole outer sequence is indexes repeated P times ($p = 1, 2, \ldots, P$) times as follows:

$$U_{QRecGNN}(\eta, \theta) = \prod_{p=1}^{P} \prod_{q=1}^{Q} e^{-i\eta_q \hat{H}_q}. \tag{16}$$

It is worth pointing out that for handling graphs with QRecGNN effectively, one has to define a quantum architecture consisting of all graph transformations and graph operations.[4]

2. Equivariant quantum graph circuits

An implementation of the equivariant concept from geometric ML has been developed by Mernyei *et al.*[96] as the equivariant quantum graph circuits (EQGCs). Specifically in this model, given input graph data consisting of tuples of nodes, links, and node features are assigned to a fixed number of qubits per node. Next, to encode the node features into quantum states, parameterized unitary is applied onto the corresponding qubits. Then, a quantum circuit (unitary matrix) of node-permutation-equivariant, i.e., the permutation and the quantum circuit can commute with each other, is applied. Finally, a measurement based on node-permutation-invariant is applied, where the outputs by applying a parametrized classical function are post-processed. The node-permutation-invariant measurement design can be the expectation value average of a node-local observable over all nodes. This computational framework as the circuit can be visualized in Fig. 12.

Constructing an EQGC can be done in two main ways as suggested in Ref. 85. One approach is similar to Verdon *et al.*'s[9] proposition for a QGNN i.e., the equivariant Hamiltonian quantum graph circuit (EH-QGC), where the QNN is composed of rotation-generating Hamiltonian operators with the same topology as the input graph.[10]

Another approach is the equivariantly diagonalizable unitary quantum graph circuits (EDU-QGCs), consisting of node and link layers. Each node layer contains a node-local unitary acting on all nodes, while each link layer contains equivariantly diagonalizable unitaries (EDUs) acting between two nodes connected by a link. EDUs act on two nodes and can be decomposed as shown in Eq. (15), where the unitary operator V acts on one node, and D is a diagonal unitary acting

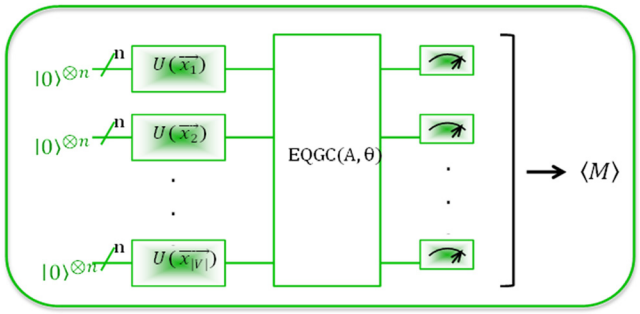

FIG. 12. The framework of equivariant quantum graph circuits (EQGC). The input graph with nodes V, adjacency matrix A, and node features $\{\vec{x}_i\}_{i \in V}$ is used in the circuit shown above.[10] The measurement is invariant under the permutation of nodes, while the EQGC is equivariant under node permutation and is trainable.[10] Reproduced from Ryu *et al.*, Materials **16**(12), 4300 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) License.

on two nodes. The link layer is equivariant under node permutation since the EDU commutes with the SWAP operator,[10]

$$EDU = (V^+ \oplus V^+) D (V \otimes V), \tag{17}$$

here, the unitary operator V acts on one node, while D is a diagonal unitary acting on two nodes. The link layer is equivariant under node permutation due to the commutation of EDU with the SWAP operator and the action of itself copy on other qubits. EDU-QGCs, as shown in Fig. 13, can approximate any real-valued function over bounded graphs and pass the 1-WL test, unlike deterministic classical MPNNs.[76] The previous approach's challenge is handling small graph due to requiring many qubits for large quantum circuits. To resolve this issue, Ai *et al.*[74] proposed a quantum subgraph decomposition model in conjunction with CNOT gates to achieve the goal of GNNs and the lack of available qubits.

The fusion of graph learning algorithms with visual networks has exhibited major capabilities to extract semantic features in various visual tasks, standing to reason the mutualism of quantum graph learning and **image classification**, which discussed in detail in Sec. VI.

VI. QML FOR IMAGE CLASSIFICATION

The implementation of image processing techniques has profoundly impacted various fields. Digital images can now be formatted to facilitate human interpretation and machine interpretation of extracted information. Image classification is crucial in many scientific and technological areas, including medical diagnosis, image retrieval, traffic sensing for autonomous vehicles, face detection, and image reconstruction.[11–15] Generally, the image classification pipeline involves image preprocessing, feature extraction, and object classification. In recent years, significant progress has been made in the practical applications of Noisy Intermediate State Quantum (NISQ) devices and the development of quantum computers, which offer high efficiency and accuracy for quantum machine learning (QML).[16] Within the QML framework, quantum image processing (QIP) and classification represent essential application areas.[17]

Unlike classical image processing, as illustrated in Fig. 14, QIP first encodes image data into a quantum circuit, performs quantum image transformation, and then decodes the image.[18] On a classical

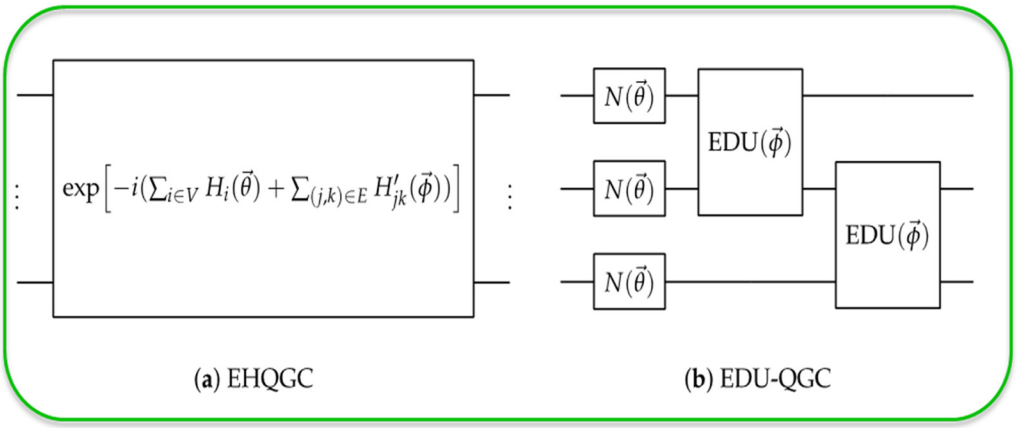

FIG. 13. Representation of two different methodologies of equivariant quantum graph circuit (EQGC). (a) Indication of EH-QGC, which is an equivariant Hamiltonian for quantum graph circuit, and (b) indication of the EDU-QGC method, which is made up of node-local unitary operators (node layers) acting on all nodes and link layers as equivariantly diagonalizable unitaries.[10] Reproduced from Ryu *et al.*, Materials **16**(12), 4300 (2023). Copyright 2023 Authors, licensed under a Creative Commons Attribution (CC BY) License.

computer, an $M \times L$ image is represented as a matrix, encoded with at least 2^n bit $[n=\lceil \log_2(ML) \rceil]$. Classical image transformation involves matrix computation. Conversely, a quantum state can represent the same image and be encoded in n qubits. Quantum image transformation is achieved through unitary evolution $\hat{U}$ under an appropriate Hamiltonian. In Secs. IV A–IV C, we summarize QML methodologies for effective image classification using qubits, quantum gates, and quantum circuits.

A. Image encoding methods

Classification is a common supervised learning task. It involves approximating a function f to map input data x to a discrete target output y, where $y = f(x)$. The primary objective of classification is to create a discrimination function to accurately predict class labels. In the QML domain, many methods exist for encoding classical images into quantum states in Hilbert space.[12,16,17,132] With the limited number of qubits in the NISQ era, the choice of encoding methods can significantly affect the quality and efficiency of the overall quantum circuit. Kharsa *et al.*[12] summarized major encoding methods, illustrated in Fig. 15. Here, we briefly introduce the most popular image encoding methods.

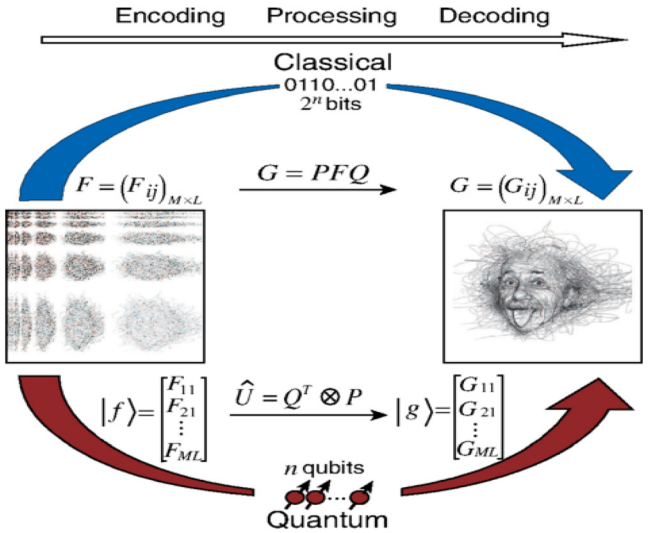

FIG. 14. Comparison of image processing by classical and quantum computers. *F* and *G* are the input and output images.[18] Reproduced from Yao *et al.*, Phys. Rev. X **7**(3), 031041 (2017). Copyright 2017 Authors, licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.

1. Basis encoding

The basis encoding method associates each classical input with the computational basis of a quantum state. Given a dataset D containing M samples with N features, the data must first be converted into a binary encoded form. Using basis encoding, dataset D is represented as a superposition of computational basis states x^m:[16]

$$|D\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} |x^m\rangle, \tag{18}$$

where classical data $D = \{x^1, x^2, \ldots, x^M\}$ is in the form of binary strings, $x^m = \{b_1, b_2, \ldots, b_N\}$, with $b_i = 0$ or 1, $i = 1, 2, \ldots, N$. Basis encoding requires as many qubits as there are bits in classical representations.

