An R-Convolution Graph Kernel Based on Fast Discrete-Time Quantum Walk

Yi Zhang, Lulu Wang, Richard C. Wilson, *Senior Member, IEEE*, and Kai Liu

***Abstract*— In this article, a novel R-convolution kernel, named the fast quantum walk kernel (FQWK), is proposed for unattributed graphs. In FQWK, the similarity of the neighborhood-pair substructure between two nodes is measured via the superposition amplitude of quantum walks between those nodes. The quantum interference in this kind of local substructures provides more information on the substructures so that FQWK can capture finer-grained local structural features of graphs. In addition, to efficiently compute the transition amplitudes of multistep discrete-time quantum walks, a fast recursive method is designed. Thus, compared with all the existing kernels based on the quantum walk, FQWK has the highest computation speed. Extensive experiments demonstrate that FQWK outperforms state-of-the-art graph kernels in terms of classification accuracy for unattributed graphs. Meanwhile, it can be applied to distinguish a larger family of graphs, including cospectral graphs, regular graphs, and even strong regular graphs, which are not distinguishable by classical walk-based methods.**

***Index Terms*— Discrete-time quantum walk (DTQW), graph classification, graph kernel, R-convolution kernel.**

I. Introduction

A. Motivation

GRAPHS are important structures for information representation, in which nodes and edges, respectively, represent the entities and the relationships in the real world. Graph processing has been widely used in many scientific fields, such as image processing [1], biochemical research [2], social network [3], and natural language processing [4]. Within these fields, graph comparison plays a core role in data mining and target recognition. For instance, two molecules with the same chemical properties usually have similar structures [5]. Thus, people can successfully perform a prediction for an unknown molecule via graph comparison with known ones.

As a general and effective similarity measurement for graphs, graph kernels have been extensively investigated over the past decades. Most existing methods belong to the group of R-convolution kernels, which are the earliest and most successful graph kernels in the whole literature [6]. Different from other graph comparison methods, such as graph edit distance [7] or the Bregman divergences [8], graph similarity is measured in the method of an R-convolution kernel via decomposing the graph into certain substructures and computing the frequency of the same substructures of the input graphs. These substructures are, e.g., walks for the random walk kernel (RWK) [9] and subtrees for the Weisfeiler–Lehman kernel (WLK) [10].

Although many sophisticated R-convolution kernels have been designed, there are still several problems.

1) The neglect of the relative locations of substructures deteriorates the classification accuracy because reliable structural correspondences between substructures cannot be established [11].
2) These kernels are generally not expressive enough to measure the similarity without the assistance of graph attributes, which may not be available or maybe expensive to capture.
3) Some similar but nonisomorphic graphs cannot be distinguished by these kernels due to the existence of cospectral graphs and regular ones [12], which leads to failures when trying to distinguish different graphs.

Since when it was first proposed in [13], the discrete-time quantum walk (DTQW) has drawn much interest from both machine learning and quantum computation communities. The reasons why the DTQW is suitable to connect quantum mechanics and R-convolution kernels are twofold: 1) due to quantum interference, the DTQW can achieve better discrimination for graphic data, compared with traditional methods and 2) in contrast to the continuous-time quantum walk, the local structural features of graph can be located in the step-by-step simulation of DTQW, which is crucial to design R-convolution kernels. However, the simulation complexity of the DTQW is quadratic to that of a random walk because of the approximately quadratic state space, which becomes the fatal bottleneck for graph processing.

In this article, a fast quantum walk kernel (FQWK) is proposed based on the DTQW. Via computing the

Manuscript received October 26, 2019; revised August 3, 2020 and September 24, 2020; accepted September 26, 2020. Date of publication October 16, 2020; date of current version January 5, 2022. This work was supported in part by the Natural Science Foundation of Hunan Province under Grant 2019JJ40340 and in part by the National Natural Science Foundation of China under Grant 61701502. *(Corresponding author: Lulu Wang.)*

Yi Zhang is with the School of Computer Science, National University of Defense Technology, Changsha 410073, China, and also with Central Future Works Technology Co., Ltd., Beijing 100094, China.

Lulu Wang is with the Artificial Intelligence Research Center, National Innovation Institute of Technology, Beijing 100071, China (e-mail: wanglulunudt@163.com).

Richard C. Wilson is with the Department of Computer Science, University of York, York YO10 5GH, U.K.

Kai Liu is with the School of Computer Science, National University of Defense Technology, Changsha 410073, China.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3027687

quantum superposition amplitudes of DTQWs on two graphs, the neighborhood-pair substructure matching can be achieved. Because the quantum amplitude of one substructure will be affected by its adjacent ones via quantum interference, the relative location information of substructures is naturally exploited in the matching computation. Meanwhile, a fast recursive method for simulating DTQW is proven to reduce the runtime consumption of the new kernel. Experimental results show the effectiveness of FQWK in terms of classification accuracy, runtime speed, and distinguishing ability.

B. Related Works

1) R-Convolution Kernels: The concept of R-convolution kernel was first proposed in [14]. A graph is first decomposed into some certain substructures, e.g., walks or subtrees, and then, graph similarity is measured by computing the frequency of the isomorphic substructure pairs. Here, we refer to the definition of the cross-product graph kernel in [15], which is a common instance of the generic R-convolution kernel. For two graphs G_A and G_B, $\{\mathcal{G}_{A;1}, \ldots, \mathcal{G}_{A;N_A}\}$ and $\{\mathcal{G}_{B;1}, \ldots, \mathcal{G}_{B;N_B}\}$ are the substructure sets of G_A and G_B, respectively. $\mathcal{G}_{i;j}$ is the jth substructure of Graph G_i, $i = A, B$, in which N_A and N_B are the total numbers of different substructures, respectively. A kernel function for graph G_A and G_B is shown as follows:

$$\mathcal{K}(G_A, G_B) = \sum_{j_A=1}^{N_A} \sum_{j_B=1}^{N_B} \Delta\left(\mathcal{G}_{A;j_A}, \mathcal{G}_{B;j_B}\right) \tag{1}$$

where Δ denotes a Dirac kernel shown as follows:

$$\Delta\left(\mathcal{G}_{A;j_A}, \mathcal{G}_{B;j_B}\right) = \begin{cases} 1, & \text{if } \mathcal{G}_{A;j_A} \cong \mathcal{G}_{B;j_B} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Here, $\mathcal{G}_{A;j_A} \cong \mathcal{G}_{B;j_B}$ indicates that the substructure $\mathcal{G}_{A;j_A}$ is isomorphic (or approximately isomorphic) to $\mathcal{G}_{B;j_B}$.

