IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 October 2025, accepted 28 November 2025, date of publication 4 December 2025,
date of current version 10 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3640210

RESEARCH ARTICLE

EQDSC: Embedded Quantum Density-Based Spatial Clustering for Community Detection

SHUO HE[1,2], BOXUAN AI[2], PENGFEI GAO[2], HONGBAO LIU[2], TAO TANG[2], AND JIE WU[1]
[1]Fudan University, Shanghai 200433, China
[2]China Unionpay, Shanghai, China

Corresponding author: Jie Wu (jwu@fdu.edu.cn)

ABSTRACT Identifying community structures in complex networks is foundational to graph theory, but classical algorithms struggle with scalability and the ''curse of dimensionality'' in high-dimensional embedding, while existing quantum methods lack unified architectures or compatibility with practical hardware. Here we present Embedded Quantum Density-based Spatial Clustering (EQDSC), an end-to-end quantum framework for community detection with key theoretical and technical advances over state-of-the-art approaches: it integrates discrete-time quantum walks (DTQW) augmented with Grover diffusion operators for efficient topological feature extraction, and adopts quantum fidelity to quantify similarity—mitigating dimensionality issues inherently. We validated EQDSC on 11 real-world and synthetic datasets via Qiskit simulations. It achieved up to 0.855 modularity and 0.887 NMI (DBLP), outperforming classical methods by 18–35% on large networks. The achieved gains stem from quantum superposition and accelerated quantum walk diffusion, with small-dataset performance restricted by constrained parameter optimization. Simulated performance confirms practical applicability, laying the groundwork for future deployment on real NISQ devices.

INDEX TERMS Community detection, quantum computing, quantum walk, quantum machine learning, quantum clustering, graph embedding.

I. INTRODUCTION

The field of network science has made substantial contributions to our understanding of complex systems by elucidating the intricate structures and dynamics that underpin their behaviors. One of the core challenges within this discipline is community detection, a process aimed at identifying cohesive subgroups or communities within networks. These communities serve as functional modules and reveal hidden patterns critical for comprehending both the organization and functionality of the network. Consequently, community detection is recognized as an essential tool with a wide array of applications across various fields.

The identification of communities within social networks is essential for comprehending user activities, examining the dissemination of information, and investigating social relationships. By recognizing clusters of users with common interests, preferences, or behaviors, one can devise personalized recommendations, execute targeted marketing strategies, and analyze social influence [1]. In the context of biological networks, community detection facilitates the discovery of protein complexes and functional pathways, thereby advancing our knowledge of cellular processes [2]. The deployment of community detection methodologies within the financial technology sector stands as an imperative yet commonly undervalued tool [3], particularly for the meticulous identification of anomalous transaction patterns and the subsequent exposure of complex networks associated with credit card fraud, cash-out schemes, and illicit financial flows, thereby facilitating robust risk management and fraud detection frameworks.

A variety of proficient community detection algorithms have recently been developed to unravel the complex connectivity structures present in networks [4], [5], [6], [7], [8], [9]. Within this domain, graph embedding is recognized as a fundamental technological method, particularly when integrated with clustering techniques that rely on Euclidean distance [10], [11]. Graph embedding

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Kashif Bashir.

involves the transformation of graph-structured data into a low-dimensional vector space, wherein nodes are represented as vectors that encapsulate their relational and attributive properties [12]. This process enables the application of clustering algorithms to discern communities based on the geometric proximity of nodes within the embedded space. Among the myriad of graph embedding methodologies, those based on random walks have garnered significant attention due to their ability to capture the graph's topology and node attributes effectively [13], [14]. However, traditional community detection algorithms, while efficacious in certain contexts, encounter substantial limitations in large-scale networks [15]. Notably, their accuracy dwindles as networks swell in size and complexity, rendering it increasingly arduous to precisely identify communities. Therefore, the computational demands of these algorithms escalate with network scale, leading to protracted processing times that are impractical for real-time applications.

The advent of quantum computing marks a seminal milestone in the realm of computational science, ushering in a new era of possibilities for solving complex problems. Quantum computing harnesses the principles of quantum mechanics, such as superposition and entanglement, to execute computations that are infeasible for classical computers [16]. The advantage of quantum computing has introduced a novel paradigm for graph embedding, which utilizes the concept of quantum walks to provide a more sophisticated and efficient means of exploring the graph's structure compared to classical random walks [17], [18]. The application of quantum walk-based graph embedding algorithms has extended to various graph-related tasks, with notable successes in network alignment [19], link prediction [20] and node proximity estimation [21].

In anticipation of the imminent realization of quantum devices' full potential, we endeavor to establish a comprehensive quantum network embedding learning paradigm. While classical methods have advanced large-scale community detection, their reliance on high-dimensional embeddings and Euclidean metrics limits performance on ultra-large or high-complexity networks. Existing quantum methods, though promising, fail to provide a unified, practical solution tailored for community detection—either lacking optimization for embedding-clustering integration or being restricted to specialized hardware/ tasks. To fill this gap, we propose Embedded Quantum Density-based Spatial Clustering (EQDSC), an end-to-end quantum framework that integrates discrete-time quantum walks (DTQW) with Grover operators for graph embedding and quantum fidelity-based density clustering, specifically designed for community detection on noisy intermediate-scale quantum (NISQ) computers. We summarize our contributions as follows:

(1) Grover-Enhanced Quantum Embedding for Community Detection: EQDSC introduces a DTQW-based embedding mechanism augmented with Grover diffusion operators to project network topology into high-dimensional Hilbert space. Unlike prior quantum walk methods, the Grover operator accelerates state diffusion, enabling more efficient extraction of fine-grained topological features. This embedding uses exponentially fewer qubits compared to classical embedding dimensions, addressing the scalability limitations of classical graph embedding.

(2) End-to-End Fully Quantum Architecture: Unlike hybrid quantum-classical methods or disjoint quantum modules with separate quantum walk and classical clustering, EQDSC seamlessly integrates quantum embedding and quantum density clustering into a single quantum pipeline. By eliminating classical preprocessing/postprocessing, it reduces hybrid system overhead and leverages intrinsic quantum parallelism— a unique advantage, as few fully quantum approaches exist for network applications with minimal qubit requirements.

(3) NISQ-Compatible Design for Practical Deployment: EQDSC is tailored for immediate implementation on existing programmable quantum hardware. Building on demonstrated DTQW feasibility in superconducting [22] and photonic [23] quantum computers, we design custom quantum gate/circuit architectures that align with NISQ device constraints. This distinguishes it from quantum annealing methods [24], which are limited to optimization tasks and incompatible with universal quantum circuits.

(4) Superior Performance on Large-Scale Real-World Datasets: We validate EQDSC on 11 datasets, achieving up to 0.855 modularity and 0.887 normalized mutual information (DBLP)—outperforming classical SOTA (e.g., Louvain, DeepWalk) by 18–35% on large networks. Unlike classical methods, EQDSC maintains stability on ultra-large datasets, confirming its robustness for practical large-scale applications.

Difference to existing works: While quantum mechanics has been increasingly integrated into community detection algorithms [25], existing approaches suffer from critical limitations in methodological unity, scalability, and alignment with practical quantum computing architectures. Our proposed EQDSC algorithm addresses these gaps through fundamental methodological innovations, distinguishing itself from three major categories of state-of-the-art quantum-related methods as follows:

(1) Comparison to Other Quantum Walk Approaches [26], [27], [28], [29]: Quantum walks have shown promise in capturing network topology, but existing community detection methods leveraging quantum walks suffer from two core drawbacks: they either treat quantum walks as auxiliary tools for similarity measurement or rely on heuristic parameter tuning, and none have systematically exploited quantum walks as a dedicated graph embedding mechanism. Most notably, prior methods are validated only on small-scale networks and fail to scale to large-scale real-world datasets. In contrast, EQDSC pioneers the integration of discrete-time quantum walks into an end-to-end quantum embedding pipeline: quantum walks transform graph data into high-dimensional Hilbert space, preserving topological structures via quantum superposition, and the resulting quantum states directly

serve as inputs for quantum clustering. This design not only eliminates classical pre-processing but also leverages the exponential state space expansion of quantum systems to capture fine-grained community structures, addressing the scalability bottleneck of existing quantum walk-based methods and ensuring stability across diverse network sizes.

(2) Comparison to Quantum Annealing Methods [24]: Quantum annealing approaches for community detection [24] are constrained to optimization-centric frameworks, limiting their applicability to networks with irregular topologies or non-uniform community structures. In contrast, EQDSC adopts a programmable quantum circuit architecture, which confers three pivotal advantages tailored to community detection: (i) *Universality*: Quantum circuits support arbitrary unitary transformations, enabling EQDSC to adapt to diverse network types without redefining optimization objectives. (ii) *Scalability*: The processing power of quantum circuits scales exponentially with the number of qubits, allowing EQDSC to handle ultra-large-scale networks — a capability unattainable by quantum annealers, whose scalability is limited by physical hardware constraints. (iii) *Maturity*: Quantum circuit-based architectures have advanced significantly in NISQ-era hardware support, enabling EQDSC to be deployed on existing quantum devices, whereas quantum annealing remains confined to specialized hardware with narrow application scopes.