2. Amplitude encoding

Since n qubits can have 2^n amplitudes, these amplitudes can encode image pixels using a practical number of qubits. Therefore, given a classical image array of N pixels as $[x_1, x_2, \ldots, x_N]$, it needs $n = \log_2(N)$ qubits for embedding in a quantum circuit. The resulting quantum state $|\psi\rangle$ is defined as

$$|\psi\rangle = \frac{1}{\sum_{i=1}^{N} ||x_i||} (x_1|0_1 0_2 \cdots 0_n\rangle + \cdots + x_N|1_1 1_2 \cdots 1_n\rangle), \tag{19}$$

where the denominator is a normalizing factor. Compared to basis encoding, amplitude embedding uses fewer qubits, with the number of

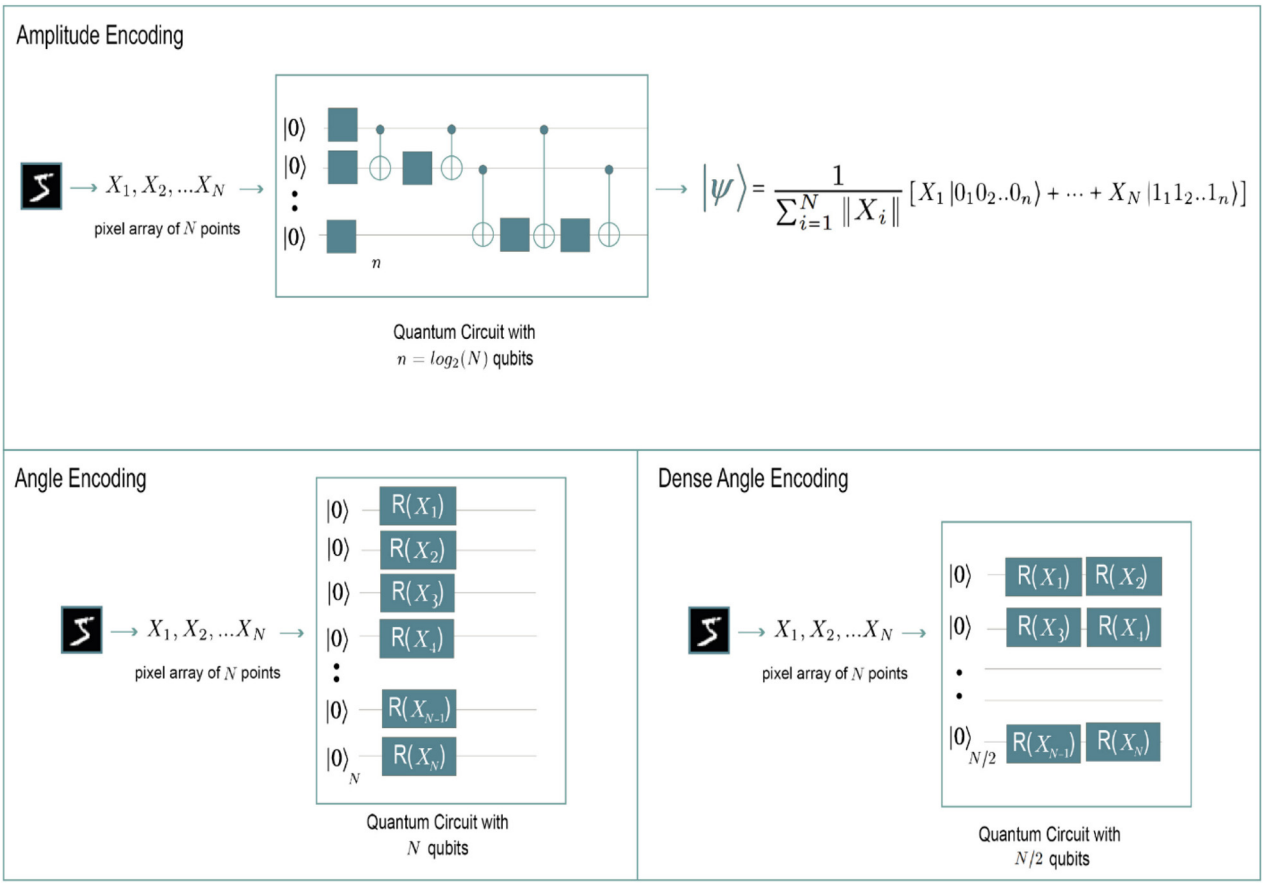

FIG. 15. Amplitude, angle, and dense angle encoding schemes.[12] Reproduced with permission from Kharsa *et al.*, Neurocomputing **560**, 126843 (2023). Copyright 2023 Elsevier.

required qubits being $O(\log_2(N))$. On the other hand, this encoding method involves computational complexity and requires longer circuit depth, scaling exponentially with the number of qubits (i.e., $O(2^n)$ for n qubits).

3. Angle encoding

Angle encoding represents image pixels x_i as rotation angles of quantum states using parameterized rotation gates ($R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$). Pixels are converted into angles using a transformation, such as $\theta_i = \pi \times x_i$. Obviously, the resulting quantum state $|\psi\rangle$ has $O(2^n)$ amplitudes and can be expressed as

$$|\psi\rangle = R(\theta_1)|0\rangle \otimes R(\theta_2)|0\rangle \otimes \cdots \otimes R(\theta_n)|0\rangle. \quad (20)$$

Angle encoding's main advantage is that it has shallow circuit depth with constant complexity $O(1)$, as only one gate is used per qubit. Hence, angle encoding requires $O(N)$ qubits.

4. Dense angle encoding

The dense angle encoding differs from regular angle encoding by encoding two pixels per qubit, which reduces the required qubits to $N/2$ and quantum state amplitudes to $O(2^{N/2})$. The second pixel in each qubit is encoded with a phase shift gate P, and the resulting quantum state $|\psi\rangle$ is expressed as

$$|\psi\rangle = R(\theta_1)R(\theta_2)|0\rangle \otimes R(\theta_2)P(\theta_3)|0\rangle \otimes \cdots \otimes R(\theta_{n-1})P(\theta_n)|0\rangle. \quad (21)$$

5. Hybrid encoding

From the above encoding schemes, one can see that reducing the number of qubits and gates can decrease circuit complexity and improve algorithm efficiency. Amplitude encoding requires fewer qubits but more circuit depth, while angle encoding demands more qubits but fewer transformation gates. Dense angle encoding balances these extremes with shorter length and shallower depth. Advanced encoding methods that combine amplitude and angle strategies have been developed. More details can be found from Ref. 133.

B. QML algorithms for image classification

Quantum machine learning (QML) encompasses a variety of approaches based on the integration of quantum or classical

algorithms and quantum or classical data. These approaches can be classified into four categories:[16]

- **Quantum Algorithm-Quantum Data (QQ)**: This category, also known as purely QML, involves quantum algorithms processing quantum data.
- **Quantum Algorithm-Classical Data (QC)**: In this category, quantum algorithms are used to learn from classical data inputs.
- **Classical Algorithm-Quantum Data (CQ)**: CQ algorithms apply classical machine learning techniques to quantum data and can be executed on quantum devices.
- **Classical Algorithm-Classical Data (CC)**: Often referred to as quantum-inspired ML, these algorithms utilize features of quantum computing (i.e., qubits, superposition, and entanglement) to enhance accuracy but are executed on classical computers.

Since image data are inherently classical, quantum image processing and classification mainly belong to the QC category. Numerous QML algorithms have been developed to address image processing and classification challenges.

1. Quantum support vector machine (QSVM)

QSVM algorithms use quantum computers to optimize finding a hyperplane in a high-dimensional space that maximally separates different classes. The core computational step in the QSVM algorithm involves solving a quadratic programming problem, which is NP-hard.[134] The solution is achieved through discretization and a brute-force algorithm based on Grover's search algorithm.[59] This algorithm offers a quadratic speedup compared to classical algorithms. Generally, QSVM employs a quantum circuit to define the kernel function for speedup and demonstrates a quantum advantage in the near future for classification problems.[135] The advantage of QSVM with certain feature maps for classically NP-hard problems has been analyzed for the regime of fault-tolerant quantum computing.[136]

2. Quantum K nearest neighbor (QKNN)

The k-nearest neighbor (k-NN) algorithm assigns an object to the class of the majority of its k-nearest neighbors in a multidimensional feature space. During the learning process, distances between unclassified objects and previously classified ones are calculated. Given the computational complexity of k-NN, which increases quadratically with the number of objects, and its inherent parallelism in solving classification problems, QKNN is a superior approach for image classification. QKNN for image classification involves four steps: (i) using a classical computer to extract feature vectors from the image, (ii) encoding feature vectors into quantum superposition states for parallel execution, (iii) employing a quantum minimum search algorithm to accelerate similarity searches, and (iv) conducting quantum measurement to realize classification.[17] In QKNN, due to the quantum superposition phenomenon, all distances can be calculated in parallel and encoded in amplitudes.

3. Quantum tensor networks (QTNs)

Tensor networks (TNs), built from multidimensional arrays called tensors, are a successful machine learning technique for decomposing large data structures into several connected low-rank tensors. These networks have been adapted to the quantum realm in quantum ML to tackle problems beyond the reach of classical computers. TNs, positioned at the interface between physics and ML, are easily deployable on quantum computers as QTNs, implementing classical TNs as variational quantum circuits (VQCs).[137,138] Different QTN architectures [e.g., Matrix Product State (MPS), variational QTN (VQTN), Multi-scale Entanglement Renormalization Ansatz (MERA), Tree Tensor Network (TTN), etc.] can be used for image classification. To perform QTN, the process involves encoding and transforming images into a quantum state $|\psi\rangle$, then processing them with a unique QTN architecture. This typically involves unitary gate U on each pair of successive qubits followed by CNOT gates to entangle them, maintaining a similar connection to the original network.[12] QTNs are expected to outperform classical TNs, as the quantum algorithms inherently implement entanglement, accessing a Hilbert space that grows exponentially with the number of qubits. This leads to increased storage capacity and a larger parameter space for QML algorithms.[137] Moreover, QTNs also offer a promising and often easier-to-train framework compared to other QML methods.

4. Quantum convolutional neural networks (QCNN)

Exploring the integration of quantum computing and deep learning is complex yet highly valuable. A quantum deep learning (QDL) framework should:[139] (i) integrate neural networks' nonlinear dynamics with quantum computing's linear unitary dynamics; (ii) be based on the principles of quantum mechanics, rather than quantum-inspired mathematical descriptions; (iii) create efficient quantum operations to extract deep semantic features; and (iv) use quantum parallelism in both storage and evolution to overcome classical neural computing challenges. Inspired by classical CNN, Cong *et al.*[118] proposed a QCNN for phase classification and optimization of quantum error correction codes. Such a QCNN had a similar network architecture to CNN. Later, quantum deep convolutional neural network (QDCNN) algorithms have achieved exponential computational acceleration.[139] Among QDL approaches, QCNN is the most prominent algorithm for image classification. Similar to classical CNNs, QCNN consists of multiple convolution and pooling layers.[118] There are two main implementations of QCNN: applying a quantum convolution filter to a classical circuit and implementing all classification layers as a quantum circuit of quantum convolution and pooling layers, such as quantum deep CNN (QDCNN) based on parameterized quantum circuits and hybrid quantum–classical convolutional neural network (HQCCNN). Merging geometric deep learning with QML leads to geometric QML (GQML) algorithms.[140]

By combining parameterized quantum circuits (PQCs) with CNN, Li *et al.*[141] proposed a hybrid quantum–classical convolutional neural network (HQCCNN) for image classification. As depicted in Fig. 16, a m × m size image is first downsampled and mapped to quantum states $|\psi\rangle$ by the rotational encoding method. Then, the quantum convolution kernel u(θ) designed by PQC is used to perform unitary transformation on $|\psi\rangle$ with qubits corresponding to the conventional window. Further, the quantum pooling unitary gate U_j, where j represents the corresponding convolution layer composed of CNOT gates is used to reduce the dimension of the convolution results. The decomposition of U_j is crucial in constructing QCNN. Following each convolution layer, the pooling layer applies single-qubit unitary gates V_j on one qubit and measures the successive one, halving the number of