Many R-convolution kernels have been proposed, and they can be categorized into three classes, namely, walk-based kernels, subtree-based kernels, and subgraph-based kernels.

a) Walk-based kernels: The random walk graph kernel (RWK) was proposed in [9] based on the calculation of the same walks between two graphs. The main drawback of RWK is that the totter problem is not considered, i.e., a walker from one node to another will possibly trackback to the starting node in the next step. This will cause the fact that much redundant information of nodes and edges is contained in a random walk, which deteriorates the computation performance of this kernel. To overcome this problem, the shortest path kernel (SPK) was proposed [16], which is computed by comparing the shortest paths of the graphs. Because there is no traceback in the shortest path, the aforementioned totter issue is settled. However, SPK only considers one path between any pair of nodes. The GraphHopper kernel (GHK) [17] fixes this problem and performs a convolution computation by counting all the subpath similarities. Meanwhile, backtrackless walk kernel (BWK) was proposed in [12] using the zeta function that has also no traceback in the corresponding loop. No repeated nodes and edges are contained in the loop except the start and end nodes. Therefore, the totter problem is utterly solved. The return probability-based graph kernel (RetGK) [18] utilizes the return probability features of the random walk to improve the RWK so that various node attributes can be effectively exploited.

b) Subtree-based kernels: One common defect of the walk-based graph kernel is the limitation on structural information because the substructures used are comparatively simple. A feasible method to overcome this problem is to construct graph kernels based on subtrees. The widely used Weisfeiler–Lehman subtree graph kernel (WLK) [10] is defined by comparing isomorphic subtree structures. Due to the effective Weisfeiler–Lehman algorithm, WLK not only can represent more powerful information than walk-based graph kernels but also has a computation complexity of $\mathcal{O}(hVE)$, where h, V, and E denote the depth of the Weisfeiler–Lehman algorithm, the number of nodes, and the number of edges, respectively. The multiscale Laplacian graph kernel proposed in [19] and the aligned subtree kernels (ASKs) proposed in [11] and [20] are the improvements of WLK. Meanwhile, the invariant kernel is explored for graphs with high-dimensional and continuous node attributes [21]. However, the repeated-structure problem still exists for subtree-based graph kernels. Nevertheless, the classification accuracy of subtree-based kernels is quite high, especially in the data sets with typical tree structures, such as molecular structures.

c) Subgraph-based kernels: From the definition in (1), the R-convolution kernel is actually a subgraph-based kernel. Walks and subtrees are only two special cases of subgraphs. Thus, a walk-based kernel and a subtree-based kernel can only capture limited structural features of graphs.

One of the classic subgraph-based kernels is the all graphlet kernel (AGK) [22]. A graphlet is a subgraph with only three to five nodes. AGK is constructed by comparing the distribution of different graphlets in two graphs. The deep graphlet kernel [23] is explored to learn the optimal representative features. In addition, the attributes of nodes and edges are considered in [24] for subgraph matching, which is another instance of subgraph-based kernels for attributed graphs. However, the subgraph decomposition is a complex procedure with high time consumption, e.g., the time complexity of AGK is exponential to the graphlet size. Therefore, the approximate methods in [25] appear for large graph applications.

In addition, some novel frameworks for R-convolution kernels are proposed in [26] and [27], which shows that the performance of R-convolution kernels can be improved via preprocessing graphs with valid optimal node assignment and exploiting substructures at multiple different scales, respectively.

2) Graph Kernels Based on Quantum Walk: Recent graph kernel research based on quantum walk [28] can be classified into two parts.

The Jensen–Shannon kernels (QJSK) were proposed in [29]–[32] based on the quantum Von Neumann entropy. Graph similarity is measured by the mutual information entropy between the density matrices of quantum walks on the two graphs. Bai *et al.* [33] improved the classification accuracy by modifying the kernels using the Jensen–Tsallis divergence information entropy. All of these kernels are called information theoretic kernels [15]. However, they suffer from

the problems of high computation complexity of calculating the quantum entropy and the neglect of the local structural features of graphs.

Distinct from the aforementioned methods, some other works appeared for graph classification, which uses the quantum walk as a tool for graph feature extraction or subgraph matching. In [34], it is found that the amplitude of the quantum interference approximates zero between similar nodes of two graphs. This property can be used to design a probability model to quantitatively evaluate the graph similarity. However, the application of this method is limited by its time complexity of $\mathcal{O}(N^6)$. Based on the density matrices of quantum walks, the edge-based matching and subtree-based matching are used to construct the convolution kernels in [11] and [35], respectively. The essence of these methods is using the quantum walk as an attribute enhancement of the edges or nodes. The problems for the traditional R-convolution kernels are still unsettled.

C. Main Contributions

The characteristics of the proposed kernel and the main contributions of this article are listed as follows.

1) A novel R-convolution graph kernel FQWK is proposed based on the fast DTQW. The FQWK characterizes the neighborhood-pair substructure between two nodes via the superposition amplitude of all quantum walks of a particular length that joins the two nodes. This allows the interference between different paths, which has been shown in previous works to allow better characterization of graph substructure and a more powerful representative ability. Experimental results show that FQWK outperforms the state-of-the-art methods on classification accuracy for most of the data sets.
2) To achieve the neighborhood-pair substructure matching effectively, a fast simulation method is proposed for DTQW. Although the superposed transition amplitudes of the original quantum walks need to be used for matching tasks, there is no need to calculate the transition matrix of the multistep DTQW. Instead, a fast recursive method is designed and proved, which can greatly reduce the computation time of the proposed kernel. Experimental results show that FQWK is the fastest one among all the quantum walk-based kernels.
3) The high-dimensional structural information of graphs can be explored via quantum mechanism so that the slight structural differences between some similar but nonisomorphic graphs can be located and even amplified via quantum constructive interference. Therefore, the proposed graph kernel can be applied to distinguish cospectral graphs, regular graphs, and even strong regular graphs.

D. Outline of This Article

The remainder of this article is structured as follows. First, the background of the graph concept and DTQW is introduced in Section II. Then, in Section III, the R-convolution kernel based on fast DTQW is proposed. The detailed method and the formal computation procedure of the novel kernel are discussed. In Section IV, numerical experiments are given to show the effectiveness of the proposed kernel for unattributed graph classification. Finally, the concluding remarks are made in Section V.

II. Background

A. Basic Concepts

Graphs are widely used to represent structural and relational information in a way that is abstracted from the actual data.

Definition 1 (Graph): A graph is a tuple $G(V, E)$, where V represents all the nodes in the graph with a set of the adjacent relation $E \subseteq V \times V$.

In this article, we will mainly focus on connected and unattributed graphs, in which all the nodes have no attributes and all the edges have no weights and directions. Only the structural information can be used to perform the graph analysis.

Definition 2 (Walk): A walk $w = (v_0, v_1, \ldots, v_k)$ in a graph $G(V, E)$ is a node sequence with $v_i \in V$, and $(v_i, v_{i+1}) \in E$. The length $\ell(w) = k$ of a walk is the number of edges traversed in the sequence.

Definition 3 (Walk Set): The order-k walk set of a graph is the set of all walks of length k, whichs exist in a graph: $W^{(k)} = \{w|\ell(w) = k\}$.

Definition 4 [Subgraph (or Substructure)]: A graph $G'(V', E')$ is a subgraph of graph $G(V, E)$, if and only if $V' \subseteq V$ and $E' \subseteq E$.

Definition 5 [Graph Isomorphic (or Graph Matching)]: A graph $G(V, E)$ is isomorphic to $G'(V', E')$ if there exists at least one bijective mapping $f : V \to V'$ so that $\forall v_1, v_2 \in V, (v_1, v_2) \in E \Leftrightarrow (f_{v_1}, f_{v_2}) \in E'$.