(3) Comparison with Quantum-Inspired Techniques [31]: Quantum-inspired algorithms augment classical learning frameworks with quantum-inspired heuristics but do not leverage genuine quantum mechanical phenomena. As such, they remain bound to classical computational limits: they cannot exploit exponential state space representation, nor can they mitigate the ''curse of dimensionality'' in high-dimensional graph embedding. In stark contrast, EQDSC is a fully quantum method: it encodes graph nodes into quantum states, uses quantum fidelity for similarity quantification, and implements density clustering entirely within the quantum domain. This architecture enables EQDSC to effectively handle high-dimensional embedding spaces while utilizing a minimal number of qubits and leveraging advancements in quantum error correction. These benefits are not achievable by quantum-inspired approaches based on classical computational frameworks. Furthermore, EQDSC's compatibility with NISQ devices ensures immediate practical relevance, whereas quantum-inspired techniques offer no path to leverage future quantum hardware advancements.

The organization of this paper is structured as follows: Section II provides an introduction to the essential concepts of quantum computing. Section III offers an extensive review and analysis of current community detection techniques that utilize quantum walks. In Section IV, we detail the conceptual framework and methods underlying our proposed algorithm. Section V showcases the effectiveness of our approach via simulation experiments, emphasizing its benefits in comparison to conventional methods. The paper concludes with Section VI, where we discuss potential future research avenues.

II. PRELIMINARIES OF QUANTUM COMPUTING

Quantum computing marks a revolutionary shift in the computing landscape by harnessing the fundamental principles of quantum mechanics to execute complex calculations. Unlike classical computing, which relies on binary digits or bits that are strictly in a state of 0 or 1, quantum computing introduces the concept of qubits. These qubits have the remarkable ability to exist in multiple states simultaneously, known as superposition. This unique property allows quantum computers to process a vast amount of information in parallel, offering the potential for exponential speed-up in solving certain types of computational challenges. In this section, we present an introductory overview of quantum computing and quantum machine learning, specifically catering to readers possessing a foundation in linear algebra yet lacking familiarity with quantum paradigms. Our objective is to facilitate a comprehensive understanding of the quantum technologies utilized within the context of this paper.

A. QUNTUM COMPUTING BASICS

A qubit serves as the elementary unit of quantum information. In contrast to classical bits, qubits possess the extraordinary capability to exist in a superposition of two states, typically designated as $|0\rangle$ and $|1\rangle$. Dirac notation $|\rangle$ is a unique representation of state vectors in quantum mechanics. Mathematically, the base state vectors are represented as:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{1}$$

In the column vectors, the elements 0 and 1 denote the square roots of the probabilities associated with the qubit being in different states. The state of a qubit can be depicted by a vector within a two-dimensional complex vector space, referred to as the Hilbert space. The overall state of a qubit is articulated as a linear combination of the basis states:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2}$$

where $|\psi\rangle$ is the quantum state, α and β are complex coefficients, and $|0\rangle$ and $|1\rangle$ are the orthonormal basis states. The complex coefficients satisfy the normalization condition $|\alpha|^2+|\beta|^2 = 1$, ensuring that the state is properly normalized. The ability of a qubit to be in a superposition of states is a cornerstone of quantum computing's computational power. When a quantum system is measured, it undergoes a collapse into one of the basis states. The probability of measuring a qubit in the state $|0\rangle$ is $|\alpha|^2$, and the probability of measuring it in the state $|1\rangle$ is $|\beta|^2$. The superposition of qubits is a fundamental phenomenon in quantum mechanics, which is consistent with the Born rule for probability amplitudes and the Copenhagen interpretation's view of quantum states. Therefore, Equation (2) can be also reformulated as follows:

$$|\psi\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{3}$$

Given that a single qubit has the potential to represent two states at the same time, a system comprising n qubits can collectively embody 2^n unique states concurrently. These states are characterized by their respective probability amplitudes, which are conveniently represented as a 2^n-dimensional column vector. An n-qubit system represents the most general form of quantum computation and information processing. Mathematically, the state space of an n-qubit system is a 2^n-dimensional Hilbert space, denoted as $\mathcal{H}^{\otimes n} = \mathcal{H} \otimes \mathcal{H} \otimes \cdots \otimes \mathcal{H}$, where $\mathcal{H}$ is the two-dimensional Hilbert space of a single qubit. A general state $|\psi\rangle$ of an n-qubit system can be expressed as a linear combination of the computational basis states:

$$|\psi\rangle = \sum_{s_1,s_2,\ldots,s_n \in \{0,1\}} \alpha_{s_1 s_2 \ldots s_n} |s_1 s_2 \ldots s_n\rangle \tag{4}$$

where $\sum_{s_1,s_2,\ldots,s_n \in \{0,1\}} |\alpha_{s_1 s_2 \ldots s_n}|^2 = 1$. Here, $|s_1 s_2 \ldots s_n\rangle = |s_1\rangle \otimes |s_2\rangle \otimes \cdots \otimes |s_n\rangle$ represents the tensor product of the individual qubit states, and $\alpha_{s_1 s_2 \ldots s_n}$ are complex amplitudes satisfying the normalization condition. This representation encapsulates the superposition principle, allowing qubits to simultaneously explore an exponential number of computational paths.

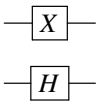
B. QUANTUM OPERATIONS

Quantum computing is fundamentally reliant on quantum gates, which serve a role similar to classical logical gates, albeit operating on qubits. These gates alter the states of qubits by implementing unitary transformations in a complex vector space, thus preserving norms due to their intrinsic reversibility, a feature characterized by unitary matrices. These transformations are crucial to quantum theory as they represent the temporal evolution of quantum systems, providing quantum computing with a unique benefit compared to irreversible classical computations, namely, enhanced error correction capabilities.

Quantum circuits, mirroring classical counterparts, provide a visual framework for designing and analyzing quantum algorithms. Two widely utilized unitary transformations are the X and H gate. The X gate, akin to the classical NOT gate, flips the state of a qubit, transforming $|0\rangle$ to $|1\rangle$ and vice versa. It plays a crucial role in qubit manipulation and state inversion. The H gate, also known as the Hadamard gate, creates an equal superposition of the basis states when applied to a qubit in a computational basis state, mapping $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. This gate is essential for creating quantum superpositions, enabling quantum interference, and is a fundamental component in various quantum algorithms. The mathematical representations of X and H are as follows:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5}$$

Quantum circuit diagrams serve as a powerful visual tool for representing quantum operations, facilitating the design and analysis of quantum algorithms. They comprise qubits depicted as wires and quantum gates as operational boxes, with time progression from left to right. In quantum circuits, X and H are represented as shown below:



Utilizing the X and H gates, we can construct the swap-test circuit [40], a powerful tool for measuring the similarity between two quantum states.

C. QUANTUM CIRCUIT

Quantum circuits constitute the core architectural framework for executing quantum algorithms, encompassing a register of qubits and a sequence of unitary operations that manipulate the quantum state of these qubits. Formally, a quantum circuit is defined as a tuple $(\mathcal{Q}, \mathcal{U}_{\text{seq}})$, where $\mathcal{Q} = \{q_1, q_2, \ldots, q_n\}$ denotes a set of n qubits-typically initialized in the computational basis state $|\psi_0\rangle = |0\rangle^{\otimes n} = |0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle$ and $\mathcal{U}_{\text{seq}} = \{U_1, U_2, \ldots, U_T\}$ represents a finite sequence of unitary quantum gates. Each gate $U_t \in \mathcal{U}_{\text{seq}}$ acts on a subset of qubits (ranging from 1 to n) and is characterized by a unitary matrix $U_t \in \mathbb{C}^{n \times n}$, satisfying the unitarity condition $U_t^\dagger U_t = U_t U_t^\dagger = I$; here, $U_t^\dagger$ denotes the Hermitian conjugate of U_t, and I is the identity matrix of appropriate dimension.

Multi-qubit gates extend manipulation to entangled states-a unique quantum phenomenon where qubits exhibit correlated behavior regardless of distance-with the controlled-NOT (CNOT) gate being the most fundamental. The CNOT gate acts on two qubits: a control qubit q_c and a target qubit q_t. It flips the target qubit's state if and only if the control qubit is in $|1\rangle$; for control qubit q_1 and target qubit q_2, its matrix representation in the two-qubit basis is:

$$U_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{6}$$

A critical property of quantum circuits is their universality: any unitary transformation on n qubits can be approximated to arbitrary precision using a finite sequence of single-qubit gates and CNOT gates. The approximation error decreases exponentially with the number of gates, enabling the implementation of complex quantum operations via elementary building blocks. For a circuit with T gates, the overall unitary evolution of the quantum state is the product of the individual gate unitaries (applied in the order of the sequence), i.e., $U_{\text{total}} = U_T U_{T-1} \cdots U_1$. This transformation maps the initial state $|\psi_0\rangle$ to the final state $|\psi_T\rangle = U_{\text{total}} |\psi_0\rangle$, which encodes the result of the quantum computation before measurement.