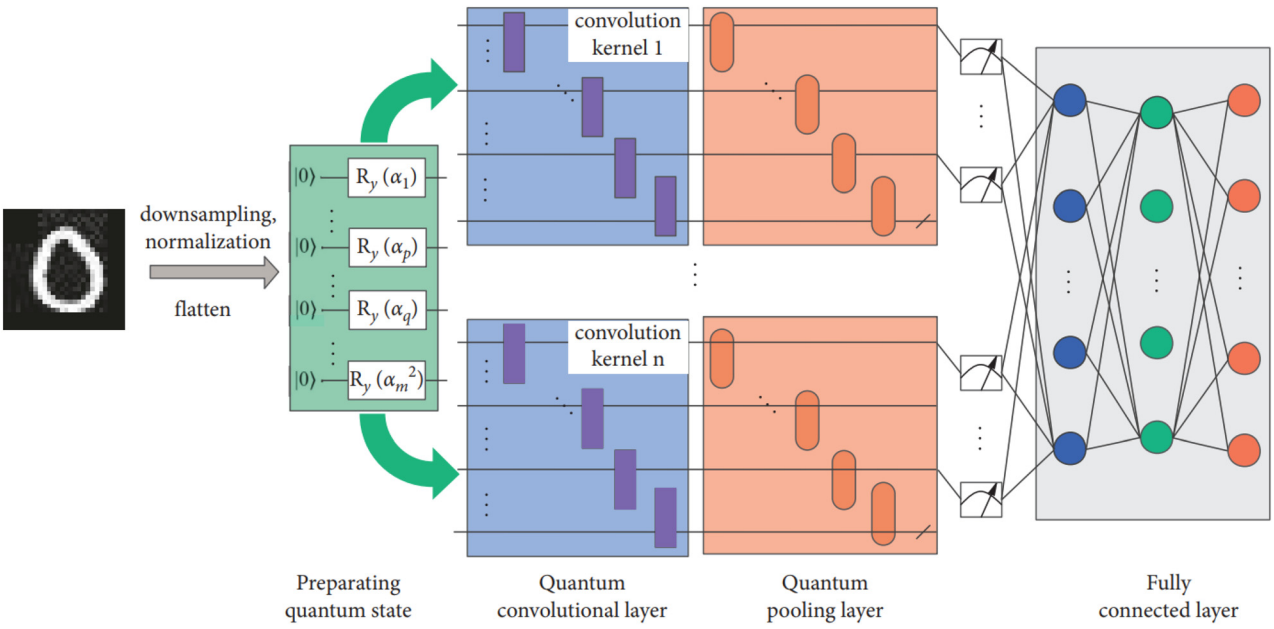

FIG. 16. Implementation scheme of HQCCNN for image classification. Reprinted from Ref. 141. Reproduced from Li *et al.*, Quantum Eng. **2022**, 5701479 (2022). Copyright 2022 Authors, licensed under a Creative Commons Attribution (CC BY 4.0) International License.

qubits in each pooling layer. After several convolution and pooling layers, when the number of qubits is suitable, the fully connected (FC) layer of QCNN entangles the remaining qubits with controlled gates (e.g., CNOT and CZ).[12] Finally, the measurement results are input into the FC layer to obtain the class of image.

C. Performance on image datasets

To validate QML algorithms for image classification, researchers have applied them to several image datasets, comparing their performance with classical ML algorithms. Several popular image datasets have been established for evaluating ML models. Here are a few datasets that have been used to test QML algorithms for image classification:

(i) **MNIST**: The Modified National Institute of Standards and Technology (MNIST) database contains handwritten digit images for ML research.[142] It includes 68 classifiers, with 60 000 training images and 10 000 test images, each measuring 28×28 pixels.

(ii) **Fashion-MNIST**: This new dataset consists of 28×28 grayscale images of 70 000 fashion products from 10 categories, with 7000 images per category. The training set contains 60 000 images, and the test set contains 10 000 images. Fashion-MNIST serves as a direct drop-in replacement for the original MNIST dataset, sharing the same image size, data format, and training/testing splits.[143]

(iii) **BAS**: Bars and Stripes (BAS) is a synthetic dataset used to study generative models for unsupervised ML.[144] It comprises binary black and white images of size $n \times n$ pixels, where either all pixels in a column have the same color (bars) or all pixels in a row have the same color (stripes). The classification task is to label the image correctly as bars or stripes.[138]

(iv) **Caltech-101**: This dataset contains digital images intended for computer vision research, comprising 101 object categories, with each category containing between 45 and 400 images.[145]

(v) **Graz-01**: This dataset includes two object classes (bikes and persons) and one background class. Graz-01 is widely used in image classification tasks to compare the accuracy and effectiveness of different methods. It contains 450 images of category "person," 350 of category "bike," and 250 of category "background."[146]

(vi) **CIFAR-10**: The Canadian Institute for Advanced Research dataset (CIFAR-10) consists of 60 000 32×32 color images in 10 classes, representing airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6000 images per class.[147]

(vii) **GTSRB**: The German TrafficSign Recognition (GTSRB) dataset consists of 51 839 color images of size $32 \times 32 \times 3$, divided into 43 categories (e.g., speed limit, no passing, priority road, no vehicles, no entry, double curve) with an imbalanced class distribution. Unlike handwritten digit recognition, GTSRB contains real-world objects.[148]

(viii) **Other datasets**: Additional image and video clip datasets used in image classification include FER2013, Gun, Knife, CK+, and CelebA.[140,149]

Table II summarizes the accuracy of various QML algorithms applied to the domain of image classification in comparison with classical ML algorithms. Each QML algorithm may use different encoding methods and numbers of qubits. Some research works apply their QML algorithms to the entire dataset (e.g., MNIST and Fashion-MNIST),

TABLE II. Performance of QML applied to the domain of image classification.

QML algorithm	Encode method	# of qubit	Dataset	Classes	Accuracy (%)	Ref.
QSVM	Angle	2	Knife	2	93, 87 (SVM)	149
			Gun	30	98, 83 (SVM)	
			CK+	1	98, 95 (SVM)	
			FER2013	7	92, 83 (SVM)	
QSVM	Amplitude	4	MNIST digits (69)	2	>99	150
QSVM	Hybrid	4–12	Credit Card	1	70,[a] 70,[b] 70 (SVM)	136
			MNIST	2	100,[a] 100,[b] 70 (SVM)	
			Fashion-MNIST	2	100,[a] 100,[b] 100 (SVM)	
QKNN	Amplitude	2	Caltech-101	9	78 (k = 3)	151
			Graz-01	2	83.1 (k = 5)	
QKNN	Basis	10	MNIST	10	97 (k = 30), 95 (KNN)	152
			Fashion-MNIST	10	90 (k = 30), 86 (KNN)	
			CIFAR-10	10	51 (k = 30), 41 (KNN)	
QKNN	Basis		MNIST	10	98 (k = 20)	153
QTN	Amplitude	2	MNIST	2	59–93 (TTN) 82–96 (MERA)	154
QTN	Amplitude	4, 8, 16	BAS	2	100	138
QTN	Angle	8	MNIST digits (01)	2	99.87	155
			MNIST digits (27)	2	98.86	
			MNIST digits (even/odd)	2	84.85	
QNN	Angle	8	MNIST	10	90	156
QNN	Angle	64	MNIST	10	70.52, 67.51 (CNN)	157
VQTN	Dense angle	8	MNIST	10	∼93	158
VQDNN	Hybrid	6, 10	MNIST digits (01)	2	100	133
			UCI	10	90.87	
QCNN	Amplitude	14	MNIST	2	96, 97 (CNN)	159
			MNIST	10	74, 80 (CNN)	
QCNN	Angle	9	Fashion-MNIST	10	98	160
			CIFAR-10	10	90	
QCNN	Amplitude	2	MNIST	10	63	161
			Fashion-MNIST	10	98	
QCNN	Hybrid	4	MNIST digits (3456)	4	85.14, 94.25 (CNN)	162
			MNIST digits (0123)	4	90.03, 95.85 (CNN)	
			Fashion-MNST (0123)	4	85.93, 89.69 (CNN)	
			Fashion-MNIST (1289)	4	93.63, 97.42 (CNN)	
QCNN	Amplitude	6	MNIST digits (36)	2	96.65	163
QDCNN	Hybrid	$2n + 5d + p + 2$	MNIST	10	98.97, 96.33 (DCNN)	139
			GTSRB	10	91.40, 91.57 (DCNN)	
HQNN	Hybrid	5	MNIST	10	99.21, 98.71 (CNN)	164
			CIFAR-10	10	82.78, 82.64 (CNN)	
HQCCNN	Angle	4	MNIST	10	87.59 (with pooling), 86.45 (CNN) 92.40 (no pooling), 90.11 (CNN)	141
GQML	Amplitude	10, 12	MNIST digits (018)	3	>96, 96 (CNN)	140
			CIFAR-10	4	80, 70 (CNN)	
			CelebA	2	74, 60 (CNN)	

[a]QSVC under noiseless quantum kernel simulations.
[b]QSVC on the IonQ Harmony quantum computer.

while others use subsets (a portion of images and classes) to demonstrate performance. In Table II, the accuracy of QML algorithms is listed alongside available corresponding values from classical ML algorithms for comparison.

From Table II, it is evident that QML algorithms for image classification generally perform better than their classical ML counterparts. Each QML algorithm has its advantages and disadvantages. In some cases, QCNN algorithms[159,162] show slightly lower accuracy than classical CNN algorithms. However, with deep learning and hybrid techniques, improved QML algorithms (e.g., QDCNN, HQNN, and HQCCNN)[139,141,164] demonstrate better accuracy than classical CNN approaches. Interestingly, using a quantum circuit simulator, both with and without noise, as well as the IonQ Harmony quantum processor, Suzuki *et al.*[136] explored the quantum support vector classification (QSVC) and quantum support vector regression (QSVR) models on three different datasets (MNIST, Fashion-MNIST, and financial credit card). Their results showed that the performance of QSVC models using four qubits of the trapped-ion quantum computer was comparable to that obtained from noiseless quantum circuit simulations. The accuracy of the credit card dataset is only 70%, while accuracies of other two datasets are reaching 100%, as summarized in Table II.

VII. APPLICATIONS FOR QUANTUM ADVANTAGE

The fusion of quantum computing and GNNs as a powerful tool emerges to overcome the scalability and computing challenges in classical counterparts, where it suffers from high computational complexity and over-smoothing in large-scale systems. Superposition and entanglement principles in quantum computing boost computational capabilities. In this review paper, various architectures of the state-of-the-art of QGNNs are compared across several fields, such as molecular chemistry, high-energy physics, earth sciences, and finance, where some challenges related to scalability and decoherence issues are addressed.