B. Discrete-Time Quantum Walk

The DTQW is the quantum counterpart of the discrete-time classical random walk [36]. In DTQW, the states need to specify both the current and the previous location of the walk because of the reversibility of quantum processes. Therefore, the state space for DTQW is the directed edge set.

Definition 6 (Directed Edge Set E_d): In an unattributed graph $G(V, E)$, every edge $e(u, v) \in E$ is replaced with a pair of directed edges $e_d(u, v)$ and $e_d(v, u)$. These directed edges construct the directed edge set E_d, which can be shown as follows:

$$E_d = \{e_d(u, v), e_d(v, u)|e(u, v) \in E\}. \tag{3}$$

For a demo graph shown in Fig. 1(a), the corresponding graph with directed edges is given in Fig. 1(b). On the directed edge set, the directed line graph can be constructed.

Definition 7 [Directed Line Graph $G_L(V_L, E_L)$]: For a graph $G(V, E)$, its directed edge set is E_d. The directed line graph $G_L(V_L, E_L)$ is a dual representation of the original graph. The node set and edge set are defined as follows:

$$\begin{aligned} V_L &= E_d, \\ E_L &= \{(e_d(i, m), e_d(m, j)) \in E_d \times E_d\}. \end{aligned} \tag{4}$$

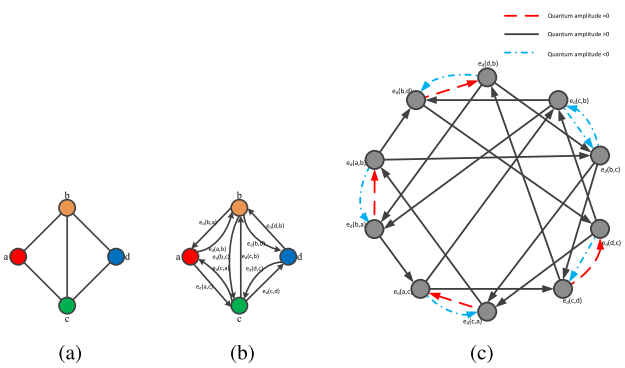


Fig. 1. (a) Demo graph with four nodes. (b) Demo graph in (a) whose edges are all replaced by the directed edge set E_d. (c) Directed line graph $G_L(V_L, E_L)$. Note that DTQW based on the Grover diffusion matrix can be regarded as a walk on G_L. The red dashed arcs, black solid arcs, and blue dashed–dotted arcs represent that the quantum transition amplitudes are 0, positive, and negative, respectively.

Fig. 1(c) shows the directed line graph of the demo graph in Fig. 1(a). In the directed line graph $G_L(V_L, E_L)$, each node corresponds to a unique directed edge residing on the corresponding edge in the original graph $G(V, E)$.

In DTQW, if there is a directed edge from a node $v_L \in V_L$ to a node $u_L \in V_L$, the transition of the quantum walk on $G(V, E)$ is allowed from the edge corresponding to v_L to the edge corresponding to u_L, and vice versa. Therefore, different from the classical random walk, the DTQW on a graph G can be regarded as a walk performed on its directed line graph G_L. The state space of the walk is the node set V_L, and the transitions are constrained by the directed edge E_L in the directed line graph.

We denote the state corresponding to the quantum walker being on the directed edge $e_d(u, v)$ as $|uv\rangle$. It can be interpreted as that a quantum particle is currently at node v and has a previous step at node u. The general state of DTQW is

$$|\phi\rangle = \sum_{e_d(u,v)\in E_d} \alpha_{uv}|uv\rangle \tag{5}$$

where the quantum amplitude α_{uv} is complex, i.e., $\alpha_{uv} \in \mathbb{C}$. The probability that the quantum walker is at the state $|uv\rangle$ is given by $\Pr(|uv\rangle) = \alpha_{uv}\alpha^*_{uv}$, where α^*_{uv} is the complex conjugate of α_{uv}.

At each step, the evolution of the DTQW is governed by a transition matrix U. The entries of U determine the transition probabilities between states, i.e., $|\phi_{t+1}\rangle = U|\phi_t\rangle$. Since the evolution of the walk is linear and conserves probability, the matrix U must be unitary, i.e., $U^{-1} = U^\dagger$, where $U^\dagger$ denotes the Hermitian transpose of U.

It is common to adopt the Grover diffusion matrix as the transition matrix. This matrix does not distinguish any forward nodes (i.e., those other than the current and previous nodes), and among such diffusion matrices, it is the furthest from the identity one. The entries of the transition matrix U are shown as follows:

$$U_{im;nj} = \begin{cases} A_{im}A_{nj}\left(\dfrac{2}{d_m} - \delta_{ij}\right), & \text{if } m = n \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Here, $U_{im;nj}$ shows the quantum amplitude for the transition $e_d(i, m) \to e_d(n, j)$, d_m denotes the node degree for node m, and δ_{ij} is the Kronecker delta, i.e., $\delta_{ij} = 1$ if $i = j$; otherwise, $\delta_{ij} = 0$. A is the adjacency matrix of the original graph.

Different from the random walk where the probability propagates, what propagates during the DTQW is quantum amplitude. Given a state $|im\rangle$, the Grover matrix assigns the same amplitude to all transitions $|im\rangle \to |mj\rangle$ and a different amplitude to the transition $|im\rangle \to |mi\rangle$, where i and j are the adjacent nodes of m. Therefore, in the directed line graph, as shown in Fig. 1(c), when the particle tracks back and $d_m > 2$, the transition amplitude is negative for this step (the blue dashed–dotted arc); when the particle tracks back and $d_m = 2$, the transition amplitude of this step is equal to 0 (the red dashed arc); otherwise, it is positive (the black solid arc). Obviously, the totter problem of the random walk will be settled via the amplitude penalties for backtracks.

Furthermore, it is observed in [37] and [38] that the directed line graph possesses some special properties that are not available in the original graph. For instance, compared to the original graph, the directed line graph spans a higher dimensional feature space and thus exposes richer graph characteristics. This is because the cardinality of the node set for the directed line graph is greater than, or at least equal to, that of the original graph. This property suggests that the DTQW may reflect richer graph characteristics than the classical random walk and the continuous-time quantum walk, which are both the walks on the original graph.

III. R-Convolution Kernel Based on Fast Discrete-Time Quantum Walk

In this section, a novel substructure matching method is designed based on the DTQW via introducing the neighborhood-pair substructure. One of the issues with using the DTQW for probing graph structure is that the transition matrix U is of size $2|E|\times 2|E|$ and so potentially quadratically bigger that the random walk equivalent. In turn, this means a computation time of $O(N^6)$ in the worst case of dense edges, which is quite impractical even for moderate-sized graphs. In order to solve this problem, a fast recursive method to calculate an alternative transition matrix is proposed. All these preparations lead to the newly proposed kernel FQWK.

A. Neighborhood-Pair Substructure Matching Based on Discrete-Time Quantum Walk

In this article, the structural features based on the neighborhood-pair substructure are analyzed, which is a kind of auxiliary substructure. The formal definition is as follows.