III. RELATED WORKS

In the domain of community detection, quantum walks have emerged as an innovative and promising paradigm, offering

TABLE 1. Notations and definitions.

Notation	Definition
G	The graph
V	The vertex set of G
E	The edge set of G
N	The number of vertices in G
A	The adjacency matrix of G
n	The number of qubits
$\|\varphi_i\rangle$	The quantum state of vertex v_i
$\|\varphi_i^t\rangle$	The quantum state of vertex v_i at time step t
$\|\Phi^t\rangle$	The quantum state of G
U_{QW}	The unitary transformation of quantum walk
F	The fidelity of two quantum states
r	The radius of quantum clustering
ρ	The density of quantum clustering

unique insights into the intricate structures of networks. Mukai [26] adopt the Fourier coin-driven discrete-time quantum walk algorithm, exploiting the asymptotic eigenvalue distribution of the evolution operator along the unit circle in the complex plane to delineate community boundaries within networks. Nonetheless, the algorithm's reliance on averaging over an extensive number of quantum walks poses a significant computational challenge for large-scale network analysis. Faccin [27] et al. introduced a framework utilizing continuous-time quantum walks for community detection. Their method optimizes temporal dynamics to enhance probability fluctuations at intracommunity nodes while suppressing them at extracommunity nodes. Despite its promise, the determination of an optimal time parameter remains a non-trivial challenge, introducing instability into the resulting community structures. Song [28] proposed a quantum walk algorithm for detecting simplicial communities, a higher-order community structure within networks, using a Fourier coin to create entangled states among adjacent simplices. Roy and Chakrabarti [29] introduced a graph clustering method employing Discrete-Time Quantum Random Walk on graphs, wherein the walker's transitions, governed by a shift operator contingent on the coin operator's results, iteratively generate new superposition states via the unitary operator, ultimately converging to a stable node visit probability distribution. Cluster assignment is then determined by the proximity of nodes in terms of their visit probabilities, grouping those with similar values into the same cluster. Umeano et al. [30] proposed the deteQt quantum protocol for community and botnet detection, using modularity matrix/graph Laplacian ground states, encoding community info in qubits, signing via quantum signal processing to mimic hypergraph states, and deterministic elimination to reduce sample scaling from exponential to polynomial. However, a notable limitation of both aforementioned methodologies is that, while their effectiveness has been demonstrated on only a single, relatively small network (the Zachary Karate Club network), the scalability and generalizability of these methods to larger and more complex networks still await rigorous validation and optimization.

IV. METHODOLOGY

The EQDSC algorithm formalizes the input network as a graph, then projects its topology into high-dimensional Hilbert space via discrete-time quantum walks with Grover operators for quantum state embedding. It employs quantum density clustering, using quantum fidelity to quantify state similarity, thereby partitioning quantum-state nodes into communities, forming an end-to-end quantum framework. Figure 1 schematically illustrates the holistic workflow of EQDSC, encompassing graph construction, discrete-time quantum walk embedding, and quantum density clustering. Table 1 summarizes the key notations and definitions used throughout the methodology of EQDSC, including graph-related symbols, quantum state representations, and clustering parameters.

A. GRAPH CONSTRUCTION

Let $G = (V, E)$ denote a graph, wherein $V = \{v_1, \ldots, v_N\}$ constitutes the set of vertices with cardinality $N = |V|$, and E represents the set of edges. We define the adjacency matrix $A \in \mathbb{R}^{N\times N}$ associated with graph G such that the entry $A_{i,j}$, positioned at the i-th row and j-th column, encodes the existence of an edge connecting vertex v_i to vertex v_j:

$$A_{i,j} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{if } (v_i, v_j) \notin E \end{cases} \tag{7}$$

The degree of a node, quantifying the number of edges incident to it in the graph, is defined for node v_j as $d_j = \sum_{i=1}^{N} A_{i,j}$, which corresponds to the sum of all elements in the j-th column of matrix A.

The Markov process is a stochastic process endowed with the memoryless property, implying that the system's next state is contingent solely upon the current state, independent of preceding states. This attribute permits a concise representation of the graph structure through state transitions, as each node considers only those directly connected to it, disregarding others. The probability transition matrix, denoted $P \in \mathbb{C}^{N\times N}$ for graph G, characterizes the transition probabilities in a Markov process and is represented as:

$$P_{i,j} = \frac{A_{i,j}}{d_j} \tag{8}$$

where $P_{i,j}$ signifies the probability of transitioning from node v_i to node v_j, mathematically expressed as $P_{i,j} = P(X_t = s_j | X_{t-1} = s_i)$, with the row sum of the probability transition matrix being unity: $\sum_{i=1}^{N} P_{i,j} = 1$.

Quantum state encoding refers to the procedure of embedding classical data into quantum states. Any node v_i within the transaction graph G can be represented by a quantum state $|i\rangle$, an $N \times 1$ dimensional column vector with a 1 in the i-th position and 0 elsewhere, adhering to the one-hot encoding

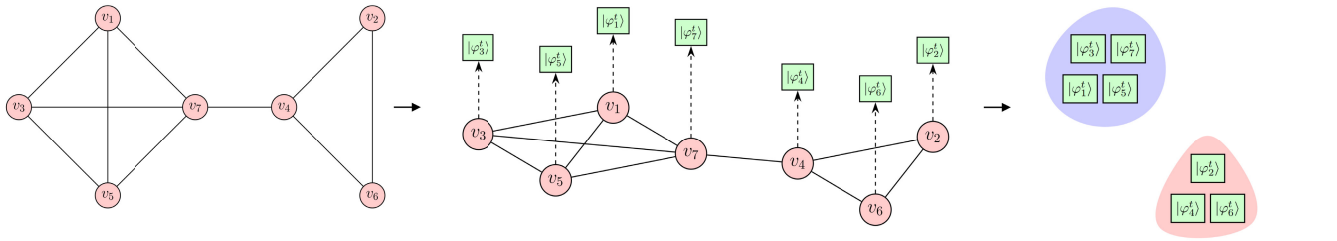


FIGURE 1. Overview of EQDSC. (a) Establish black edges connecting red nodes, There are two distinct communities: (1,3,4,5) and (2,4,6). (b) Apply a discrete-time quantum walk on the network for embedding purposes, where dashed arrows depict quantum transitions. Green squares signify the quantum state of each vertex after this embedding. (c) Utilize quantum density clustering to categorize quantum vertices into respective clusters.

principle:

$$|1\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, |2\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots, |N\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{9}$$

Given a graph G with N nodes, a minimum of $n \geq \log_2 N$ qubits are requisite to represent the quantum state $|i\rangle$ of an individual node v_i.

B. QUANTUM WALK

The notion of random walk, which describes the erratic and unpredictable movement of an entity within a constrained trajectory or spatial domain, has found widespread application in the realm of graph structures. In this context, a random walk entails the stochastic navigation of a particle across the nodes of a graph, guided solely by the edges that delineate the connectivity of the network. This process is particularly adept at capturing the subtle topological intricacies inherent in complex graphs, rendering it an indispensable tool for a myriad of graph analysis tasks, including but not limited to graph embedding [13], [14] and node ranking [33].

The advent of quantum walk, as introduced by Aharonov [34], represents a paradigm shift from the classical random walk framework. Leveraging the principle of quantum superposition, the particle in a quantum walk is endowed with the extraordinary capability to exist in a state of "bifurcation," wherein it probabilistically propagates along all feasible pathways simultaneously. This departure from deterministic traversal enables the concurrent exploration of multiple, divergent trajectories. The probability amplitudes of the wave function encapsulate the likelihood of the particle's presence at various nodes, evolving dynamically with each step of the walk. This unique attribute facilitates the translation of intricate graph data into quantum states, while preserving the underlying topological characteristics of the network. Notably, quantum walk exhibits an exponentially accelerated diffusion rate compared to its classical counterpart, achieving $O(\sqrt{N})$ steps to reach node N, as opposed to the $O(N)$ steps required by classical random walk [32].

Quantum walk can be delineated into two primary categories: discrete-time and continuous-time quantum walks. Within the discrete-time domain, further distinctions are made between coin quantum walk and scattering quantum walk. Coin quantum walk primarily unfolds at the nodes of the graph, necessitating a coin space with a dimensionality commensurate with the degree of the node. This makes it particularly well-suited for regular graph structures, where node connectivity is uniform [35]. Conversely, scattering quantum walk encodes all graph edges as quantum basis states, eliminating the need for a coin operator and rendering it more versatile in the context of irregular graphs [36].