High-energy physics: Particle track reconstruction[165] in the Large Hadron Collider (LHC) for predicting new particles encounters scalability challenges to distinguish signals, which reveal difficulty in rapid track reconstruction.[166] To speed up and improve the efficiency of particle track reconstruction, QGNNs have been applied, where for expanding input data dimensions, a hybrid of input network and edge and node networks combines classical and quantum layers.[81,167] Variations of hidden dimension size ND are used in this promising architecture as well as the quantum circuit and the number of qubits, where varying hyperparameters were implemented to analyze their results. Moreover, the type of elementary particles that initiate jets can be identified by using QGNNs as another promising application of this architecture.[81,167]

Molecular biology and chemistry: Prediction of material properties of large molecules based on traditional laboratory techniques undergoes high computational costs; therefore, GNNs have become essential.[168] However, growing datasets require data-driven approaches,[169] where GNNs suffer from convergence issues of the message-passing approach and insufficient training data for exotic compounds.[170] To resolve these issues, QGNNs leverage QML, improving performance in molecular engineering by attributing molecular structure to higher-dimensional spaces and figuring out long-range correlations through quantum texturally.[171,172] For instance, for predicting the formation energy of perovskites,[173] implement a Hybrid Quantum-Classical Convoluted Graph Neural Network (HyQCGNN) using a specific dataset.[174]

Analyzing complex proteins and their roles and challenges in disease mechanisms applicable in drug discovery seeks reliable method,[175,176] where QML and QGNNs play a significant and critical role in this field.[177] Integration of SGCNN and 3D-CNN with a QNN as a hybrid model can enhance performance and stable convergence on the PDBbind dataset with improvements in metrics in comparison to classical models.[178] Additionally, to analyze cancer cells' interaction precisely for selecting appropriate treatments, QGNNs have been applied,[179,180] which offer potential solution for challenges of classical and traditional GNNs such as over-squashing and over-smoothing. For instance, breast cancer subtyping dataset[181,182] by implementing hybrid QGNNs model exhibits results for larger output dimensions as a practical tool in these biological scenarios.

Complex systems: For analyzing complex systems, QGNNs demonstrate a robust advantage to process and capture the full complexity of spatiotemporal characteristics such as traffic congestion prediction (TCP)[183] and traffic collision avoidance systems (TCAS) in intelligent transportation systems;[184] in the sector of reconstructing airplane routes to avoid two craft collided[185] and forecasting of the Ocean Niño Index (ONI).[186]

Sensor networks: Estimation of hidden parameters in weak qubit phase rotation signals in quantum communication/networking and quantum sensing technologies is an active field in the application of QConvGNN.[187,188] In fact, a GHZ state as a special type of entangled quantum state provides an advantage, which QConvGNN can learn to prepare a GHZ state without knowing the entire structure of the quantum network.[189]

Learning quantum Hamiltonian dynamics: The high potential of QRecGNN is effectively learning the dynamics of an Ising spin model of a closed quantum system,[9,190,191] where the Ising model emulates the modeling of non-deterministic polynomial (NP) problems with QUBO.[192,193] In a supervised learning scenario, where each node is mapped to a labeled quantum state, Beer *et al.*[78] defined graph-structured quantum data with each node of a given graph to a quantum state, and the existence of links between two quantum states with a certain information-theoretical distance. They designed loss functions and training methods using dissipative quantum neural networks.

VIII. PRACTICAL IMPLEMENTATION OF QUANTUM GRAPH LEARNING/NEURAL NETWORK

QISKIT and PennyLane are both practical implementation of quantum graph learning, which are discussed in this section. A practical approach to implementing the quantum graph learning concepts as discussed in this paper is represented by IBM's framework in QISKIT, an open-source and Python-based framework,[194–197] which one can run the quantum simulations and design quantum circuits on a real quantum device or on the local workstation and measure the results and analyze them.

A. QISKIT

In QISKIT, to find a certain quantum state from a known starting state as $|0\rangle$, a method has been programmed as sequential quantum gates or directly in Python. By acquiring a broad state of determined number of qubits with different chains of events, quantum exploration

could be the objective goal, where applying certain gates will concern users, i.e., several concepts such as gate cost, complexity, general design issues, quantum processor limitation, and integration can be rushed by these future constraints.[198] In this context, an exploration graph is created, in which all possible states for an initial configuration of the qubits emerge and are select by the user. Various types of information from each node of the graph can be obtained in this platform, such as the necessary cost, the transition route, i.e., the applied gates reflected from the initial state and amplitude of each state as well as revealing of the quantum circuits. In this framework, to minimize the cost function some algorithms are defined as well as to use the states of the graph for investigating a configuration of a QNN.[198]

1. Practical steps for implementation of QGRNN on QISKIT framework

To configure a QNN using the graph states, an exploration graph can be generated with the following feature elements: (i) The information of the quantum circuit has been held on each node, which includes the number of qubits and used gates for each state. (ii) Acting quantum gates on one or more qubits represents a transition state. (iii) Analyzing the intermediate state. (iv) Verification of the newly obtained state to investigate its capability as a new node in the graph or as a circuit design by using Statevector amplitudes (amplitude list in QISKIT).

To generate and analyze the graph, the classes are built on top of Qiskit, including quantum circuits and quantum registers as the QunatumGate-class and QuantumState-class for keeping the qubits' information, and a QuantumTransition-class and QuantumGraph-class for generating and processing the local circuit using a graph (simulator, Statevector class).[196–198]

B. PennyLane quantum platform

One of the promising experimental quantum platforms for revealing advantages of QGNN is PennyLane as a compatible quantum Python package.[194] In this context, assigning numbers to nodes and edges (i.e., weights) encodes information into graphs. Mapping data of a given graph into a low-dimensional vector space is trained by GNN, assigning feature vectors to nodes and edges, where both features and topology of graphs (i.e., connection of nodes and edges) preserved in the learned vectors. A similar process will be done by QGNN, where features are quantum-mechanical states and properties. To learn the dynamics of quantum systems, the QGRNN ansatz as an approximation of the time evolution of quadratic Hamiltonian, which includes some terms of interaction between two qubits and other terms of energy of individual qubits, describing them by graphs.[195,196] The weighted edges between nodes have been taken for second-order terms in this class of Hamiltonians, where the node weights are considered for first-order terms. The transverse field Ising model is an example of a quadratic Hamiltonian by definition of

$$\hat{H}_{Ising}(\theta) = \sum_{(ij)\epsilon E} \theta_{ij}^{(1)} Z_i Z_j + \sum_i \theta_i^{(2)} Z_i + \sum_i X_i, \qquad (22)$$

where $\theta^{(1)}$ represents the edge weights and $\theta^{(2)}$ is the weights on the nodes; and $\theta = \left\{\theta^{(1)}, \theta^{(2)}\right\}$. In this Ising Hamiltonian, the set of edges E indicates pairs of qubits with ZZ interactions. In the next step, turning the quadratic Ising Hamiltonian into quantum circuits by unitaries in the graph, which complicated unitary for a quantum computer, can be approximated by using Trotter–Suzuki decomposition as follows:[195,196]

$$U_{Ising} = e^{-it\hat{H}_{Ising}(\theta)} \approx \prod_{k=1}^{\frac{t}{\Delta}} \left[\prod_{j=1}^{Q} e^{-it\hat{H}_{Ising}^{j}(\theta)}\right], \qquad (23)$$

where $\hat{H}_{Ising}^{j}(\theta)$ represents the jth term of the Hamiltonian and Δ is a small number, indicating a specific example of a circuit of QGRNN by defining a variational ansatz of the form for a parametrized quadratic Hamiltonian $\hat{H}(\mu)$ as[195,196]

$$U_H(\mu, \gamma) = \prod_{i=1}^{P} \left[\prod_{j=1}^{Q} e^{-i\gamma_j \hat{H}^j(\mu)}\right]. \qquad (24)$$

1. Practical steps for implementation of QGRNN on PennyLane platform

To learn the dynamics of a quantum system, using the QGRNN ansatz assists in figuring out the time evolution of some quadratic Hamiltonian. Following the previous quadratic Hamiltonian example, one supposes the $\hat{H}_{Ising}(\alpha)$ with unknown target interaction graph G and unknown parameters (α), where low-energy states of the target Hamiltonian is accessible to quantum data as $|\psi_0$ as well as a set of time evolved states $|\psi(t_1)$, $|\psi(t_{12}), \ldots, |\psi(t_N)$, where[195,196]

$$|\psi(t_k) = e^{-it_k\hat{H}_{Ising}(\alpha)} |\psi_0, \qquad (25)$$

in the next step, a number of time evolved states have been picked from the collection of quantum data, and evolved states should compare to

$$U_{\hat{H}_{Ising}}(\mu, \Delta) |\psi_0 \approx e^{-it_k\hat{H}_{Ising}(\mu)} |\psi_0, \qquad (26)$$

in which one of the copies of $|\psi_0$ feeds to a quantum circuit with some guessed interaction graph and parameters (μ) by implementing the QGRNN ansatz. The last step will be figuring out the similarity of the time evolved states acquired by QGRNN with those of time evolved states from a classical optimizer, learning the unknown parameters of the Hamiltonian.[195,196]

IX. CHALLENGES AND OPPORTUNITIES

Quantum graph computing and learning hold significant potential in tackling complex learning tasks. However, several challenges exist, including the instability and complexity of quantum states and the scale limitations of quantum computational devices. While quantum graph learning addresses some limitations of classical graph learning, such as handling complex graph data through the integration of quantum theory, it still faces serious challenges, particularly with large-scale graphs. Future research should focus on improving encoding reliability, data-driven adaptivity, and computational efficiency. Preserving the structural information—features of edges and nodes—requires encoding graph data into quantum representations. Moreover, implementing nonlinear activation functions in deep neural networks involves considering nonlinear effects when embedding classical data, as linear unitary transformations drive quantum system evolution.

To realize the potential for quantum speedups as QGNN architectures are scaled, one must overcome numerous challenges despite significant advancements in QGNNs. To develop effective strategies

for enhancing quantum hardware, designing more practical algorithms, and ultimately proving performance guarantees, the limitations of QGNNs have been thoroughly characterized. These challenges have been addressed precisely, which can transition from theoretical to practical and robust steps toward quantum computing.

Noise and decoherence of quantum hardware as a main challenge can rigorously degrade the performance of quantum algorithms as well as QGNNs. The origin of noise is due to imperfect gate operations, environmental interactions, and readout errors, which can be local or uncorrelated and correlated types. Then, the loss of quantum information raised by these interactions creates decoherence, threatening delicate quantum states used in QGNNs. Various local (uncorrelated) noise channels such as bit flip, dephasing, and depolarizing introduce systematic bias in objective function evaluations, which cause to reevaluation of convergence analysis for optimization procedures.[199,200] Some correlated errors, such as non-Markovian 1/f noise,[201] interactions with environmental fluctuations,[202] and crosstalk,[203] which are prevalent in NISQ devices, complicate the implementation of QGNNs.

To mitigate noise, one strategy is based on reducing gate count in compiled quantum circuits, as different hardware architectures impact connectivity and circuit depth. In this approach, a promising research road map could be noise-resilient techniques that exploit graph data structure and figure out QGNN-specific graph-to-circuit. This practical approach could enhance graph connectivity for significantly efficient error correction strategies, leading to feasible QGNN implementations on NISQ devices.