Definition 8 (k-Level Neighborhood-Pair Substructure $S^{(k)}_{ab}$): In a graph, for each node pair a and b, the k-level neighborhood-pair substructure $S^{(k)}_{ab}$ is constructed which contains all the k-length walks between a and b

$$S^{(k)}_{ab} = \left\{w \in W^{(k)} | v_0 = a, v_k = b\right\}. \tag{7}$$

Fig. 2(a) illustrates that in the demo graph in Fig. 1(a), there are totally 5 three-length walks from node a to node b, which make up the auxiliary substructure $S^{(3)}_{ab}$. All the walk-based

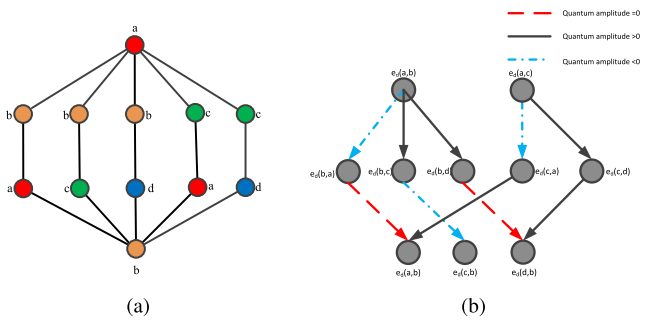


Fig. 2. (a) Five three-length walks from node a to node b. (b) Corresponding five two-step quantum walks on the directed line graph G_L.

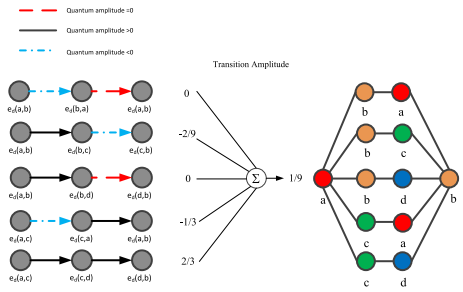


Fig. 3. Quantum transition amplitudes of the five three-length walks and the neighborhood-pair substructure $S_{ab}^{(3)}$.

kernels can be regarded as counting over the auxiliary substructure $S_{ab}^{(k)}$. For example, there are five walks contributing to the RWK. The SPK and GHK compare only one walk which is the nonrepeating path $(acdb)$. Similarly, the BWK counts the single nonbacktracking walk $(acdb)$. However, the BWK, GHK, and SPK do not account for the edges (ab) and (bc) and the RWK treats all five paths identically. Therefore, the topology characteristics of this substructure cannot be well represented by the aforementioned simple features.

Our goal is to consider all these walks in $S_{ab}^{(k)}$ via a superposition of the quantum amplitudes. As shown in Fig. 2(b), the 5 three-length walks correspond to the 5 two-step DTQWs on the directed line graph given in Fig. 1(c). According to (6), the transition amplitudes of each walk can be calculated. And we use different line types to denote that the walk is positive, negative, or 0. In Fig. 3, we show explicitly that among all of the 5 three-length walks, there are 2 walks with quantum amplitudes 0, 2 with negative amplitudes, and 1 with positive amplitude.

As we noted earlier, it is too expensive to compute the amplitude for each individual walk. Instead, we would like to compute the superposition of all the walks in $S_{ab}^{(k)}$, for example with $S_{ab}^{(2)}$ this includes all walks of length 2 and therefore every walk which begins at a and ends at b via any single intermediate node. This can be calculated by the summation of amplitudes over all possible intermediate nodes

$$M_{ab} = \sum_{m=1}^{N}\sum_{n=1}^{N} U_{am;nb}. \tag{8}$$

In a similar way, U^t provides the amplitudes for all $(t+1)$-length walks, and again we can compute the transition amplitude between a and b by summing the t-step quantum amplitudes over all possible first and last steps

$$M_{ab}^{(t)} = \sum_{m=1}^{N}\sum_{n=1}^{N} U_{am;nb}^{t}. \tag{9}$$

This superposes all t-step quantum walks which start at a and end at b, i.e., all the walks in $S_{ab}^{(t+1)}$. This defines the transition matrix $M^{(t)}$.

Note that the superscript t means the tth power of a matrix, while the superscript (t) is the level index. In Fig. 3, the summation of the amplitudes of these 5 walks achieves 1/9, which is just the superposition amplitude of the auxiliary substructure $S_{ab}^{(3)}$.

It is interesting that quantum interference exists in the quantum walk on a graph and is therefore present in our transition matrix $M^{(t)}$. Destructive interference will happen on the intersection of two isomorphic substructures with opposite amplitudes [34], which is always exploited to locate the local symmetric subgraphs. On the other hand, constructive interference may occur on the crossings of different quantum walks, thereby the slight structural difference can be amplified.

We choose this particular transition matrix for our graph kernel based on three factors. First, it is more compact than the DTQW, since it is only of size $|V|\times|V|$. Second, it retains information about constructive and destructive interference since it is a superposition of individual walks. Finally, as we demonstrate in the next section, it can be computed efficiently in only $O(t|V|^3)$ steps in contrast to the $O(t|V|^6)$ steps of the DTQW.

B. Fast Simulation Method

According to the definition, after a t-step DTQW, the superposition amplitude matrix $M^{(t)}$ is computed as

$$M_{ij}^{(t)} = \sum_{m=1}^{N}\sum_{n=1}^{N} U_{im;nj}^{t} \neq \left[M^t\right]_{ij}. \tag{10}$$

It is obvious that though the size of the transition matrix M is only $|V|\times|V|$, as shown in Fig. 4, $M^{(t)}$ cannot be computed via the exponential operation of M. However, the computation complexity of the exponential operation of U is still about $\mathcal{O}(|V|^6)$, which is potentially quadratic to the transition matrix of the random walk. Therefore, it is not time-acceptable to perform graph processing directly.

In Fig. 4, the demo graph contains four nodes and eight directed edges. The size of the transition matrix of the random walk is 4×4 because the state space of the random walk is the node set. However, the size of the quantum walk transition matrix is 8×8 as the state space is the directed edge set.

Therefore, we need to find a fast method to simulate the DTQW and compute the matrix $M^{(t)}$. In Fig. 5, an arbitrary t-step DTQW from node i to node j is shown as an example. This walk consists of a $(t-1)$-step quantum walk from i to n and a one-step walk from n to j. We find a recursive method for fast computing the superposition transition matrices of $t-2$, $t-1$, and t-step DTQW.

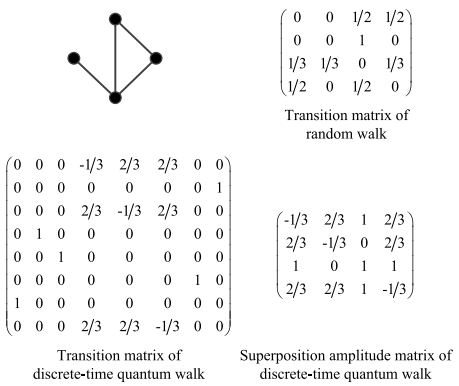


Fig. 4. Demo graph and the comparisons of its three transition matrices.

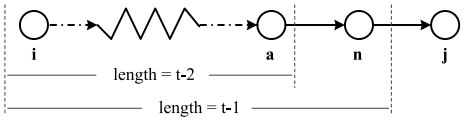


Fig. 5. t-step quantum walk from node i to node j. Node n is the previous step of node j. Node a is the previous step of node n.