In the formulation of quantum algorithms, individuals are typically represented as independent standard basis states within a Hilbert space. The problem is then transformed into a specific quantum state through the superposition of these basis states. Within the framework of scattering quantum walk, the standard basis state $|i, j\rangle$ is defined for graph G, with the corresponding Hilbert space $\mathcal{H}$ spanned by the set of all edges in the graph. For two nodes, v_i and v_j, interconnected by an edge in graph G, the quantum basis state $|i, j\rangle$ signifies the current position of the walker at node v_i, with the subsequent intended destination being v_j. Conversely, the state $|j, i\rangle$ represents the walker's current position at node v_j, with the intended next stop being v_i. The state $|i, j\rangle$ is constructed as the tensor product of the column vectors $|i\rangle$ and $|j\rangle$ ($|i\rangle \otimes |j\rangle = |i, j\rangle$), implying that $|i, j\rangle$ constitutes an $N \times N$ matrix with a single entry of 1 at the intersection of the i-th row and j-th column, and all other entries being 0.

Each edge in the graph corresponds to two different ground states, and for each node v_i, there exist two non-overlapping subspaces: the subspace C_{v_i} spanned by the ground states starting from v_i, and the subspace Ω_{v_i} spanned by the ground states ending at v_i. Quantum walks can be seen as mappings from C_{v_i} to Ω_{v_i}. Firstly, the graph G is transformed into the quantum state $|\Phi\rangle$ by superposing the standard basis states $|i, j\rangle$:

$$|\Phi\rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} \phi_{i,j} |i, j\rangle \tag{10}$$

where $\Phi \in \mathbb{C}^{N\times N}$, and the element in the i-th row and j-th column is $\phi_{i,j} = \left(\frac{P_{i,j}}{N}\right)^{1/2}$, satisfying the normalization requirement of quantum superposition. This represents the superposition of all possible paths starting from v_i:

$$|\phi_i\rangle = \sum_{j=1}^{N} \sqrt{P_{i,j}}\,|j\rangle \tag{11}$$

Mathematically, $|\phi_i\rangle$ can be regarded as the transpose of the square root of the i-th row of the transition matrix P. Specifically, $|\phi_i\rangle = \left[\sqrt{P_{i,1}}\ \sqrt{P_{i,2}}\ \cdots\ \sqrt{P_{i,N}}\right]^T$. By taking the tensor product of the state $|i\rangle$ and the corresponding $|\phi_i\rangle$, we obtain the projected state $|\psi_i\rangle$ of the Markov chain:

$$|\psi_i\rangle = |i\rangle \otimes \left(\sum_{j=1}^{N} \sqrt{P_{i,j}}\,|j\rangle\right) = \sum_{j=1}^{N} \sqrt{P_{i,j}}\,|i,j\rangle \tag{12}$$

In the context of quantum mechanics, a collection of orthogonal and normalized vectors denoted as $\{|\psi_i\rangle : i = 1,\ldots,N\}$ has the capacity to span an N-dimensional Hilbert space labeled Ψ. The unitary operator Π serves the purpose of projecting any vector residing in the Hilbert space H onto the space Ψ:

$$\Pi = \sum_{i=1}^{N} |\psi_i\rangle\,\langle\psi_i| \tag{13}$$

Central to the Grover quantum search algorithm is the Grover diffusion operator. This operator amplifies the quantum state by employing a reflection transformation, which is analogous to a qubit rotation operation. Utilizing the projection operator Π, it is possible to formulate the Grover operator $\mathcal{C}$ as follows:

$$\mathcal{C} = 2\Pi - I \tag{14}$$

Among all operators that satisfy unitarity and permutation symmetry, the Grover operator is the farthest from the identity operator, which intuitively can speed up the diffusion process [37]. The swap operator S can be used to exchange the memory of two quantum registers, such that $S\,|i,j\rangle = |j,i\rangle$:

$$S = \sum_{i=1}^{N}\sum_{j=1}^{N} |i,j\rangle\,\langle j,i| \tag{15}$$

The mathematical representation of the quantum walk process is the unitary evolution operator $U_{QW} \in \mathbb{C}^{N\times N}$. Applying U_{QW} to the initial state $|\Phi\rangle$ achieves the unitary evolution of the walk state. Combining equations (13) and (14), the expression for U_{QW} is constructed as:

$$U_{QW} = S\mathcal{C} = S(2\Pi - I) \tag{16}$$

After U_{QW} applys on each node, the incoming edge state is mapped to the outgoing edge state. The state obtained after one step of quantum walk evolution from $|i,j\rangle$ is:

$$|i,j\rangle \xrightarrow{U_{QW}} \left(\frac{2}{d_j} - 1\right)|j,i\rangle + \frac{2}{d_j}\sum_{k\neq i,(j,k)\in E} |j,k\rangle \tag{17}$$

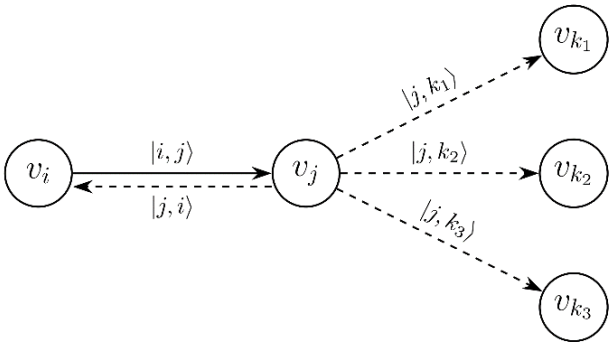


FIGURE 2. Discrete-time Quantum Walk. The solid arrows represent the current walk state, while the dashed arrows indicate potential states for the next walk step.

Equation (16) represents that after one step of quantum walk (Figure.2), the walker has moved from v_i to v_j. In the next step, starting from v_j, there is a probability of $\left(\frac{2}{d_j} - 1\right)^2$ to return to v_i, and a probability of $\left(\frac{2}{d_j}\right)^2$ to reach another node v_k that is adjacent to v_j but different from v_i.

In a quantum computer, quantum walk does not require a real physical particle to move between nodes. Instead, it simulates the probability changes of a walker at different nodes by altering the states of qubits through quantum circuits. As the fundamental operation unit in quantum computation, quantum gates function similarly to logic gates in classical computation, both operating on basic units of information to change their states. Quantum walk utilizes quantum gates to implement unitary evolution on n qubits, thereby adjusting the probabilities of the qubit superposition states and simulating the dynamic effect of the walker moving in a graph. The quantum walk at step t is a unitary evolution operation applied to the current state $|\Phi^{t-1}\rangle$, expressed as follows:

$$U_{QW}\,|\Phi^{t-1}\rangle = |\Phi^{t}\rangle \tag{18}$$

When $t = 0$, $|\Phi^0\rangle$ represents the initial quantum state of graph G, i.e., before the quantum walk begins. The preparation of the initial quantum state involves using quantum gates $U(\theta)$ to convert multiple low-energy state qubits $|0\rangle$ into a uniform superposition state as $|\varphi_i^0\rangle$, as shown in formula (2). The U_{QW} quantum gate directly acts on the node $|\varphi_i^{t-1}\rangle$ as follows:

$$U_{QW}\,|\varphi_i^{t-1}\rangle = |\varphi_i^{t}\rangle \tag{19}$$

The process of performing t quantum walks on $|\varphi_i^0\rangle$ is as follows:

$$|\varphi_i^0\rangle \xrightarrow{U_{QW}} |\varphi_i^1\rangle \xrightarrow{U_{QW}} \cdots |\varphi_i^{t-1}\rangle \xrightarrow{U_{QW}} |\varphi_i^t\rangle \tag{20}$$

By repeatedly performing unitary evolution, the probability transition matrix P also changes with each unitary evolution ($P^0 \rightarrow P^t$), allowing the quantum computer to simulate the process of a walking particle moving in a graph. The quantum

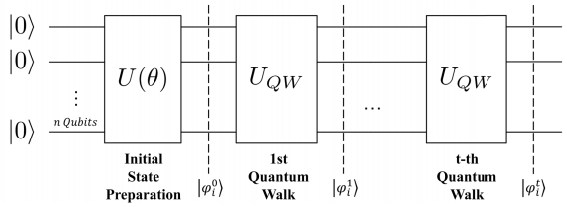


FIGURE 3. The n-qubit quantum circuit for t-step quantum walk from $|\varphi_1^0\rangle$ to $|\varphi_1^t\rangle$. The repeated U_{QW} blocks represent multiple steps of the quantum walk.

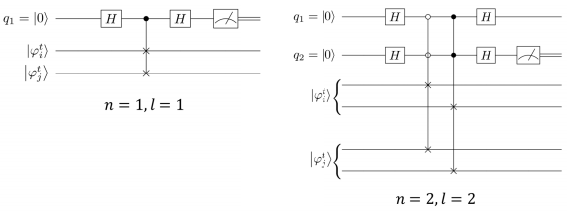


FIGURE 4. Quantum circuit for Swap-test with single ($n = 1$, left) and double qubits ($n = 2$, right).

circuit outputs the quantum states of all nodes sequentially, with the quantum states corresponding to nodes $v_1, \ldots, v_N$ represented as $|\varphi_1^t\rangle, |\varphi_2^t\rangle, \ldots, |\varphi_N^t\rangle$.