Another critical challenge of QGNNs is **scalability and adaptability**. The quantum circuits by increasing the number of qubits and the size of the graph grow, which requires greater quantum hardware. The challenge of qubit demands grows linearly with graph size for current QGNN approaches, which causes analysis of large-scale graph NISQ devices to be impractical. Using topography-aware quantum circuit synthesis as one approach to address scalability for qubit-efficient encoding and mappings of graph data onto quantum circuits.[152,153,204,205] Unitary synthesis techniques with respect to hardware-specific qubit topologies can offer optimal gate counts. Moreover, strategies of circuit optimization, including reinforcement learning and parallelization, and gate pruning assist to manage execution times and circuit depth.[206–209] As a practical solution for scalability, hybrid quantum–classical approaches handle larger graphs by offloading parts of the computation to classical layers, which would be unfeasible to process on quantum hardware alone, and it has been successfully applied in refs.80,81,210

Other challenges of QGNNs are **lack of performance guarantees**. QGNN algorithms, unlike classical QAOA-based and ML algorithms, cannot be analyzed and bounded theoretically, and this gap originates from the inherent challenges in empirical validation and theoretical analysis to evaluate performance effectively. To address this gap, recent studies on VQAs have assembled bounds on the expressiveness of QNNs, which express conditions of quantum models to outperform classical counterpart.[211–213] The development of theoretical frameworks, including the role of entanglement and quantum correlations will be a promising research roadmap for providing concrete performance guarantees and enhancing the expressive power of QGNNs.

Barren plateau phenomenon, i.e., in training variational quantum algorithms (VQCs) for certain ansatz families, regions in the cost function landscape can be flat, where the gradients of the cost function become exponentially small or vanish relative to the number of qubits, is one of the challenges of training variational QNNs, including QGNNs.[214] In classical NNs, by increasing the number of layers, gradients vanish, while QNNs by growing the number of qubits encounter this issue more extensively. Even by applying gradient-free optimization algorithms for QGNN,[173] the training of quantum models of high-dimensional data might be hindered by barren plateau.[215] Quantum noise[216] and the design of the cost function[217] can exacerbate the barren plateau as well as the architecture of the quantum circuit. EQGCs, which are scaled effectively with the model as proposed in Ref. 76, are less biased to barren plateaus than other QNN architectures based on circuits with six to ten qubits. Nevertheless, barren plateaus are more likely to emerge in deeper circuits, requiring further empirical validation, where barren plateaus in QGNNs are extensively recognized. To mitigate this issue, several initialization and training strategies have been proposed,[218,219] but to fully resolve and understand the challenges raised by barren plateaus in variational quantum graph-based architectures, further development and research are mandatory.

A critical step to major impact on QGNNs performance is **quantum circuits' parameters initialization**. Effective initialization strategies are necessary as a good starting point for the parameters of quantum circuits, where restrains the barren plateaus[214] due to random initialization. The major topic of initialization with great impact, as shown by Ryu *et al.*[10] is not extensively covered in the research. For instance, as evidence of the significance of initialization strategies,[76] reported a failure in EDU-QGC model due to bad initialization and a poor 50% accuracy result, where unfortunate starting points with using only a small number of qubits (6–10) can lead to significant learning issues. Several heuristic-based initialization approaches can improve convergence in VQCs, such as initializing parameters based on using domain-specific knowledge or based on classical solutions.[218] Other techniques such as layer-wise training,[156] where parameters from shallow to deeper layers are initialized progressively, and Gaussian initialization,[219] which sets initial parameters by using Gaussian distributions, could provide promising alternatives. These techniques ensure more favorable starting conditions for the optimization process, which might help mitigate the barren plateau challenge.

QML methods aim to be easily applicable to various tasks and independent of experts. Thus, the future of quantum graph learning will rely on developing data-driven quantum graph algorithms to enhance quantum graph computing and learning capabilities. The non-convex nature of training quantum algorithms presents challenges related to the computational expense of optimizations and modeling quantum circuits. For instance, additional qubits and qubit entanglement are often required to embed structural information into quantum expressions. Addressing these challenges necessitates efficient gradient calculation through intelligent circuit design.

Image classification is a critical research field due to its wide-ranging practical applications. Machine learning (ML) is a powerful tool for tackling such complex tasks. Classical ML algorithms for image classification have achieved significant success in various applications by relying on manually designed feature extraction and classifier construction.[13–15] However, the performance of classical ML algorithms is increasingly limited by high-dimensional data and feature complexity.

Recently, with the rapid development of quantum computing, quantum image classification has emerged as a promising field, drawing significant interest from researchers. Applying QML algorithms to image classification offers the potential to reduce the computational costs associated with classical ML algorithms.[12,17,133,164] Leveraging the intrinsic features of quantum superposition and entanglement, QML algorithms exploit the power of large Hilbert spaces, fundamentally enhancing the accuracy and efficiency of image classification. This is achieved by mapping image data onto quantum circuits and states and utilizing quantum computing's parallelism and interference effects. As the circuit depth grows logarithmically with the number of input qubits and the gate parameters are learned, the QCNN model is expected to be well suited for NISQ devices.

Despite recent advancements in QML algorithms for image classification, the development of quantum computing hardware remains in its early stages, with a limited number of qubits available. This limitation impacts QML algorithm performance. Further investigation into the theoretical foundations and practical operations of QML algorithms is needed to fully harness the advantages of quantum information science for image classification. With the advancing capabilities of quantum computation, several perspectives can be anticipated soon:[17,139]

(i) **Enhanced efficiency**: QKNN and QCNN are poised to exhibit improved efficiency in image classification. Quantum-specific computing might lead to lower costs and more precise outcomes, promoting the adoption of QML techniques in various practical applications.

(ii) **Noise and error mitigation**: The implementation of QML algorithms is impacted by the noise and errors inherent in quantum computing. Developing robust quantum error correction codes and error mitigation methods can ensure the stability and reliability of QML algorithms on practical quantum computing devices.

(iii) **Generalized QCNN models**: Expanding QCNN circuit structures to classify higher-dimensional data with intrinsic topological order could help identify nonlocal parameter orders with low sample complexity. Relaxing translation-invariance constraints may result in $O(N\log(N))$ parameters for size N and employ ancilla qubits to implement parallel feature maps, inspired by classical CNN architecture.

(iv) **VQDNN models**: VQDNN models can achieve excellent classification accuracy and outperform classical counterparts in various QML scenarios, potentially implemented on NISQ devices with suitable encoding methods. Combining VQDNN with novel VQC architectures like QCNN can enhance image classification performance.

(v) **Quantum DCNN**: As one of the most representative models in deep learning, classical DCNN is widely used across diverse fields, including computer vision, speech recognition, and natural language processing. Despite these achievements, computational costs increase dramatically with the growth in layer width and depth, representing a primary bottleneck in deep learning. QDCNN presents a potential parallel device to improve classical computing efficiency. Further development of QDCNN algorithms is needed to integrate QCNN with conventional deep learning, fully leveraging quantum computing characteristics to enhance image classification performance.

X. CONCLUSION AND FUTURE DIRECTIONS

This paper provides a comprehensive review of the rapidly evolving field of quantum graph computing and quantum graph learning, showcasing their potential to solve complex computational problems in quantum computing, materials discovery, and image classification. By integrating graph theory with quantum computing, researchers can harness the unique properties of quantum mechanics, such as superposition and entanglement, to achieve exponential speedups and tackle problems beyond the capabilities of classical systems.

Despite the promising advancements, significant challenges remain. Current quantum hardware limitations and the scalability of quantum algorithms pose obstacles to the widespread adoption of these technologies. Future research should focus on developing scalable quantum algorithms that efficiently handle large-scale graph data and complex network structures. Additionally, advancing quantum hardware technology will be essential to fully realize the potential of quantum graph learning.

Practical applications of quantum graph computing and learning are diverse and impactful. In materials science, quantum graph neural networks could significantly enhance materials discovery processes by accurately predicting chemical and physical properties. In the realm of image classification, quantum machine learning algorithms offer improved performance for autonomous systems, medical diagnostics, and security applications. As quantum computing technology matures, its integration with graph learning will drive innovation across scientific and industrial sectors.

In summary, the intersection of quantum computing and graph theory presents new opportunities for research and application, offering solutions to some of the most challenging problems in computational science. By addressing current limitations and exploring novel methodologies, quantum graph computing and learning are poised to play a pivotal role in the future of technology and innovation.

ACKNOWLEDGMENTS

Y.D. was in part supported by the National Energy Technology Laboratory (NETL) Laboratory Directed Research and Development (LDRD) program (# 1024903). Neither the United States Government nor any agency thereof, nor any of its employees, nor the support contractor, nor any of their employees, make any warranty, expressor implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Farzaneh Shayeganfar: Data curation (equal); Formal analysis (equal); Investigation (equal); Writing – original draft (equal). **Ali Ramazani:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Resources (equal); Supervision (equal); Validation (equal); Writing – original draft (equal). **Veera Sundararaghavan:** Conceptualization (equal); Funding acquisition (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal). **Yuhua Duan:** Conceptualization (equal); Data curation (equal); Investigation (equal); Methodology (equal); Resources (equal); Validation (equal); Writing – original draft (equal).