Theorem 1: Assume that a graph has N nodes. A is the adjacency matrix, and d_n is the degree of node n. Let a diagonal matrix $D = \mathrm{diag}((2/d_1), (2/d_2), \ldots, (2/d_N))$, and let another matrix $Q = DA$. The series of the superposition amplitude matrices $\{M^{(t)}\}$ can be computed as

$$M^{(t)} = \begin{cases} AQ - 2D^{-1}, & t = 1 \\ MQ - A, & t = 2 \\ M^{(t-1)}Q - M^{(t-2)}, & t \geq 3. \end{cases} \tag{11}$$

Proof:

1) When $t = 1$, for every item in matrix M

$$\begin{aligned} M_{ij} &= \sum_{m,n} U_{im;nj} = \sum_{m} U_{im;mj} \\ &\quad \left(U_{im;nj} \neq 0 \Rightarrow m = n\right) \\ &= \sum_{m} \frac{2A_{im}A_{mj}}{d_m} - \sum_{m} \delta_{ij} A_{im} A_{mj} \\ &= \left[AQ - 2D^{-1}\right]_{ij}. \end{aligned} \tag{12}$$

2) When $t = 2$

$$\begin{aligned} M^{(2)}_{ij} &= \sum_{m,n} U^2_{im;nj} = \sum_{m,n,a} U_{im;an}U_{an;nj} \\ &= \sum_{n,m,a} U_{im;an}U_{an;nj} \\ &= \sum_{n,m,a} \frac{2U_{im;an}A_{an}A_{nj}}{d_n} - \sum_{n,m,a} U_{im;an}\delta_{aj}A_{an}A_{nj} \\ &= \sum_{n,m,a} \frac{2U_{im;an}A_{nj}}{d_n} - \sum_{n,m} U_{im;jn}A_{jn}A_{nj} \\ &\quad \left(\text{for an unattributed graph } A_{jn} = A_{nj}\right) \\ &= \sum_{n,m,a} \frac{2U_{im;an}A_{nj}}{d_n} - \sum_{n,m} U_{im;jn} \\ &= \sum_{n} \frac{2M_{in}A_{nj}}{d_n} - \sum_{n} U_{ij;jn} \\ &= \sum_{n} \frac{2M_{in}A_{nj}}{d_n} - \left(\sum_{n} \frac{2A_{ij}A_{jn}}{d_j} - A_{ij}A_{ji}\right) \\ &= [MQ]_{ij} - A_{ij}. \end{aligned} \tag{13}$$

3) When $t \geq 3$

$$\begin{aligned} M^{(t)}_{ij} &= \sum_{m,n} U^t_{im;nj} = \sum_{m,n,a} U^{t-1}_{im;an}U_{an;nj} \\ &= \sum_{n,m,a} U^{t-1}_{im;an}U_{an;nj} \\ &= \sum_{n,m,a} \frac{2U^{t-1}_{im;an}A_{an}A_{nj}}{d_n} - \sum_{n,m} U^{t-1}_{im;jn}A_{jn}A_{nj} \\ &\quad \left(U^{t-1}_{im;an} \neq 0 \Rightarrow A_{an} = 1\right) \\ &= \sum_{n,m,a} \frac{2U^{t-1}_{im;an}A_{nj}}{d_n} - \sum_{n,m} U^{t-1}_{im;jn} \\ &= \sum_{n} \frac{2M^{(t-1)}_{in}A_{nj}}{d_n} - \sum_{n,m} U^{t-1}_{im;jn} \\ &= \left[M^{(t-1)}Q\right]_{ij} - \sum_{n,m} U^{t-1}_{im;jn}. \end{aligned} \tag{14}$$

Then, we focus on how to compute the later part on the right-hand side of the above equation

$$\begin{aligned} \sum_{n,m} U^{t-1}_{im;jn} &= \sum_{n,m,a} U^{t-2}_{im;aj}U_{aj;jn} \\ &= \sum_{n,m,a} \frac{2U^{t-2}_{im;aj}A_{jn}}{d_j} - \sum_{n,m} U^{t-2}_{im;nj} \\ &= \sum_{n} \frac{2M^{(t-2)}_{ij}A_{jn}}{d_j} - M^{(t-2)}_{ij} \\ &= M^{(t-2)}_{ij}. \end{aligned} \tag{15}$$

Thus, we get $M^{(t)} = M^{(t-1)}Q - M^{(t-2)}$. This is exactly the formula stated in Theorem 1 when $t \geq 3$.

□

From Theorem 1, all the operations in this recursion are no more than $\mathcal{O}(N^3)$ for a graph with N nodes. It means that the series matrices $M^{(t)}$ can be computed in cubic time of the graph size.

C. Kernel Design

1) Kernel Definition for Two Graphs: Here, we will propose a novel graph kernel based on the fast DTQW, named FQWK.

As a novel R-convolution kernel, the FQWK for two graphs, G_A and G_B, is defined as follows:

$$\mathcal{K}_{FQWK}(G_A, G_B) = \sum_{t} \mathcal{K}_t(G_A, G_B). \tag{16}$$

After the tth step, a subkernel $\mathcal{K}_t(G_A, G_B)$ will be performed to count all the tth-level isomorphic neighborhood-pair substructures. The formal definition of the subkernel is

$$\mathcal{K}_t(G_A, G_B) = \sum_{m,n \in G_A} \sum_{u,v \in G_B} \Delta\left(S^{(t)}_{mn}, S^{(t)}_{uv}\right) \tag{17}$$

where Δ is a Dirac function as follows:

$$\Delta\left(S_{mn}^{t}, S_{uv}^{t}\right) = \begin{cases} 1, & \text{if } M_{mn}^{(t)} = M_{uv}^{(t)} \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Here, $M_{mn}^{(t)}$ and $M_{uv}^{(t)}$ are the quantum superposition amplitude of the t-level neighborhood-pair substructure $S_{mn}^{(t)}$ and $S_{uv}^{(t)}$, in the graph G_A and G_B, respectively.

2) Kernel Computation for a Graph Data Set: In the abovementioned definition, the computation of the simulation of the DTQW is the most time-consuming procedure. Therefore, for a graph data set with many graphs, this procedure needs to be performed only once on each graph actually, before the pairwise kernel computation.

Suppose that a graph data set includes K graphs and each one has N unattributed nodes.

First, a T-step DTQW will be processed step-by-step on every graph. By using the fast simulation method, the superposition matrices $M^{(t)}$ can be computed according to Theorem 1. In order to reduce the amplitude comparisons, for the ith matrices $M^{(i)}$ of every graph G, the histogram of all the N^2 items is constructed as the ith feature $F_G^{(i)}$ of graph G. Therefore, for each graph, a T-dimension feature vector can be extracted. The pseudocode for the implementation of graph feature extraction algorithm is shown in Algorithm 1.