C. QUANTUM FIDELITY

Aimeur proposed using fidelity F to measure the similarity between two independent quantum states [39]. For example, the similarity between v_i and v_j can be measured by the fidelity $F(v_i, v_j)$ of their quantum states, mathematically represented as:

$$F(v_i, v_j) = |\langle\varphi_i^t|\varphi_j^t\rangle|^2 \tag{21}$$

The inner product $\langle\varphi_i^t|\varphi_j^t\rangle$ of two quantum states refers to their scalar product in Hilbert space, calculated as the sum of the complex conjugate products of the two state vectors. The absolute value of the inner product represents the modulus of the cosine of the angle between the two pure states $|\varphi_i^t\rangle$ and $|\varphi_j^t\rangle$ in the complex vector space. $|\langle\varphi_i^t|\varphi_j^t\rangle|^2$ represents the overlap probability of the two states, i.e., the intensity of the projection of one state onto the other. For $|\varphi_i^t\rangle$ and $|\varphi_j^t\rangle$ represented by n-bit qubits, the fidelity ranges from $0 \leq F(v_i, v_j) \leq n$, with a larger F value indicating greater similarity between the two quantum states. If the two states are identical, the fidelity is n.

The Swap-test quantum circuit serves as an elegant mechanism for determining the squared modulus of the inner product between two arbitrary quantum states, as depicted in Fig.4 [40]. This circuit incorporates a Hadamard gate, designated by H, and operates on inputs $|\varphi_i^t\rangle$ and $|\varphi_j^t\rangle$, supplemented by $l \in \mathbb{Z}^+$ ancilla qubits, with the constraint $l \geq 1 + \log_2 n$.

In the realm of quantum computing, direct inspection of qubits by classical means is infeasible; rather, qubits are subjected to the process of quantum measurement. Quantum states manifest as superpositions of basis states, each endowed with a specific probability amplitude. Consequently, the outcome of any single measurement is inherently probabilistic, necessitating repeated measurements to accurately estimate the probabilities associated with each basis state.

At the circuit's output, multiple measurements are exclusively performed on a single ancilla qubit, denoted q_l. The probability that q_l is observed in the $|0\rangle$ state is given by the following expression:

$$P(q_l = |0\rangle) = \frac{1}{2} + \frac{1}{2n}\left|\langle\varphi_i^t \mid \varphi_j^t\rangle\right|^2 \tag{22}$$

From this probability, we can derive the fidelity between the two quantum states, a measure of their similarity, using the formula:

$$F(v_i, v_j) = 2n \cdot P(q_l = |0\rangle) - n \tag{23}$$

D. QUANTUM DENSITY CLUSTERING

Exploiting quantum computation to enhance and refine artificial intelligence algorithms is a prominent area of contemporary research. Clustering, as a vital component of traditional machine learning algorithms, can divide datasets into several groups, maximizing the similarity of data points within the same group while minimizing the similarity between data points in different groups. This method is widely applied in various fields such as data analysis, data mining, and image processing [38]. However, as the dimensionality of data (i.e., the number of features) increases, data points become more sparsely distributed in high-dimensional space, and the distance between any two points tends to be equal. The traditional Euclidean distance metric ceases to be an effective means of distinguishing data points, leading to the potential failure of machine learning algorithms that rely on distance metrics, such as clustering. This phenomenon is known as the "curse of dimensionality" [41].

Quantum similarity is not deduced from numerical manipulations of vectors; instead, it is extracted through the process of measuring qubits. This act of measurement is fundamentally a physical procedure, distinct from mathematical computation, thereby dispelling the concept of computation in this context. Regardless of the dimensionality of the vectors that mathematically represent the two quantum states, the similarity between them can be determined through repeated measurements of an auxiliary bit within the Swap-test circuit. Therefore, the adoption of quantum fidelity in lieu of the conventional Euclidean distance measure eliminates the necessity for complex mathematical computations and mitigates the "curse of dimensionality" often encountered in high-dimensional vector analysis.

The EQDSC introduces an innovative density clustering algorithm that leverages quantum fidelity to group similar quantum nodes into the same "state cluster". The logical framework of this algorithm bears resemblance to traditional density clustering algorithms [42], yet it diverges in its

approach to measuring the distance between data points. Specifically, it replaces the conventional Euclidean distance with the 'quantum distance' in Hilbert space. A notable advantage of this algorithm is its ability to determine the clusters without the need for presetting the number of state clusters. Instead, it ascertains the clusters through a rigorous analysis of the local density of data points that represent quantum state nodes. Before density clustering begins, two parameters need to be set: radius r and density threshold ρ. $r \in (0, 1)$ is the ratio of the fidelity and the number of qubits, defining the range of a point's neighborhood; ρ is the minimum number of data points required in a neighborhood to be considered as a dense region. The neighborhood $N(v_j)$ of node v_j refers to the set of all points with a distance from v_j not exceeding r, mathematically represented as:

$$N(v_j) = \{v_i \in V | F(v_i, v_j)/n \leq r\} \tag{24}$$

Before the algorithm starts, each data point is marked as unvisited. The algorithm sequentially selects an unvisited data point from the dataset and identifies all nearby points within a distance r. If a point has ρ or more neighbors within this range, indicating a dense area around it, the point is marked as a "core point"; if the total number of nearby points within distance r from the visited point is less than ρ, it is marked as a "noise point". Two points are considered density-reachable if one can reach the other through a series of intermediate points, each of which is a core point. After defining these concepts, the following steps are executed:

1) Starting from any unvisited core point, find all data points that are density-reachable from this core point to form a state cluster;
2) Within the current state cluster, starting from each core point, find all points that are density-reachable from it and add them to the current cluster until no new core points can be added;
3) When all core points within the state cluster have been visited, find another unvisited core point in the dataset and repeat steps (1) and (2) to form a new state cluster. This process continues until all core points have been visited, i.e., all state clusters have been formed, and the algorithm ends.

The pseudo code of the overall algorithm can be described in Figure 5.

Algorithm: Embedded Quantum Density-based Spatial Clustering (EQDSC)

Input: Graph G, radius r, density threshold ρ
Output: The community detection result of G (list of clusters)

for *each* $v_i \in G$ **do**
  Mark v_i as "unvisited"
  $|\varphi_i^0\rangle \leftarrow \sum_{j=1}^{N} \sqrt{P_{i,j}}|j\rangle$
  for $t = 1$ **to** T **do**
    $|\varphi_i^t\rangle \leftarrow U_{QW}|\varphi_i^{t-1}\rangle$
    $\boldsymbol{\varphi}.append(|\varphi_i^t\rangle)$
  end
end
Initialize *clusters*
Function `QuantumClustering`($\boldsymbol{\varphi}$, r, ρ)**:**
  Initialize $C = 0$ **for** *each quantum vertex* p *in* $\boldsymbol{\varphi}$ **do**
    if p *is "unvisited"* **then**
      $NearPts$ = `regionQuery`(p, r)
      if $sizeof(NearPts) < \rho$ **then**
        Mark p as "noise"
      end
      else
        $C \leftarrow C + 1$
        $clusters.append()$ Mark p as "visited"
        `expandCluster`(p, $NearPts$, $C - 1$, r, ρ, $clusters$)
      end
    end
  end
  return *clusters*
Function `expandCluster`(p, $NearPts$, $current\_cluster\_idx$, r, ρ, $clusters$)**:**
  Add p to $clusters[current\_cluster\_idx]$
  for *each quantum vertex* q *in* $NearPts$ **do**
    if q *is "unvisited"* **then**
      Mark q as "visited"
      $NewNearPts$ = `regionQuery`(q, r)
      if $sizeof(NewNearPts) \geq \rho$ **then**
        $NearPts = NearPts \cup NewNearPts$
      end
    end
    if q *is not in any cluster and* q *is not "noise"* **then**
      Add q to $clusters[current\_cluster\_idx]$
    end
  end
  return
return
Function `regionQuery`(p, r)**:**
  return points within distance r from p
return

FIGURE 5. The pseudo code of EQDSC.

V. EXPERIMENTAL EVALUATION

A. EXPERIMENT ENVIRONMENT

All experimental procedures were executed on IBM Qiskit v1.0 quantum framework, utilizing a high-performance personal computer, equipped with an Intel Core i9 processor, NVIDIA GeForce RTX 4090 graphics card, 64GB of DDR5 RAM, and an ultra-fast 1TB NVMe SSD.

B. DATASETS

Public real-world datasets for community detection problems provide a unified benchmarking platform for community detection algorithms. Researchers can apply their algorithms to these datasets and compare the results against known ground-truth partitions or widely accepted community structures, thereby assessing the accuracy of their algorithms. Synthetic datasets such as LFR also prove highly beneficial for assessing the performance of community detection algorithms. In this paper, we have selected 11 widely-used datasets for testing purposes (TABLE 2).

TABLE 2. Dataset information.