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

REFERENCES

[1]R. P. Feynman, "Simulating physics with computers," Int. J. Theor. Phys. **21**(6), 467–488 (1982).
[2]P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," J. Stat. Phys. **22**(5), 563–591 (1980).
[3]P. Benioff, "Quantum mechanical models of Turing machines that dissipate no energy," Phys. Rev. Lett. **48**(23), 1581–1585 (1982).
[4]A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," IEEE Trans. Neural Networks **8**(3), 714–735 (1997).
[5]M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," paper presented at the Proceedings of 2005 IEEE International Joint Conference on Neural Networks (2005).
[6]F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Trans. Neural Networks **20**(1), 61–80 (2009).
[7]G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nat. Rev. Phys. **3**(6), 422–440 (2021).
[8]S. Yu, C. Peng, Y. Wang, A. Shehzad, F. Xia, and E. R. Hancock, "Quantum graph learning: Frontiers and outlook," arXiv:2302.00892 (2023).
[9]G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, "Quantum graph neural networks," arXiv:1909.12264 (2019).
[10]J.-Y. Ryu, E. Elala, and J.-K. K. Rhee, "Quantum graph neural network models for materials search," Materials **16**(12), 4300 (2023).
[11]H. I. Garcia-Hernandez, R. Torres-Ruiz, and G.-H. Sun, "Image classification via quantum machine learning," arXiv:2011.02831 (2020).
[12]R. Kharsa, A. Bouridane, and A. Amira, "Advances in quantum machine learning and deep learning for image classification: A survey," Neurocomputing **560**, 126843 (2023).
[13]H. Xiong, X. Duan, Y. Yu, J. Zhang, and H. Yin, "Image classification based on quantum machine learning," paper presented at the 5th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP) (2023).
[14]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**, 42–53 (2023).
[15]Z. Zhang, X. Mi, J. Yang, X. Wei, Y. Liu, J. Yan, P. Liu, X. Gu, and T. Yu, "Remote sensing image scene classification in hybrid classical–quantum transferring CNN with small samples," Sensors **23**(18), 8010 (2023).
[16]E. H. Houssein, Z. Abohashima, M. Elhoseny, and W. M. Mohamed, "Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision," Expert Syst. Appl. **194**, 116512 (2022).
[17]Q. Li, Z. Huang, H. Yang, and M. Song, "A review of quantum machine learning for image classification," in 2nd International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology, Nanjing, P. R. China (2023).
[18]X.-W. Yao, H. Wang, Z. Liao, M.-C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo, W. Zheng, J. Li, M. Zhao, X. Peng, and D. Suter, "Quantum image processing and its application to edge detection: Theory and experiment," Phys. Rev. X **7**(3), 031041 (2017).
[19]L.-P. Henry, S. Thabet, C. Dalyac, and L. Henriet, "Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits," Phys. Rev. A **104**(3), 032416 (2021).
[20]K. Kishi, T. Satoh, R. Raymond, N. Yamamoto, and Y. Sakakibara, "Graph kernels encoding features of all subgraphs by quantum superposition," IEEE J. Emerging Sel. Top. Circuits Syst. **12**(3), 602–613 (2022).
[21]J. Kempe, "Quantum random walks: An introductory overview," Contemp. Phys. **44**(4), 307–327 (2003).
[22]S. E. Venegas-Andraca, "Quantum walks: A comprehensive review," Quantum Inf. Process. **11**(5), 1015–1106 (2012).
[23]T. Albash and D. A. Lidar, "Adiabatic quantum computation," Rev. Mod. Phys. **90**(1), 015002 (2018).
[24]S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," Nat. Phys. **14**(6), 595–600 (2018).
[25]T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," Phys. Rev. E **58**(5), 5355–5363 (1998).
[26]C. Silva, A. Aguiar, P. M. V. Lima, and I. Dutra, "Mapping graph coloring to quantum annealing," Quantum Mach. Intell. **2**(2), 16 (2020).
[27]D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, 2014), pp. 283–292.
[28]A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, "Toward the first quantum simulation with quantum speedup," Proc. Natl. Acad. Sci. U. S. A. **115**(38), 9456–9461 (2018).
[29]E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," arXiv:1411.4028 (2014).
[30]S. Morita and H. Nishimori, "Mathematical foundation of quantum annealing," J. Math. Phys. **49**(12), 125210 (2008).
[31]S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science **220**(4598), 671–680 (1983).
[32]R. Y. Li, R. Di Felice, R. Rohs, and D. A. Lidar, "Quantum annealing versus classical machine learning applied to a simplified computational biology problem," npj Quantum Inf. **4**(1), 14 (2018).
[33]S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum annealing for industry applications: Introduction and review," Rep. Prog. Phys. **85**(10), 104001 (2022).
[34]P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, "Perspectives of quantum annealing: Methods and implementations," Rep. Prog. Phys. **83**(5), 054401 (2020).
[35]R. B. Stinchcombe, "Ising model in a transverse field. I. Basic theory," J. Phys. C **6**(15), 2459 (1973).
[36]Ü. V. Çatalyürek, J. Feo, A. H. Gebremedhin, M. Halappanavar, and A. Pothen, "Graph coloring algorithms for multi-core and massively multithreaded architectures," Parallel Comput. **38**(10), 576–594 (2012).
[37]E. D'Hondt, "Quantum approaches to graph colouring," Theor. Comput. Sci. **410**(4), 302–309 (2009).
[38]S. Srivastava, "Graph theoretic algorithms adaptable to quantum computing," Ph.D. thesis (University of Michigan, 2021).
[39]V. Sundararaghavan and N. Zabaras, "A statistical learning approach for the design of polycrystalline materials," Stat. Anal. **1**(5), 306–321 (2009).
[40]P. Acar and V. Sundararaghavan, "Utilization of a linear solver for multiscale design and optimization of microstructures," AIAA J. **54**(5), 1751–1759 (2016).
[41]S. Sun and V. Sundararaghavan, "A probabilistic crystal plasticity model for modeling grain shape effects based on slip geometry," Acta Mater. **60**(13), 5233–5244 (2012).
[42]S. Sun and V. Sundararaghavan, "A peridynamic implementation of crystal plasticity," Int. J. Solids Struct. **51**(19), 3350–3360 (2014).
[43]V. Sundararaghavan and S. Srivastava, "MicroFract: An image based code for microstructural crack path prediction," SoftwareX **6**, 94–97 (2017).
[44]R. Quey, P. R. Dawson, and F. Barbe, "Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing," Comput. Methods Appl. Mech. Eng. **200**(17), 1729–1745 (2011).

[45]K. Shimizu and R. Mori, "Exponential-time quantum algorithms for graph coloring problems," Algorithmica **84**(12), 3603–3621 (2022).
[46]S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, "Hybrid quantum-classical algorithms for approximate graph coloring," Quantum **6**, 678 (2022).
[47]A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," Phys. Rev. Lett. **103**(15), 150502 (2009).
[48]B. Zanger, C. B. Mendl, M. Schulz, and M. Schreiber, "Quantum algorithms for solving ordinary differential equations via classical integration methods," Quantum **5**, 502 (2021).
[49]J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," arXiv:1312.6203v3 (2013).
[50]S. Srivastava and V. Sundararaghavan, "Box algorithm for the solution of differential equations on a quantum annealer," Phys. Rev. A **99**(5), 052355 (2019).
[51]S. Yu, H. Huang, M. N. Dao, and F. Xia, "Graph augmentation learning," in *Companion Proceedings of the Web Conference 2022* (Association for Computing Machinery, Virtual Event, Lyon, France, 2022), pp. 1063–1072.
[52]F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," IEEE Trans. Artif. Intell. **2**(2), 109–127 (2021).
[53]S. Yu, F. Xia, Y. Sun, T. Tang, X. Yan, and I. Lee, "Detecting outlier patterns with query-based artificially generated searching conditions," IEEE Trans. Comput. Soc. Syst. **8**(1), 134–147 (2021).
[54]F. Xia, S. Yu, C. Liu, J. Li, and I. Lee, "CHIEF: Clustering with higher-order motifs in big networks," IEEE Trans. Network Sci. Eng. **9**(3), 990–1005 (2022).
[55]S. Yu, J. Ren, S. Li, M. Naseriparsa, and F. Xia, "Graph learning for fake review detection," Front. Artif. Intell. **5**, 922589 (2022).
[56]S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (Association for Computing Machinery, Melbourne, Australia, 2015), pp. 891–900.
[57]S. Miao, M. Liu, and P. Li, "Interpretable and generalizable graph learning via stochastic attention mechanism," in *39th International Conference on Machine Learning* (MLR Press, Baltimore, MD, USA, 2022), Vol. PMLR 162, pp. 15524–15543.
[58]P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," paper presented at the Proceedings 35th Annual Symposium on Foundations of Computer Science (1994).
[59]L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, Philadelphia, PA, USA, 1996), pp. 212–219.
[60]Y.-H. Luo, H.-S. Zhong, M. Erhard, X.-L. Wang, L.-C. Peng, M. Krenn, X. Jiang, L. Li, N.-L. Liu, C.-Y. Lu, A. Zeilinger, and J.-W. Pan, "Quantum teleportation in high dimensions," Phys. Rev. Lett. **123**(7), 070505 (2019).
[61]S. Yu, Y. Feng, D. Zhang, H. D. Bedru, B. Xu, and F. Xia, "Motif discovery in networks: A survey," Comput. Sci. Rev. **37**, 100267 (2020).
[62]S. Yu, J. Xu, C. Zhang, F. Xia, Z. Almakhadmeh, and A. Tolba, "Motifs in big networks: Methods and applications," IEEE Access **7**, 183322–183338 (2019).
[63]J. Xu, S. Yu, K. Sun, J. Ren, I. Lee, S. Pan, and F. Xia, "Multivariate relations aggregation learning in social networks," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (Association for Computing Machinery, Virtual Event, China, 2020), pp. 77–86.
[64]M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).
[65]M. Rath and H. Date, "Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy," arXiv:2311.10375v1 (2023).
[66]Z. Tabi, K. H. El-Safty, Z. Kallus, P. Haga, T. Kozsik, A. Glos, and Z. Zimboras, "Quantum optimization for the graph coloring problem with space-efficient embedding," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, 2020), pp. 56–62.
[67]E. Pelofske, G. Hahn, and H. Djidjev, "Decomposition algorithms for solving NP-hard problems on a quantum annealer," J. Signal Process. Syst. **93**(4), 405–420 (2021).
[68]D. Wang, K. Zheng, F. Teng, and Q. Chen, "Quantum annealing with integer slack variables for grid partitioning," IEEE Trans. Power Syst. **38**(2), 1747–1750 (2023).
[69]Z. Zhang, D. Chen, J. Wang, L. Bai, and E. R. Hancock, "Quantum-based subgraph convolutional neural networks," Pattern Recognit. **88**, 38–49 (2019).
[70]G. Yan, Y. Tang, and J. Yan, "Towards a native quantum paradigm for graph representation learning: A sampling-based recurrent embedding approach," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, Washington, DC, USA, 2022), pp. 2160–2168.
[71]L. Bai, Y. Jiao, L. Cui, L. Rossi, Y. Wang, P. S. Yu, and E. R. Hancock, "Learning graph convolutional networks based on quantum vertex information propagation," IEEE Trans. Knowl. Data Eng. **35**(2), 1747–1760 (2023).
[72]L. Bai, L. Rossi, A. Torsello, and E. R. Hancock, "A quantum Jensen–Shannon graph kernel for unattributed graphs," Pattern Recognit. **48**(2), 344–355 (2015).
[73]M. Schuld, K. Brádler, R. Israel, D. Su, and B. Gupt, "Measuring the similarity of graphs with a Gaussian boson sampler," Phys. Rev. A **101**(3), 032314 (2020).
[74]B. Collis, S. Patel, D. Koch, M. Cutugno, L. Wessing, and P. M. Alsing, "Physics simulation via quantum graph neural network," AVS Quantum Sci. **5**, 023801 (2023).
[75]Y. Chen, C. Wang, H. Guo, and J. Wu, "Novel architecture of parameterized quantum circuit for graph convolutional network," arXiv:2203.03251 (2022).
[76]P. Mernyei, K. Meichanetzidis, and İ. Ceylan, "Equivariant quantum graph circuits: Constructions for universal approximation over graphs," Quantum Mach. Intell. **5**(1), 6 (2023).
[77]J. Zheng, Q. Gao, and L. Y, "Quantum graph convolutional neural networks," paper presented at the 40th Chinese Control Conference (CCC) (2021), pp. 6335–6340.
[78]K. Beer, M. Khosla, J. Köhler, T. J. Osborne, and T. Zhao, "Quantum machine learning of graph-structured data," Phys. Rev. A **108**(1), 012410 (2023).
[79]S. Dernbach, A. Mohseni-Kabir, S. Pal, and D. Towsley, "Quantum walk neural networks for graph-structured data," paper presented at the Complex Networks and Their Applications VII, Cham (2019), pp. 182–193.
[80]S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Hybrid quantum-classical graph convolutional network," arXiv:2101.06189 (2021).
[81]C. Tüysüz, C. Rieger, K. Novotny, B. Demirköz, D. Dobos, K. Potamianos, S. Vallecorsa, J.-R. Vlimant, and R. Forster, "Hybrid quantum classical graph neural networks for particle track reconstruction," Quantum Mach. Intell. **3**(2), 29 (2021).
[82]S. S. Du, K. Hou, R. R. Salakhutdinov, B. Poczos, R. Wang, and K. Xu, "Graph neural tangent kernel: Fusing graph neural networks with graph kernels," paper presented at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada (2019).
[83]X. Ai, Z. Zhang, L. Sun, J. Yan, and E. Hancock, "Towards quantum graph neural networks: An ego-graph learning approach," arXiv:2201.05158 (2022).
[84]Y. Tang, J. Yan, and H. Edwin, "From quantum graph computing to quantum graph learning: A survey," arXiv:2202.09506 (2022).
[85]A. Lucas, "Ising formulations of many NP problems," Front. Phys. **2**, 5 (2014).
[86]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), 2631 (2021).
[87]M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," arXiv:2104.13478 (2021).
[88]P. Veličković, "Everything is connected: Graph neural networks," Curr. Opin. Struct. Biol. **79**, 102538 (2023).
[89]P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. F. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, Ç. Gülçehre, H. F. Song, A. J. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. R. Allen, C. Nash, V. Langston, C. Dyer, N. M. O. Heess, D. Wierstra, P. Kohli, M. M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks," arXiv:1806.01261 (2018).
[90]Y. Liao, X.-M. Zhang, and C. Ferrie, "Graph neural networks on quantum computers," arXiv:2405.17060v1 (2024).
[91]T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv:1609.02907 (2017).