Algorithm 1 Graph Feature Extraction Algorithm

Require: Graph G, with N nodes; $d_i, i = 1, \ldots, N$ are the node degrees; T is the fixed step of discrete-time quantum walk, $T \geq 3$.
Ensure: $F_G^{(t)}$: the t-step feature of G, $t = 1, \ldots, T$.

function $GraphFeature(G, T)$
  Get the adjacency matrix A
  $D = diag\left(\frac{2}{d_1}, \frac{2}{d_2}, \cdots, \frac{2}{d_N}\right)$
  $M \leftarrow ADA - 2D^{-1}$
  $F_G^{(1)} \leftarrow histogram(M)$
  $M^{(2)} \leftarrow MDA - A$
  $F_G^{(2)} \leftarrow histogram\left(M^{(2)}\right)$
  for $t = 3 \rightarrow T$ **do**
    $M^{(t)} \leftarrow M^{(t-1)}DA - M^{(t-2)}$
    $F_G^{(t)} \leftarrow histogram\left(M^{(t)}\right)$
  end for
end function

Then, for each graph pair G_A and G_B in the data set, every feature pair $F_{G_A}^{(t)}$ and $F_{G_B}^{(t)}$ will be compared to obtain the number of the neighborhood-pair substructures with the same quantum superposition amplitude. For convenience of calculations, for every feature pair $F_{G_A}^{(t)}$ and $F_{G_B}^{(t)}$, an aligned and padding operation will be performed to obtain $\mathbf{f}_{G_A}^{(t)}$ and $\mathbf{f}_{G_B}^{(t)}$, so that the inner product $\langle \mathbf{f}_{G_A}^{(t)}, \mathbf{f}_{G_B}^{(t)} \rangle$ is just the frequency of the isomorphic substructure pairs. The implementation of the proposed kernel FQWK is shown in Algorithm 2. For the input graph data set, the algorithm of FQWK can output a kernel matrix $\mathcal{K}_{FQWK} \in \mathbb{R}^{K \times K}$, where the entry $\mathcal{K}_{FQWK}(G_i, G_j)$ denotes the number of isomorphic neighborhood-pair substructures of G_i and G_j.

Algorithm 2 FQWK Algorithm

Require: Graph dataset $\{G_1, G_2, \ldots, G_K\}$, where K denotes the number of graphs in the dataset; T is the fixed step of node-to-node discrete-time quantum walk, $T \geq 3$.
Ensure: Graph kernel $\mathcal{K}_{FQWK} \in \mathbb{R}^{K \times K}$

1: **for** $i = 1 \rightarrow K$ **do**
2: $\quad F_{G_i}^{(t)} \leftarrow GraphFeature(G_i, T), t = 1, \ldots, T$
3: **end for**
4: **for** each graph pair G_i and G_j **do**
5: $\quad$ **for** $t = 1 \rightarrow T$ **do**
6: $\quad\quad \mathbf{f}_{G_i}^{(t)}, \mathbf{f}_{G_j}^{(t)} = alignment\left(F_{G_i}^{(t)}, F_{G_j}^{(t)}\right)$
7: $\quad\quad \mathcal{K}_t(G_i, G_j) \leftarrow \left\langle \mathbf{f}_{G_i}^{(t)}, \mathbf{f}_{G_j}^{(t)} \right\rangle$
8: $\quad$ **end for**
9: $\quad \mathcal{K}_{FQWK}(G_i, G_j) = \sum_{t=1}^{T} \mathcal{K}_t(G_i, G_j)$
10: **end for**

Finally, we will evaluate the time complexity of calculating the kernel matrix. For each graph G of the data set, a T-step DTQW needs to be performed first to extract the T-dimension feature vector F_G. It will cost no more than $\mathcal{O}(KTN^3 + KTN^2 \log N)$ for lines 1–3 in Algorithm 2. In lines 4–9, to compute the kernel matrix, for all the graph pairs, the neighborhood-pair matching procedures are performed, which costs about $\mathcal{O}(K^2TN^2)$. Overall, the total time complexity of N^{2}QWK is about $\mathcal{O}(KT(N^3 + N^2 \log N) + K^2TN^2)$.

D. Discussion

1) Kernel Validation: According to the Mercer' theorem [39], a valid graph kernel must be symmetric and positive semidefinite (p.s.d.). Here, we will give a brief proof of the validation of FQWK.

Theorem 2: The proposed graph kernel FQWK is valid.
Proof: The FQWK is symmetric and p.s.d, and thus, it is valid.

1) *Symmetric:* Based on the definition of FQWK in (16), it is obvious that $\mathcal{K}_{FQWK}(G_A, G_B) = \mathcal{K}_{FQWK}(G_B, G_A)$.
2) *p.s.d.:* In the definition of FQWK, after the tth step of quantum walk, a matching subkernel $\mathcal{K}_t(G_A, G_B)$ will be performed to count all the tth-level isomorphic neighborhood-pair substructures. It is known that the summation kernel of some p.s.d. ones is still p.s.d. Because the Dirac function is p.s.d, the subkernel $\mathcal{K}_t(G_A, G_B)$ is p.s.d. Therefore, FQWK is p.s.d.

□

2) Theoretical Analysis of FQWK: From the abovementioned definition, FQWK seems an instance of walk-based graph kernels. It is close to RWK and BWK because we only use DTQW to replace classical random walk and backtrackless walk. Therefore, the computation procedure of FQWK is comparatively simple compared with other R-convolution kernels.

Besides, compared with RWK and BWK, the proposed FQWK can represent more powerful structural characteristics. The reason is analyzed using the following example. Fig. 6(a)

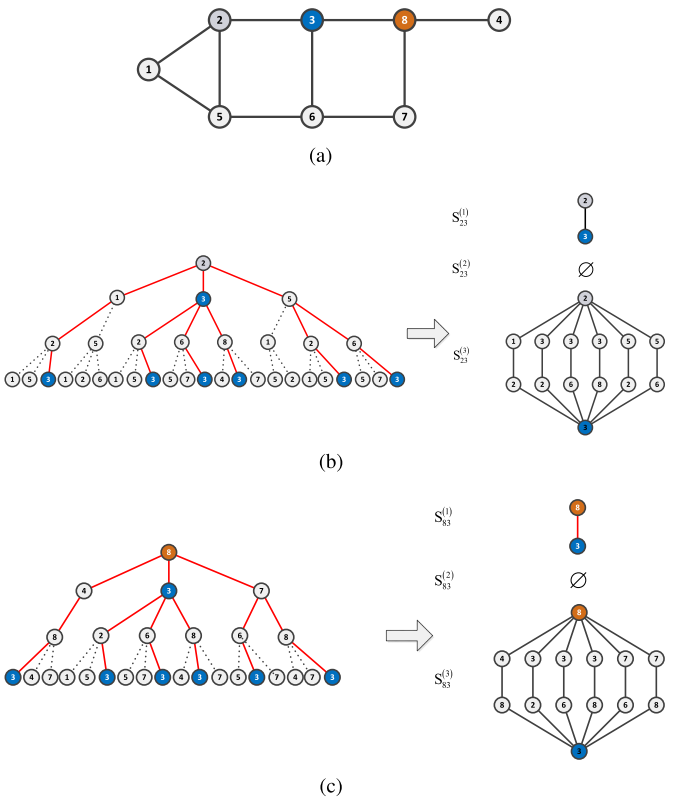


Fig. 6. (a) Example graph with node 8. (b) Three-level neighborhood of node v_2 and the substructures $S_{23}^{(1)}$, $S_{23}^{(2)}$, and $S_{23}^{(3)}$. (c) Neighborhood of v_8 and the substructures $S_{83}^{(1)}$, $S_{83}^{(2)}$, and $S_{83}^{(3)}$.