No.	Dataset	|V|	|E|	Communities
1	Karate	34	78	2
2	Dolphins	62	159	4
3	Les Mis	77	508	11
4	Polbooks	105	441	3
5	Football	115	613	12
6	Jazz	198	2743	4
7	Email	1133	10903	10
8	Facebook	3622	72964	130
9	Amazon	13178	33767	4517
10	DBLP	114095	466761	4559
11	LFR-200k	200000	2003578	8000

1) **Zachary Karate Club Network:** [45] A social dataset derived from observations of a karate club, where nodes signify members and edges denote friendships.
2) **Dolphin Social Network:** [46] Comprising 62 bottlenose dolphins in New Zealand, nodes are dolphins and edges representing frequent interactions.
3) **Les Misérables Network:** [47] Based on Victor Hugo's novel, nodes are characters, and edges indicate their co-occurrences in scenes.
4) **Polbooks Network:** [48] This dataset explores political book purchases on Amazon, with nodes representing books and edges signifying co-purchases.
5) **American College Football Network:** [2] Featuring data on college football matches, nodes are teams, and edges represent matches played between them.
6) **Jazz Network:** [49] Depicting jazz dance scenarios, nodes are dancers, and edges signify partnerships where they have danced together.
7) **Email Communication Network:** [50] nodes represent email addresses, and edges denote at least one instance of communication between them.
8) **Facebook Social Network:** [51] nodes represent users of the social media platform Facebook, and edges signify friend relationships between them.
9) **Amazon Product Co-purchase Network:** [52] Nodes are products on Amazon, and edges connect those frequently purchased together.
10) **DBLP Dataset:** [53] A widely-used computer science research paper database, where nodes are authors and edges represent co-authorships.
11) **LFR:** [54] synthetic benchmark dataset.

C. EVALUATION METRIC

Modularity [9] and NMI (Normalized Mutual Information) [43] are established evaluation metrics utilized to assess the efficacy of community detection algorithms [44]. Due to the substantial classical resources required to simulate qubit behavior, the time complexity in our simulation experiments exceeds the inherent complexity of the quantum algorithms themselves [17]. Therefore, we do not consider the runtime comparisons among different algorithms in this study.

Modularity provides a quantitative measure of the quality of the community structure division within a network, by comparing the density of intra-community connections to that of inter-community connections. In the realm of community detection algorithms, a high modularity score indicates a network wherein nodes are more densely interconnected within their respective communities, while exhibiting relatively sparse connections with nodes belonging to different communities. The formula for calculating modularity is as follows:

$$Q = \frac{1}{2|E|}\sum_{ij}\left[A_{ij} - \frac{d_i d_j}{2|E|}\right]\delta(v_i, v_j) \tag{25}$$

where $|E|$ is the total number of edges in the graph; $\delta(v_i, v_j)$ is the step function, which equals 1 if v_i and v_j belong to the same community, and 0 otherwise.

NMI employs information entropy to gauge the discrepancy between the community partition produced by an algorithm and the ground-truth community partition:

$$\mathrm{NMI}(T, C) = \frac{-2\sum_{i=1}^{F_T}\sum_{j=1}^{F_C} F_{ij}\log\left(\frac{F_{ij}\cdot n}{F_i\cdot F_j}\right)}{\sum_{i=1}^{F_T} F_{i,s}\log\left(\frac{F_{i,s}}{n}\right) + \sum_{j=1}^{F_C} F_{s,j}\log\left(\frac{F_{s,j}}{n}\right)} \tag{26}$$

where $T = \{T_1, T_2, \ldots, T_k\}$ denotes the actual community partition, while $C = \{C_1, C_2, \ldots, C_k\}$ represents the community partition detected by the model. F is the confusion matrix, where F_{ij} indicates the number of nodes that belong to both community C_i and community T_j, reflecting the number of correctly identified nodes. $F_{i,s}$ is the sum of elements in the i-th row of matrix F; $F_{s,j}$ is the sum of elements in the j-th column of matrix F. A higher NMI value signifies greater similarity between the detected community structure T and the true community distribution C; NMI $= 1$ implies perfect alignment between T and C, whereas NMI $= 0$ indicates complete dissimilarity.

D. BASELINE METHODS

To assess the efficacy of our proposed method, we incorporate two distinct categories of baseline approaches: random walk-based techniques and other established community detection algorithms. For the random walk-based methods, we undertake a comprehensive comparative analysis utilizing four distinct algorithms:

1) **Infomap** [4]: optimizes community partitions by minimizing an objective function that is defined as the Huffman coding length of random walk paths.
2) **Walktrap** [5]: evaluates node similarities based on random walks, where nodes frequently co-visited across multiple walks are likely to be classified within the same community.
3) **DeepWalk** [13]: adopts a uniform random walk strategy and the skip Skip-Gram language model for embedding, emphasizing the capture of local structural features of the network.

TABLE 3. Modularity(Q) comparison of different algorithms on various real-world datasets.

No.	Dataset	EQDSC	Deepwalk [13]	Node2Vec [14]	Infomap [4]	Walktrap [5]	Louvain [7]	Leiden [8]	Fastgreedy [9]	LPA [6]
1	Karate	0.437	0.372	0.404	0.179	0.429	**0.445**	**0.445**	0.427	0.402
2	Dolphins	0.562	0.305	0.514	0.052	0.408	0.504	0.578	0.536	**0.773**
3	Les Mis	0.473	0.393	0.450	0.083	0.453	0.496	**0.507**	0.303	0.457
4	Polbooks	0.485	0.337	0.455	0.346	0.475	0.528	**0.531**	0.395	0.327
5	Football	0.601	0.595	0.428	0.016	0.427	0.589	0.575	0.495	**0.602**
6	Jazz	**0.493**	0.349	0.423	0.015	0.438	0.411	0.428	0.439	0.429
7	Email	**0.772**	0.228	0.517	0.537	0.423	0.696	0.68	0.527	0.554
8	Facebook	**0.625**	0.497	0.532	0.327	0.423	0.434	0.531	0.328	0.339
9	Amazon	**0.814**	0.783	0.722	0.640	0.591	0.795	0.776	0.746	0.727
10	DBLP	**0.855**	0.632	0.684	0.665	0.611	0.722	0.714	0.655	0.673
11	LFR-200k	**0.894**	0.752	0.693	0.794	0.541	0.783	0.646	0.678	0.567

TABLE 4. NMI comparison of different algorithms on various real-world datasets.

No.	Dataset	EQDSC	Deepwalk [13]	Node2Vec [14]	Infomap [4]	Walktrap [5]	Louvain [7]	Leiden [8]	Fastgreedy [9]	LPA [6]
1	Karate	0.839	0.574	0.733	0.146	0.563	**1.000**	**1.000**	0.577	0.703
2	Dolphins	0.689	0.269	0.557	0.019	0.554	**0.697**	**0.697**	0.438	0.635
3	Les Mis	0.662	0.631	0.632	0.429	0.602	0.533	**0.701**	0.635	0.569
4	Polbooks	0.682	0.419	0.628	0.556	0.574	0.642	**0.685**	0.593	0.497
5	Football	0.923	0.887	0.922	0.275	0.245	0.903	0.886	0.596	**0.935**
6	Jazz	0.541	0.372	0.391	0.476	0.415	**0.587**	0.532	0.579	0.569
7	Email	**0.680**	0.492	0.524	0.437	0.445	0.548	0.596	0.656	0.625
8	Facebook	**0.814**	0.540	0.523	0.673	0.614	0.785	0.536	0.447	0.477
9	Amazon	**0.761**	0.750	0.493	0.539	0.331	0.629	0.581	0.672	0.535
10	DBLP	**0.887**	0.588	0.563	0.731	0.572	0.813	0.522	0.757	0.470
11	LFR-200k	**0.903**	0.467	0.357	0.823	0.476	0.724	0.433	0.829	0.565

4) **Node2Vec** [14]: as an improved version of Deepwalk, uses biased random walks enhances the model's flexibility and capability in exploring both local and global network structures by adjusting the return and in-out parameters during walks.

DeepWalk and Node2Vec are first utilized to transform graph nodes into embedding vectors, then Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [42] are applied to group similar vectors into clusters, where each cluster signifies a community. Additionally, there are four other approaches to community detection that are also worth comparing:

5) **Louvain** [7]: optimizes modularity by iteratively aggregating nodes into communities based on greedy principles.
6) **Leiden** [8]: an improved version of the Louvain algorithm specifically designed to address its limitations by ensuring that detected communities are well-connected and internally cohesive.
7) **FastGreedy** [9]: optimizes modularity by iteratively merging pairs of communities that result in the largest increase in modularity until no further improvement is possible.
8) **LPA** [6]: discovers community structures by iteratively propagating labels among nodes based on neighbor majority.

By contrasting these methods, we aim to provide a comprehensive understanding of their strengths and limitations in community detection tasks.