[92]L. Wu, P. Cui, J. Pei, L. Zhao, and X. Guo, "Graph neural networks: Foundation, frontiers and applications," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, Long Beach, CA, USA, 2023), pp. 5831–5832.

[93]W. L. Hamilton, R. Ying, and J. Leskovec, "*Inductive representation learning on large graphs*," in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., Long Beach, CA, USA, 2017), pp. 1025–1035.

[94]J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning* (JMLR.org, Sydney, NSW, Australia, 2017), pp. 1263–1272.

[95]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), 6329 (2021).

[96]P. Mernyei, K. Meichanetzidis, and I. I. Ceylan, "Equivariant quantum graph circuits," in *39th International Conference on Machine Learning* (MLR Press, Baltimore, Maryland, USA, 2022), Vol. PMLR 162, pp. 15401–15420.

[97]T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," Phys. Rev. Lett. **120**(14), 145301 (2018).

[98]M. Dai, M. F. Demirel, Y. Liang, and J.-M. Hu, "Graph neural networks for an accurate and interpretable prediction of the properties of polycrystalline materials," npj Comput. Mater. **7**(1), 103 (2021).

[99]Z. Yang and M. J. Buehler, "Linking atomic structural defects to mesoscale properties in crystalline solids using graph neural networks," npj Comput. Mater. **8**(1), 198 (2022).

[100]P. C. St. John, C. Phillips, T. W. Kemper, A. N. Wilson, Y. Guan, M. F. Crowley, M. R. Nimlos, and R. E. Larsen, "Message-passing neural networks for high-throughput polymer screening," J. Chem. Phys. **150**(23), 234111 (2019).

[101]M. Zeng, J. Kumar, Z. Zeng, S. Ramasamy, V. Chandrasekhar, and K. Hippalgaonkar, "Graph convolutional neural networks for polymers property prediction," arXiv:1811.06231 (2018).

[102]M. Aldeghi and C. W. Coley, "A graph representation of molecular ensembles for polymer property prediction," Chem. Sci. **13**(35), 10486–10498 (2022).

[103]K. Swanson, S. Trivedi, J. Lequieu, K. Swanson, and R. Kondor, "Deep learning for automated classification and characterization of amorphous materials," Soft Matter **16**(2), 435–446 (2020).

[104]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**(11), 139–144 (2020).

[105]D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," arXiv:1312.6114 (2013).

[106]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**(7747), 209–212 (2019).

[107]M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," Phys. Rev. Lett. **122**(4), 040504 (2019).

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

[109]S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," Nat. Phys. **10**(9), 631–633 (2014).

[110]P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," Phys. Rev. Lett. **113**(13), 130503 (2014).

[111]N. Wiebe, D. Braun, and S. Lloyd, "Quantum algorithm for data fitting," Phys. Rev. Lett. **109**(5), 050505 (2012).

[112]M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," Quantum Sci. Technol. **4**(4), 043001 (2019).

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

[114]M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," Nat. Rev. Phys. **3**(9), 625–644 (2021).

[115]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**(3), 032430 (2021).

[116]S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, "Quantum machine learning beyond kernel methods," Nat. Commun. **14**(1), 517 (2023).

[117]S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, "Quantum embeddings for machine learning," arXiv:2001.03622 (2020).

[118]I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," Nat. Phys. **15**(12), 1273–1278 (2019).

[119]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), 808 (2020).

[120]K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. L. Billinge, E. Holm, S. P. Ong, and C. Wolverton, "Recent advances and applications of deep learning methods in materials science," npj Comput. Mater. **8**(1), 59 (2022).

[121]J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," Quantum Sci. Technol. **2**(4), 045001 (2017).

[122]S. Lloyd and C. Weedbrook, "Quantum generative adversarial learning," Phys. Rev. Lett. **121**(4), 040502 (2018).

[123]C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," npj Quantum Inf. **5**(1), 103 (2019).

[124]A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, "Transfer learning in hybrid classical-quantum neural networks," Quantum **4**, 340 (2020).

[125]N. Schetakis, D. Aghamalyan, P. Griffin, and M. Boguslavsky, "Review of some existing QML frameworks and novel hybrid classical–quantum neural networks realising binary classification for the noisy datasets," Sci. Rep. **12**(1), 11927 (2022).

[126]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**(3), 031010 (2022).

[127]D. Arthur and P. Date, "A hybrid quantum-classical neural network architecture for binary classification," arXiv:2201.01820 (2022).

[128]M. Liu, J. Liu, R. Liu, H. Makhanov, D. Lykov, A. Apte, and Y. Alexeev, "Embedding learning in hybrid quantum-classical neural networks," paper presented at the *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2022), pp. 79–86.

[129]K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," Phys. Rev. A **98**(3), 032309 (2018).

[130]M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," Phys. Rev. A **99**(3), 032331 (2019).

[131]T. Jones and J. Gacon, "Efficient calculation of gradients in classical simulations of variational quantum algorithms," arXiv:2009.02823 (2020).

[132]Y. Ruan, X. Xue, and Y. Shen, "Quantum image processing: Opportunities and challenges," Math. Probl. Eng. **2021**, 6671613.

[133]Y. Wang, Y. Wang, C. Chen, R. Jiang, and W. Huang, "Development of variational quantum deep neural networks for image recognition," Neurocomputing **501**, 566–582 (2022).

[134]F. Ablayev, M. Ablayev, J. Z. Huang, K. Khadiev, N. Salikhova, and D. Wu, "On quantum methods for machine learning problems. Part II: Quantum classification algorithms," Big Data Min. Anal. **3**(1), 56–67 (2020).

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

[136]T. Suzuki, T. Hasebe, and T. Miyazaki, "Quantum support vector machines for classification and regression on a trapped-ion quantum computer," Quantum Mach. Intell. **6**(1), 31 (2024).

[137]H.-M. Rieser, F. Köster, and A. P. Raulf, "Tensor networks for quantum machine learning," Proc. R. Soc. A **479**(2275), 20230218 (2023).

[138]D. Guala, S. Zhang, E. Cruz, C. A. Riofrío, J. Klepsch, and J. M. Arrazola, "Practical overview of image classification with tensor-network quantum circuits," Sci. Rep. **13**(1), 4427 (2023).

[139]Y. Li, R.-G. Zhou, R. Xu, J. Luo, and W. Hu, "A quantum deep convolutional neural network for image recognition," Quantum Sci. Technol. **5**(4), 044003 (2020).

[140]M. T. West, M. Sevior, and M. Usman, "Reflection equivariant quantum neural networks for enhanced image classification," Mach. Learn. **4**(3), 035027 (2023).