shows an example graph with eight nodes; focus on two node pairs: node v_2 to node v_3 and node v_8 to node v_3. The neighborhood of node v_2 and node v_8 and the one-to-three level neighborhood-pair substructures are shown in Fig. 6(b) and Fig. 6(c), respectively. It is obvious that both $S_{23}^{(1)}$ and $S_{83}^{(1)}$ have only one walk (path). $S_{23}^{(2)}$ and $S_{83}^{(2)}$ are both empty. Meanwhile, there are six three-length walks included in $S_{23}^{(3)}$ and $S_{83}^{(3)}$, and only one path (or backtrackless walk) exists in each of them, namely, $v_2v_5v_6v_3$ and $v_8v_7v_6v_3$, respectively. Based on the counting of walks or paths, the neighborhood-pair substructures are isomorphic when using the RWK and other walk-based kernels. Therefore, node v_2 and node v_8 can be matched. However, it is obvious that the node v_2 and node v_8 should not be matched as the adjacent node v_1 and v_4 cannot be matched. Therefore, classification error will be caused by using the RWK and BWK if such a kind of structure exists. If the proposed FQWK is used, the superposition amplitude of the substructure $S_{v_2v_3}^{(3)}$ is $-1/9$, which is different from the substructure $S_{v_8v_3}^{(3)}$ with amplitude 1, so the two substructures do not match. Therefore, the proposed FQWK outperforms the existing walk-based kernels as more powerful features can be represented by the auxiliary substructure $S_{ab}^{(k)}$. The false-positive substructure matching will be reduced, and the classification performance can be improved by the FQWK.

In fact, FQWK is a member of the subgraph-based kernel family as the core procedure is the neighborhood-pair substructure matching based on the comparison of the quantum superposition amplitudes over the sets $S_{ab}^{(k)}$, rather than the

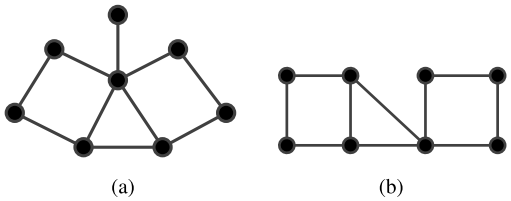


Fig. 7. Two example graphs with node 8. (a) Eight-node example graph with two squares and one triangle. (b) Another eight-node example graph with two squares and one triangle but the relative position is different with (a).

counting of individual walks. Compared with the existing subgraph-based kernels, the FQWK is better because the relative position between the substructures is considered. For the graph in Fig. 6(a) and two graphs in Fig. 7, one triangle and two squares are included in all the three example graphs, but the relative position among these three graphlets are different. Therefore, the three graphs should not be matched. In the computation of the traditional subgraph-based kernels, the relative position between the adjacent substructures is usually ignored in the traditional kernel computation, which will lead to the error matching of the three graphs. In FQWK, the auxiliary substructure $S_{ab}^{(k)}$ can be regarded as the intersection between the k-level neighborhood of node a and node b, and quantum superposition amplitude of every neighborhood-pair substructure $S_{ab}^{(k)}$ will be affected by two kinds of quantum interference.

1) The inter-interference among all the quantum walks that are included in the neighborhood-pair substructure. In Fig. 6(b) and 6(c), the red solid lines denote these included quantum walks. The quantum interference among them will affect the superposition amplitude of the substructure $S_{ab}^{(k)}$.
2) The intra-interference from the adjacent neighborhood-pair substructures. In Fig. 6(b) and 6(c), the black dashed lines denote the quantum walks excluded in $S_{ab}^{(k)}$. However, via quantum interference, the superposition amplitude of one substructure will also be affected by its adjacent ones.

Therefore, compared with other subgraph-based methods, FQWK can extract more powerful substructure features by catching more extra information on the location relationships of local substructures.

Previous work on quantum walks [28], [32], [34] has shown that quantum walks are sensitive to structures in which random walks are not. We demonstrate, empirically in Section IV, that our transition matrix retains these properties and produces excellent performance on standard data sets.

IV. Experiments

In this section, the newly proposed kernel FQWK is evaluated on classification problems for unattributed graphs. Here, we perform a fast DTQW in the computation of FQWK. In the test, we choose the user-defined fixed step T as 10. The discussion of the choice of parameter T is given in Section IV-B3. We also compare FQWK with several other popular graph classification methods as follows.

1) *RWK:* The version of [40] is used to test.

2) *WLK [10]:* For the sake of fairness, the highest dimension of the Weisfeiler–Lehman isomorphism is set to be 10.
3) *GHK [17]:* The Dirac kernel is used to be the base node kernel.
4) *AGK [22]:* The graphlet size is chosen to be 4.
5) *ASK [11]:* The density matrix of the continuous-time quantum walk is used to enhance the node attributes. The entropic representation layer is also set to be 10. For WLK, GHK, and ASK, we use node degree as the original node attribute for unattributed graphs.
6) *Quantum Jensen–Shannon kernel (QJSK) [32]:* The density matrix of ten-step DTQW is used for the computation of Von Neumann entropy.
7) *Edge-Based Matching Kernel Based on DTQW (DQMK) [35]:* A ten-step DTQW is used to perform edge matching, and the highest layer of the depth-based representation is set to be 10.
8) *PATCHY-SAN Convolutional Neural Network (PSCN) [41]:* Similar to the behavior of CNNs on images, PSCN first extracts fixed-sized local patches from nodes and neighborhoods as the receptive fields for convolution filters and then uses the graph canonization tool NAUTY to apply CNNs on these patches. Here, we set the receptive field size as 10.
9) *Deep Graph Convolutional Neural Network (DGCNN) [42]:* DGCNN inherits the PSCN idea of imposing an order for graph nodes but integrates this step into the network structure, namely the SortPooling layer. We set the sortpooling parameter as 0.6 and the learning rate as 0.0001. Both PSCN and DGCNN are trained with the epochs of 150 and the batch size of 25.

All the experiments were tested in MATLAB R2016b on an Intel Xeon Core E5-1620 CPU with 8-GB memory. All the runtime consumption tests were executed with a single thread.

A. Data Sets

Both real-world and synthetic data sets are used to evaluate graph kernels.

1) Real-World Data Sets: To evaluate the classification accuracy, seven chemical data sets with ground-truth labels are collected [43]. All the chemical molecules are converted into unattributed graphs by representing atoms as nodes and the covalent bonds as edges.