E. OVERALL EVALUATION

The detailed examination of the results presented in Table 3 and Table 4 leads to several pivotal conclusions regarding the performance of EQDSC, in comparison to classical community detection algorithms. Across all datasets analyzed, the quantum algorithm surpasses the performance of all classical algorithms based on random walks. This observation underscores the robustness and adaptability of the quantum algorithm in diverse network environments, indicating its

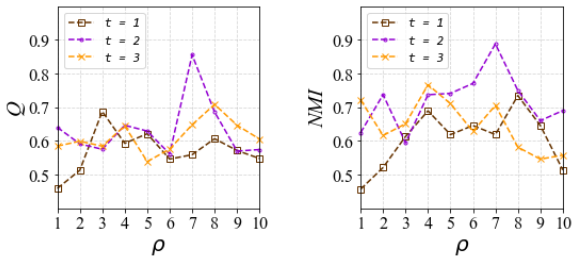


FIGURE 6. Density ρ versus Q and NMI when $r = 0.8$.

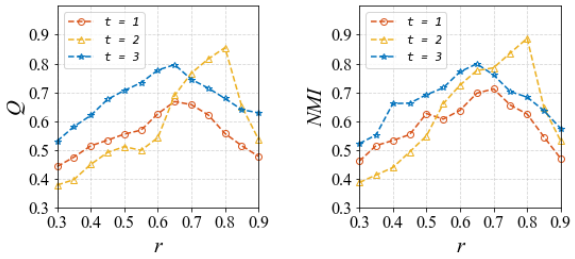


FIGURE 7. Radius r versus Q and NMI when $\rho = 7$.

capacity to uncover more accurate and meaningful community partitions.

Significantly, when utilized with extensive datasets like Facebook, Amazon, and DBLP, the quantum algorithm demonstrates impressive stability and consistently surpasses all traditional community detection algorithms, even those that do not rely on random walks. This observation underscores the algorithm's extraordinary capability to manage intricate and large-scale networks with notable accuracy and efficiency, even under demanding conditions. The the LFR-200k experiment further validates EQDSC's status as a superior community detection algorithm. In an ultra-large-scale synthetic network with a large number of small, potentially overlapping communities, EQDSC retains its advantages over classical methods—reinforcing its scalability, robustness, and quantum-driven efficiency.

However, it is important to note that in certain smaller datasets, the performance of the quantum algorithm may not exceed that of some classical community detection algorithms which are not based on random walks. The Quantum Density Clustering demonstrates considerable sensitivity to parameter selection. Within the context of small datasets, the scope for parameter optimization is severely restricted, consequently resulting in substantial instability in algorithmic performance. Despite not attaining optimal efficacy across all datasets, quantum algorithms consistently demonstrate performance within the upper-middle tier when juxtaposed against the majority of classical counterparts.

F. PARAMETER SENSITIVITY STUDY

This part focuses on the meticulous calibration of three pivotal parameters that are integral to the functionality of the proposed algorithm: the quantum walk step count t, the quantum clustering radius r, and the quantum density ρ. These parameters play a crucial role in dictating the algorithm's performance. We embark on a systematic exploration of the parameter space, aiming to identify configurations that optimize the algorithm's performance metrics. The potential quantum advantage in handling large datasets is evident and warrants further investigation. Therefore, we utilized the DBLP dataset, which is the largest among all datasets considered in our experiments, to illustrate the sensitivity of hyperparameters.

A crucial hyperparameter in our analysis is the number of steps involved in the quantum walk stage. Traditional network embedding techniques often limit their consideration to one-hop information, while more recent methods have begun to incorporate multi-hop information. The quantum walk approach we employed facilitates the extraction of structural data from multiple steps, making it essential to determine the optimal number of steps for our model. As depicted in Fig. 5 and 6, we conducted experiments by varying the number of steps t from 1 to 3. Our findings reveal that precision does not necessarily improve with an increase in steps; instead, the best results were obtained at $t = 2$. This outcome can be attributed to the fact that as the walk lengthens, more irrelevant information is introduced, leading to an averaging effect in the embedding of the target vertex. Therefore, we selected a walk length of $t = 2$ for our subsequent experiments.

Figure 6 illustrates a pattern where community detection metrics initially escalate as ρ increases, achieving a maximum point before declining. Figure 7 demonstrates an oscillatory behavior in the performance of community detection, marked by variations as the density r elevates. A larger r generally expands community boundaries, which may compromise coherence by including unrelated elements; conversely, a smaller r restricts these boundaries, improving precision but potentially disrupting natural connections. In a similar vein, a higher ρ conservatively identifies fewer but denser communities, possibly missing more subtle ones, while a lower ρ enhances sensitivity to smaller communities but risks inaccurately dividing cohesive ones.

The pattern observed highlights the presence of an ideal parameter setup that enhances the algorithm's performance. Notably, setting the quantum walk step t to 2 results in peak performance when $r = 0.8$ and $\rho = 7$. Under these conditions, an optimal balance is reached between cohesion within clusters and separation between clusters, which corresponds to the highest modularity and NMI scores obtained in our experiments.

G. ADVANTAGES OF EQDSC

Based on the numerical results presented above, we highlight the quantum advantages of the proposed approach.

The first advantage is efficiency of quantum superposition. Real-world networks often encompass billions of

vertices, making them extremely challenging for classical computers to process. In our experimental setup for the DBLP dataset, we set the vector dimensions for Deepwalk and Node2Vec to 512. However, to attain comparable effectiveness as the quantum method, specifically regarding the dimension of $|\varphi_i^t\rangle$, the embedding representations would require vectors with dimensions up to 114,095, which is equivalent to the number of vertices in DBLP. On traditional computing systems, each floating-point number occupies 4 bytes, meaning that a 114,095-dimensional vector would require approximately 456,380 bytes, for storage. In contrast, quantum computing demands merely 17 qubits to encode 114,095 distinct vector states.

The second advantage is the accelerated performance of the quantum walks compared to the sampling inherent in classical random walks. As documented in [55], quantum walk demonstrate exponentially faster hitting times [25] and quadratically faster mixing times [56] on certain networks against classical random walk. Moreover, the progression of quantum walk is not predominantly governed by the low-frequency components of the Laplacian spectrum, thereby granting them an enhanced capacity to discern and extract distinctive structural features of the network [57].

VI. CONCLUSION

In summary, the EQDSC algorithm presents a novel and promising approach for community detection in large-scale and complex networks. By embedding graphs into high-dimensional Hilbert space via quantum walks and utilizing quantum fidelity for clustering, EQDSC mitigates the ''curse of dimensionality'' and enables precise community identification. The discrete-time quantum walk eliminates the need for classical preprocessing, thus reducing computational overhead and improving performance. The experimental results on diverse real-world datasets demonstrate the robustness and scalability, particularly in large networks where classical methods struggle. For future work, we aim to provide an extensive evaluation of EQDSC on real quantum devices, exploring its performance and limitations in Noisy Intermediate-Scale Quantum (NISQ) computers. Another promising direction for future research involves investigating the potential of leveraging the Gaussian Boson Sampling algorithm on photonic quantum computers to enable efficient and scalable graph-theoretic community detection, thereby exploring new paradigms at the intersection of photonic quantum computing and network science.

REFERENCES

[1] S. Das and A. Biswas, ''Deployment of information diffusion for community detection in online social networks: A comprehensive review,'' *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 5, pp. 1083–1107, Oct. 2021.

[2] M. Girvan and M. E. Newman, ''Community structure in social and biological networks,'' *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002.

[3] P. Barucca, S. Battiston, F. Caccioli, G. Cimini, D. Garlaschelli, F. Saracco, T. Squartini, and G. Caldarelli, ''The physics of financial networks,'' *Nature Rev. Phys.*, vol. 3, pp. 490–507, Jul. 2021.

[4] M. Rosvall and C. T. Bergstrom, ''Maps of random walks on complex networks reveal community structure,'' *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.

[5] P. Pons and M. Latapy, ''Computing communities in large networks using random walks,'' *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[6] U. N. Raghavan, R. Albert, and S. Kumara, ''Near linear time algorithm to detect community structures in large-scale networks,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.

[7] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, ''Fast unfolding of communities in large networks,'' *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[8] V. A. Traag, L. Waltman, and N. J. van Eck, ''From Louvain to Leiden: Guaranteeing well-connected communities,'' *Sci. Rep.*, vol. 9, no. 1, p. 5233, Mar. 2019.

[9] M. E. J. Newman and M. Girvan, ''Finding and evaluating community structure in networks,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

[10] S. Kojaku, F. Radicchi, Y.-Y. Ahn, and S. Fortunato, ''Network community detection via neural embeddings,'' *Nature Commun.*, vol. 15, no. 1, p. 9446, Nov. 2024.

[11] F. Hu, J. Liu, L. Li, and J. Liang, ''Community detection in complex networks using Node2vec with spectral clustering,'' *Phys. A, Stat. Mech. Appl.*, vol. 545, May 2020, Art. no. 123633.