[141]W. Li, P.-C. Chu, G.-Z. Liu, Y.-B. Tian, T.-H. Qiu, and S.-M. Wang, "An image classification algorithm based on hybrid quantum classical convolutional neural network," Quantum Eng. **2022**, 5701479.
[142]L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," IEEE Signal Process. Mag. **29**(6), 141–142 (2012).
[143]H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," arXiv:1708.07747 (2017).
[144]M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, "A generative modeling approach for benchmarking and training shallow quantum circuits," npj Quantum Inf. **5**(1), 45 (2019).
[145]F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," Comput. Vision Image Understanding **106**(1), 59–70 (2007).
[146]A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 416–431 (2006).
[147]A. Masca and G. D. Magoulas, "Learning input features representations in deep learning," in *Advances in Computational Intelligence Systems. Advances in Intelligent Systems and Computing*, edited by P. Angelov, A. Gegov, C. Jayne, and Q. Shen (Springer, Cham, 2017), Vol. 513, pp. 433–448.
[148]J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," Neural Networks **32**, 323–332 (2012).
[149]R. Golchha and G. K. Verma, "Quantum-enhanced support vector classifier for image classification," paper presented at the IEEE 8th International Conference for Convergence in Technology (I2CT) (2023).
[150]Z. Li, X. Liu, N. Xu, and J. Du, "Experimental realization of a quantum support vector machine," Phys. Rev. Lett. **114**(14), 140504 (2015).
[151]Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, "Image classification based on quantum K-Nearest-Neighbor algorithm," Quantum Inf. Process. **17**(9), 239 (2018).
[152]N.-R. Zhou, X.-X. Liu, Y.-L. Chen, and N.-S. Du, "Quantum K-nearest-neighbor image classification algorithm based on K-L transform," Int. J. Theor. Phys. **60**(3), 1209–1224 (2021).
[153]Y. Ruan, X. Xue, H. Liu, J. Tan, and X. Li, "Quantum algorithm for K-nearest neighbors classification based on the metric of hamming distance," Int. J. Theor. Phys. **56**(11), 3496–3507 (2017).
[154]M. Lazzarin, D. E. Galli, and E. Prati, "Multi-class quantum classifiers with tensor network circuits for quantum phase recognition," Phys. Lett. A **434**, 128056 (2022).
[155]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), 65 (2018).
[156]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), 5 (2021).
[157]R. Potempa and S. Porebski, "Comparing concepts of quantum and classical neural network models for image classification task," paper presented at the Progress in Image Processing, Pattern Recognition and Communication Systems, Cham (2022).
[158]R. Huang, X. Tan, and Q. Xu, "Variational quantum tensor networks classifiers," Neurocomputing **452**, 89–98 (2021).
[159]S. Wei, Y. Chen, Z. Zhou, and G. Long, "A quantum convolutional neural network on NISQ devices," AAPPS Bull. **32**(1), 2 (2022).
[160]A. K. Ahmed, B. Tantawi, and G. I. Sayed, "Multiclass image classification based on Quantum-Inspired convolutional neural network," paper presented at the 6th International Workshop on Computer Modeling and Intelligent Systems (CMIS 2023), Zaporizhzhia, Ukraine (2023), pp. 177–187.
[161]T. Hur, L. Kim, and D. K. Park, "Quantum convolutional neural network for classical data classification," Quantum Mach. Intell. **4**(1), 3 (2022).
[162]D. Bokhan, A. S. Mastiukova, A. S. Boev, D. N. Trubnikov, and A. K. Fedorov, "Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning," Front. Phys. **10**, 1069985 (2022).
[163]Y. Lü, Q. Gao, J. Lü, M. Ogorzałek, and J. Zheng, "A quantum convolutional neural network for image classification," paper presented at the 40th Chinese Control Conference (CCC) (2021), pp. 6329–6334.
[164]A. Senokosov, A. Sedykh, A. Sagingalieva, B. Kyriacou, and A. Melnikov, "Quantum machine learning for image classification," Mach. Learn. **5**(1), 015040 (2024).
[165]S. Amrouche, L. Basara, P. Calafiura, D. Emeliyanov, V. Estrade, S. Farrell, C. Germain, V. V. Gligorov, T. Golling, S. Gorbunov, H. Gray, I. Guyon, M. Hushchyn, V. Innocente, M. Kiehn, M. Kunze, E. Moyse, D. Rousseau, A. Salzburger, A. Ustyuzhanin, and J.-R. Vlimant, "The tracking machine learning challenge: Throughput phase," Comput. Software Big Sci. **7**(1), 1 (2023).
[166]G. Apollinari, O. Brüning, T. Nakamoto, and L. Rossi, "High luminosity large hadron collider HL-LHC," CERN Yellow Report FERMILAB-PUB-15-699-TD (2015), Chap 1, pp. 1–19.
[167]C. Tüysüz, B. Demirkoz, D. Dobos, F. Fracas, F. Carminati, J.-R. Vlimant, K. Potamianos, K. Novotny, S. Vallecorsa, and C. Tüysüz, "CTD2020: A quantum graph network approach to particle track reconstruction," in *Connecting the Dots Workshop 2020 (CTD2020)* (Zenodo, 2020).
[168]L. E. Ratcliff, S. Mohr, G. Huhs, T. Deutsch, M. Masella, and L. Genovese, "Challenges in large scale quantum mechanical calculations," WIREs Comput. Mol. Sci. **7**(1), e1290 (2017).
[169]A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, "Commentary: The materials project: A materials genome approach to accelerating materials innovation," APL Mater. **1**(1), 4812323 (2013).
[170]J. Liu, K. Kawaguchi, B. Hooi, Y. Wang, and X. Xiao, "EIGNN: Efficient infinite-depth graph neural networks," in *Proceedings of the 35th International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2021), Article 1434.
[171]A. S. Bhatia, M. K. Saggi, and S. Kais, "Quantum machine learning predicting ADME-Tox properties in drug discovery," J. Chem. Inf. Model. **63**(21), 6476–6486 (2023).
[172]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. **4**(1), 015023 (2023).
[173]M. Vitz, H. Mohammadbagherpoor, S. Sandeep, A. Vlasic, R. Padbury, and A. Pham, "Hybrid quantum graph neural network for molecular property prediction," arXiv:2405.05205 (2024).
[174]I. E. Castelli, D. D. Landis, K. S. Thygesen, S. Dahl, I. Chorkendorff, T. F. Jaramillo, and K. W. Jacobsen, "New cubic perovskites for one- and two-photon water splitting using the computational materials repository," Energy Environ. Sci. **5**(10), 9034–9043 (2012).
[175]M. D. Parenti and G. Rastelli, "Advances and applications of binding affinity prediction methods in drug discovery," Biotechnol. Adv. **30**(1), 244–250 (2012).
[176]J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao, "Applications of machine learning in drug discovery and development," Nat. Rev. Drug Discovery **18**(6), 463–477 (2019).
[177]Y. Cao, J. Romero, and A. Aspuru-Guzik, "Potential of quantum computing for drug discovery," IBM J. Res. Dev. **62**(6), 6:1–6:20 (2018).
[178]*wwPDB consortium.* "Protein Data Bank: The single global archive for 3D macromolecular structure data," Nucleic Acids Res. **47**(D1), D520–D528 (2019).
[179]D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI, 2020), Vol. 34(4), pp. 3438–3445.
[180]J. Topping, F. D. Giovanni, B. P. Chamberlain, X. Dong, and M. M. Bronstein, "Understanding over-squashing and bottlenecks on graphs via curvature," arXiv:2111.14522 (2022).
[181]A. Ray, D. Madan, S. Patil, M. A. Rapsomaniki, and P. Pati, "Hybrid quantum-classical graph neural networks for tumor classification in digital pathology," arXiv:2310.11353 (2023).
[182]P. Pati, G. Jaume, A. Foncubierta, F. Feroce, A. M. Anniciello, G. Scognamiglio, N. Brancati, M. Fiche, E. Dubruc, D. Riccio, M. d. Bonito, G. D. Pietro, G. Botti, J.-P. Thiran, M. Frucci, O. Goksel, and M. Gabrani, "Hierarchical graph representations in digital pathology," Med. Image Anal. **75**, 102264 (2022).
[183]M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," IEEE Trans. Intell. Transp. Syst. **21**(8), 3152–3168 (2020).

[184]P. Jiang, Z. Liu, L. Zhang, and J. Wang, "Advanced traffic congestion early warning system based on traffic flow forecasting and extenics evaluation," Appl. Soft Comput. **118**, 108544 (2022).
[185]K. Novotny, C. Tüysüz, C. Rieger, D. Dobos, K. Potamianos, B. Demirkoz, F. Carminati, S. Vallecorsa, J. R. Vlimant, and F. Fracas, "Quantum track reconstruction algorithms for non-HEP applications," in Proceedings of 40th International Conference on High Energy Physics—PoS (ICHEP2020) (2021).
[186]S. G. H. Philander, "El Niño Southern Oscillation phenomena," Nature **302**(5906), 295–301 (1983).
[187]H. J. Kimble, "The quantum internet," Nature **453**(7198), 1023–1030 (2008).
[188]K. Qian, Z. Eldredge, W. Ge, G. Pagano, C. Monroe, J. V. Porto, and A. V. Gorshkov, "Heisenberg-scaling measurement protocol for analytic functions with quantum sensor networks," Phys. Rev. A **100**(4), 042304 (2019).
[189]D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond Bell's theorem," in *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, edited by M. Kafatos (Springer Netherlands, Dordrecht, 1989), pp. 69–72.
[190]J. Choi, S. Oh, and J. Kim, "A tutorial on quantum graph recurrent neural network (QGRNN)," paper presented at the *International Conference on Information Networking (ICOIN)* (IEEE, 2021), pp. 46–49.
[191]N. Wiebe, C. Granade, C. Ferrie, and D. G. Cory, "Hamiltonian learning and certification using quantum resources," Phys. Rev. Lett. **112**(19), 190501 (2014).
[192]C. Papalitsas, T. Andronikos, K. Giannakis, G. Theocharopoulou, and S. Fanarioti, "A QUBO model for the traveling salesman problem with time windows," Algorithms **12**(11), 224 (2019).
[193]M. Z. Alom, B. V. Essen, A. T. Moody, D. P. Widemann, and T. M. Taha, "Quadratic unconstrained binary optimization (QUBO) on neuromorphic computing system," paper presented at the International Joint Conference on Neural Networks (IJCNN) (2017).
[194]See https://qiskit.org (2022).
[195]See https://qiskit.org/documentation/stubs/qiskit.circuit.QuantumCircuit.html (2023).
[196]See https://qiskit.org/documentation/stubs/qiskit.circuit.QuantumRegister.html (2023).
[197]See https://qiskit.org/documentation/stubs/qiskit.quantum_info.Statevector.html (2023).
[198]A.-G. Tudorache, "Graph generation for quantum states using Qiskit and its application for quantum neural networks," Mathematics **11**(6), 1484 (2023).
[199]E. Fontana, N. Fitzpatrick, D. M. Ramo, R. Duncan, and I. Rungger, "Evaluating the noise resilience of variational quantum algorithms," Phys. Rev. A **104**(2), 022403 (2021).
[200]V. Kungurtsev, G. Korpas, J. Marecek, and E. Y. Zhu, "Iteration complexity of variational quantum algorithms," Quantum **8**, 1495 (2024).
[201]G. Burkard, "Non-Markovian qubit dynamics in the presence of 1/*f* noise," Phys. Rev. B **79**(12), 125317 (2009).
[202]S. Schlör, J. Lisenfeld, C. Müller, A. Bilmes, A. Schneider, D. P. Pappas, A. V. Ustinov, and M. Weides, "Correlating decoherence in transmon qubits: Low frequency noise by single fluctuators," Phys. Rev. Lett. **123**(19), 190502 (2019).
[203]I. Heinz and G. Burkard, "Crosstalk analysis for single-qubit and two-qubit gates in spin qubit arrays," Phys. Rev. B **104**(4), 045420 (2021).
[204]M. G. Davis, E. Smith, A. Tudor, K. Sen, I. Siddiqi, and C. Iancu, "Towards optimal topology aware quantum circuit synthesis," paper presented at the IEEE International Conference on Quantum Computing and Engineering (QCE) (2020).
[205]M. Weiden, J. Kalloor, J. Kubiatowicz, E. Younis, and C. Iancu, "Wide quantum circuit optimization with topology aware synthesis," paper presented at the IEEE/ACM Third International Workshop on Quantum Computing Software (QCS) (2022).
[206]Y. Nam, N. J. Ross, Y. Su, A. M. Childs, and D. Maslov, "Automated optimization of large quantum circuits with continuous parameters," npj Quantum Inf. **4**(1), 23 (2018).
[207]G. Yan, W. Wu, Y. Chen, K. Pan, X. Lu, Z. Zhou, Y. Wang, R. Wang, and J. Yan, "Quantum circuit synthesis and compilation optimization: Overview and prospects," arXiv:2407.00736 (2024).
[208]S. Sim, J. Romero, J. F. Gonthier, and A. A. Kunitsa, "Adaptive pruning-based optimization of parameterized quantum circuits," Quantum Sci. Technol. **6**(2), 025019 (2021).
[209]M. Ostaszewski, L. M. Trenkwalder, W. Masarczyk, E. Scerri, and V. Dunjko, "Reinforcement learning for optimization of variational quantum circuit architectures," in *Proceedings of the 35th International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2021), Article 1391.
[210]F. Mauro, A. Sebastianelli, B. L. Saux, P. Gamba, and S. L. Ullo, "A hybrid MLP-quantum approach in graph convolutional neural networks for Oceanic Niño Index (ONI) prediction," paper presented at the IGARSS 2024—IEEE International Geoscience and Remote Sensing Symposium (2024).
[211]A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," Nat. Comput. Sci. **1**(6), 403–409 (2021).
[212]Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, "Expressive power of parametrized quantum circuits," Phys. Rev. Res. **2**(3), 033125 (2020).
[213]Y. Yang, Z. Zhang, A. Wang, X. Xu, X. Wang, and Y. Li, "Maximizing quantum-computing expressive power through randomized circuits," Phys. Rev. Res. **6**(2), 023098 (2024).
[214]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), 4812 (2018).
[215]A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, "Effect of barren plateaus on gradient-free optimization," Quantum **5**, 558 (2021).
[216]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), 6961 (2021).
[217]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), 1791 (2021).
[218]E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, "An initialization strategy for addressing barren plateaus in parametrized quantum circuits," Quantum **3**, 214 (2019).
[219]K. Zhang, L. Liu, M.-H. Hsieh, and D. Tao, "Escaping from the barren plateau via Gaussian initializations in deep variational quantum circuits," in *Proceedings of the 36th International Conference on Neural Information Processing Systems* (Curran Associates Inc., New Orleans, LA, USA, 2022), p. 1352.