[12] H. Cai, V. W. Zheng, and K. C.-C. Chang, ''A comprehensive survey of graph embedding: Problems, techniques, and applications,'' *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, ''DeepWalk: Online learning of social representations,'' in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[14] A. Grover and J. Leskovec, ''node2vec: Scalable feature learning for networks,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[15] J. Zhang, J. Fei, X. Song, and J. Feng, ''An improved Louvain algorithm for community detection,'' *Math. Problems Eng.*, vol. 2021, pp. 1–14, Nov. 2021.

[16] F. Arute et al., ''Quantum supremacy using a programmable superconducting processor,'' *Nature*, vol. 574, pp. 505–510, Mar. 2019.

[17] X. Wang, S. Jian, K. Lu, Y. Zhang, and K. Liu, ''RED: Learning the role embedding in networks via discrete-time quantum walk,'' *Int. J. Speech Technol.*, vol. 52, no. 2, pp. 1493–1507, Jan. 2022.

[18] R. Sato, S. Haruta, K. Saito, and M. Kurokawa, ''QWalkVec: Node Embedding by Quantum Walk,'' in *Proc. Adv. Knowl. Discovery Data Mining*, 2024, pp. 93–104.

[19] X. Ye, G. Yan, and J. Yan, ''VQNE: Variational quantum network embedding with application to network alignment,'' in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2023, pp. 3105–3115.

[20] J. P. Moutinho, A. Melo, B. Coutinho, I. A. Kovács, and Y. Omar, ''Quantum link prediction in complex networks,'' *Phys. Rev. A, Gen. Phys.*, vol. 107, no. 3, Mar. 2023, Art. no. 032605.

[21] X. Wang, K. Lu, Y. Zhang, and K. Liu, ''QSIM: A novel approach to node proximity estimation based on discrete-time quantum walk,'' *Int. J. Speech Technol.*, vol. 51, no. 4, pp. 2574–2588, Apr. 2021.

[22] F. Acasiete, F. P. Agostini, J. K. Moqadam, and R. Portugal, ''Implementation of quantum walks on IBM quantum computers,'' *Quantum Inf. Process.*, vol. 19, no. 12, Dec. 2020.

[23] M. Gong et al., ''Quantum walks on a programmable two-dimensional 62-qubit superconducting processor,'' *Science*, vol. 372, no. 6545, pp. 948–952, May 2021.

[24] C. F. A. Negre, H. Ushijima-Mwesigwa, and S. M. Mniszewski, ''Detecting multiple communities using quantum annealing on the D-wave system,'' *PLoS ONE*, vol. 15, no. 2, Feb. 2020, Art. no. e0227538.

[25] S. Akbar and S. K. Saritha, ''Towards quantum computing based community detection,'' *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100313.

[26] K. Mukai and N. Hatano, ''Discrete-time quantum walk on complex networks for community detection,'' *Phys. Rev. Res.*, vol. 2, no. 2, Jun. 2020, Art. no. 023378.

[27] M. Faccin, P. Migdał, T. H. Johnson, V. Bergholm, and J. D. Biamonte, "Community detection in quantum complex networks," *Phys. Rev. X*, vol. 4, no. 4, Oct. 2014, Art. no. 041047.
[28] E. Song, "Quantum walk on simplicial complexes for simplicial community detection," *Quantum Inf. Process.*, vol. 23, no. 6, p. 199, May 2024.
[29] S. G. Roy and A. Chakrabarti, "A novel graph clustering algorithm based on discrete-time quantum random walk," in *Quantum Inspired Computational Intelligence*, 2017, pp. 361–389.
[30] C. Umeano, S. Scali, and O. Kyriienko, "Quantum community detection via deterministic elimination," 2024, *arXiv:2412.13160*.
[31] S. Akbar and S. K. Saritha, "Quantum inspired community detection for analysis of biodiversity change driven by land-use conversion and climate change," *Sci. Rep.*, vol. 11, no. 1, p. 14332, Jul. 2021.
[32] T. G. Wong, "Faster quantum walk search on a weighted graph," *Phys. Rev. A, Gen. Phys.*, vol. 92, no. 3, Sep. 2015, Art. no. 032320.
[33] J. Jung, W. Jin, and U. Kang, "Random walk-based ranking in signed social networks: Model and algorithms," *Knowl. Inf. Syst.*, vol. 62, no. 2, pp. 571–610, Feb. 2020.
[34] Y. Aharonov, L. Davidovich, and N. Zagury, "Quantum random walks," *Phys. Rev. A, Gen. Phys.*, vol. 48, no. 2, pp. 1687–1690, 1993.
[35] J. Kempe, "Quantum random walks: An introductory overview," *Contemp. Phys.*, vol. 44, no. 4, pp. 307–327, Jul. 2003.
[36] M. Hillery, J. Bergou, and E. Feldman, "Quantum walks based on an interferometric analogy," *Phys. Rev. A, Gen. Phys.*, vol. 68, no. 3, Sep. 2003, Art. no. 032314.
[37] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 212–219.
[38] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, and C. I. Eke, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Eng. Appl. Artif. Intell.*, vol. 110, Apr. 2022, Art. no. 104743.
[39] E. Aïmeur, G. Brassard, and S. Gambs, "Machine learning in a quantum world," in *Proc. Conf. Can. Soc. Comput. Stud. Intell.*, Berlin, Germany. Springer, 2006, pp. 431–442.
[40] P. Li, J. Guo, B. Wang, and M. Hao, "Quantum circuits for calculating the squared sum of the inner product of quantum states and its application," *Int. J. Quantum Inf.*, vol. 17, no. 5, Aug. 2019, Art. no. 1950043.
[41] D. Morgenstern and R. Bellman, *Adaptive Control Processes—A Guided Tour*. Princeton, NJ, USA: Princeton Univ. Press, 1961, p. 255.
[42] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
[43] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.
[44] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Comput. Surv.*, vol. 50, no. 4, pp. 54:1–54:37, 2017.
[45] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropolog. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.
[46] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, Sep. 2003.
[47] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*. Reading, MA, USA: Addison-Wesley, 1993.
[48] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.
[49] P. M. Gleiser and L. Danon, "Community structure in Jazz," *Adv. Complex Syst.*, vol. 6, no. 4, pp. 565–573, Dec. 2003.
[50] R. Guimera, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 6, Dec. 2003, Art. no. 065103.
[51] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 548–556.
[52] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proc. IEEE 12th Int. Conf. Data Mining*, Brussels, Belgium, Dec. 2012, pp. 745–754.
[53] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: Membership, growth, and evolution," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Philadelphia, PA, USA, Aug. 2006, pp. 44–54.
[54] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110, doi: 10.1103/physreve.78.046110.
[55] J. Kempe, "Discrete quantum walks hit exponentially faster," in *Proc. 7th Int. Workshop Randomization Approximation Techn. Comput. Sci.*, Princeton, NJ, USA, 2003, pp. 354–369.
[56] A. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, "Quantum walks on graphs," in *Proc. 33rd Annu. ACM Symp. Theory Comput.*, Jul. 2001, pp. 50–59.
[57] 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 (extended abstract)," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 3132–3133.

SHUO HE received the B.S. and M.S. degrees from Shanghai Jiao Tong University, in 1999 and 2002 respectively. He is currently pursuing the Eng.D. degree in computer science and engineering with Fudan University, Shanghai. He is with the Research Institute of FinTech, China UnionPay. His research interests include quantum computing, machine learning, software architecture, network security, and privacy computing.

BOXUAN AI received the Master of Engineering (M.Eng.) degree in electrical and computer engineering from Cornell University, in 2021. He is currently a Researcher with the Research Institute of FinTech, China UnionPay. His research interests include artificial intelligence, graph theory, quantum computing, and quantum machine learning.

PENGFEI GAO is currently the Vice Dean of the Research Institute of FinTech, China UnionPay. She has been long engaged in research on the payment industry and financial technology, with 17 years of experience in the fintech field. Her main research areas in financial technology include privacy computing, quantum computing, algorithm analysis, blockchain, digital currency, and artificial intelligence.

HONGBAO LIU received the master's degree from Beijing University of Posts and Telecommunications. He is currently a Senior Researcher with the Research Institute of FinTech, China UnionPay, with research interests include data mining, artificial intelligence, quantum computing, and quantum machine learning.

TAO TANG received the B.S. and Ph.D. degrees in mechanical engineering from Shanghai Jiao Tong University, in 2011 and 2018, respectively, and the Post Doctor degree in computer science and engineering from Fudan University, Shanghai. From 2018 to 2025, he was a Researcher with the Research Institute of FinTech, China UnionPay. His research mainly engaged in the fields of risk prevention and control, fraud detection, machine learning, quantum computing, graph computing, and other new technologies.

JIE WU was the Executive Vice Dean of the Institute of Financial technology, Fudan University. Concurrently, he holds the positions of the Vice Dean with the School of Computer Science and Technology, a Research Fellow, and the Doctoral Supervisor. He is the Director of the Engineering Research Center for Network Information Security Auditing and Monitoring under the Ministry of Education, and the Vice Dean of the National Institute of Secrecy.