AIDS consists of graphs representing molecular compounds from the AIDS Antiviral Screen Database of Active Compounds. There are 2000 elements totally (1600 inactive elements and 400 active elements), which represents molecules with activity against HIV or not. COX2 has 467 cyclooxygenase-2 inhibitors and has been assembled in this data set. DHFR consists of 756 inhibitors of dihydrofolate reductase for the inhibition of the enzymatic reduction that converts dihydrofolate to tetrahydrofolate. ENZYME collects 600 graphs representing tertiary protein structures, each labeled with one of the six EC top-level classes. MUTAG includes 188 graphs representing mutagenetic compounds, labeled according to their mutagenic effects. NCI1 is a set of 4110 graphs representing a subset of chemical compounds screened for activity against nonsmall cell lung cancer cell lines. The Predictive Toxicology Challenge (PTC) data set records the carcinogenicity of several hundred chemical compounds. These graphs are very small and sparse, with 20–30 nodes and 25–40 edges. We select the graphs of male mice (PTC MM) for evaluation. There are 336 test graphs in the MM class.

TABLE I
DETAILED INFORMATION OF THE REAL-WORLD DATA SETS

Dataset	#Set	#Class	Avg.#Node	Avg.#Edge
AIDS	2000	2	15.69	16.20
COX2	467	2	41.22	43.45
DHFR	756	2	42.43	44.54
ENZYMES	600	6	32.63	62.14
MUTAG	188	2	17.93	19.79
NCI1	4110	2	29.87	32.30
PTC_MM	336	2	13.97	14.32

TABLE II
THREE NONISOMORPHIC GRAPH DATA SETS. IN COSGRAPH, EVERY COSPECTRAL GRAPH PAIR IS USED AS A TEST. IN REGGRAPH AND SRGRAPH, PAIRWISE COMPARISONS OF THE GRAPHS IN EACH CLASS ARE EVALUATED

Dataset	#Set	#Class	Avg.#Node	#Test Pair
CosGraph	10096	5048	10	5048
RegGraph	6490	31	16.34	885128
SRGraph	7303	11	37.63	5099490

Table I shows the statistical information of these chemical data sets.

2) Synthetic Data Sets: In order to further evaluate the distinguishing ability of the kernels, some special data sets are chosen [44], as shown in Table II. CosGraph includes 5048 pairs of ten-node graphs. Each pair of graphs has the same graph spectrum, which is called a cospectral graph pair. RegGraph and SRGraph consist of 31 classes of regular graphs and 11 classes of strong regular graphs, respectively. Within each class, every graph is regular or strong regular but not isomorphic with others.

B. Results

1) Test Result for Graph Classification: Graph classification is an important application, which is quite related to the measurement of the graph similarity. Here, we investigate the performance of the novel graph kernel for the data sets in Table I.

For each data set, tenfold cross-validation tests are performed using all the mentioned methods. For graph kernels, the classification accuracy is accomplished via using the C-support vector machine (C-SVM) with the optimal parameters [45]. The average accuracy ($\pm$ standard error) and runtime results are reported in Tables III and IV.

Table III reports the average classification accuracy and the relative standard error of each method and data set. For the data sets AIDS, COX2, DHFR, ENZYMES, and PTC_MM, the new kernel FQWK achieves the highest accuracy, which yields a remarkable improvement compared with the other state-of-the-art kernels and graph neural networks. Only for MUTAG and NCI1, GHK and WLK show a slightly better performance than FQWK, respectively. Although the classifi-

TABLE III

AVERAGE ACCURACY (IN % ± STANDARD ERROR) ON GRAPH CLASSIFICATION BENCHMARK DATA SETS

Methods \ Datasets	AIDS	COX2	DHFR	ENZYMES	MUTAG	NCI1	PTC_MM
RWK	80.00±.28	78.20±.61	60.96±.56	14.20±.42	80.56±.72	53.16±.30	61.59±.86
WLK	98.89±.07	79.38±.57	82.43±.45	37.69±.62	83.22±.89	**81.87±.20**	61.72±.81
GHK	99.33±.58	78.65±.89	79.77±.75	37.34±.58	**85.40±.85**	68.02±.36	61.45±.59
AGK	99.07±.07	78.20±.61	60.96±.56	28.88±.61	82.01±.90	62.54±.25	63.65±.82
PSCN	**99.53±.03**	77.66±.14	60.00±.27	15.50±.09	83.16±1.1	56.91±.09	59.41±.34
DGCNN	98.50±.03	78.26±.46	66.67±.26	40.12±.11	77.78±.51	69.34±.44	54.55±.76
ASK	96.74±.12	78.17±.61	74.15±.48	30.26±.60	84.96±.84	64.52±.24	61.15±.81
QJSK	79.57±.28	78.73±.61	78.41±.47	34.61±.62	83.62±.68	67.20±.22	60.58±.85
DQMK	79.99±.67	78.15±.64	76.77±.96	28.91±.71	76.42±.88	65.18±.33	61.09±.73
FQWK	**99.53±.06**	**80.87±.57**	**82.92±.41**	**41.55±.61**	84.27±.83	80.39±.19	**63.77±.78**

TABLE IV

COMPUTATION TIME OF THE GRAPH CLASSIFICATION METHODS (FOR GRAPH KERNELS, WE FOCUS ON THE COMPUTATION TIME OF KERNEL MATRIX. FOR GRAPH NEURAL NETWORKS, THE TRAIN TIME OF THE NETWORK MODEL IS LISTED. THE LAST FOUR KERNELS ARE DESIGNED BASED ON THE QUANTUM WALK)

Methods \ Datasets	AIDS	COX2	DHFR	ENZYMES	MUTAG	NCI1	PTC_MM
RWK	37′7″	6′4″	8′16″	4′20″	15″	197′13″	50″
WLK	26″	7″	23″	15″	2″	1′57″	2″
GHK	24′41″	2′42″	6′23″	4′49″	14″	164′11″	35″
AGK	26″	15″	23″	32″	3″	1′34″	4″
PSCN	1′48″	47″	1′21″	1′7″	14″	5′57″	20″
DGCNN	4′5″	1′30″	2′32″	1′54″	23″	11′43″	44″
ASK	92′25″	28′50″	61′16″	25′5″	1′25″	773′58″	2′51″
QJSK	141′58″	21′18″	58′2″	23′50″	1′3″	611′10″	2′32″
DQMK	97′44″	28′51″	76′37″	88′56″	1′26″	1203′9″	2′53″
FQWK	38′52″	8′47″	13′57″	5′15″	27″	269′50″	58″

cation for unattributed graphs is quite difficult, FQWK turns out to be the best competitor in terms of accuracy on most of these benchmark data sets.

The reasons for the effectiveness are threefold.

1) Due to the powerful discrimination of quantum interference, FQWK can establish the substructure location relationship, which is deficient in the traditional methods RWK, WLK, GHK, AGK, PSCN, and DGCNN.
2) Compared with the information-theoretic kernel QJSK that computes the graph similarity via the global structural information, FQWK can reflect richer local convolutional characteristics of graphs via performing step-by-step evolution of DTQW.
3) The quantum walk kernels, ASK and DQMK, perform the edge matching and subtree matching to compute graph similarity, while FQWK can capture finer-grained features of graphs via matching the neighborhood-pair substructures.

Table IV shows the runtime comparison of these classification methods for each data set. FQWK achieves a moderate performance. However, compared with ASK, QJSK, and DQMK, FQWK is the fastest one among all the kernels based on the quantum walk. Via the fast recursive method, the runtime of FQWK is close to that of RWK.

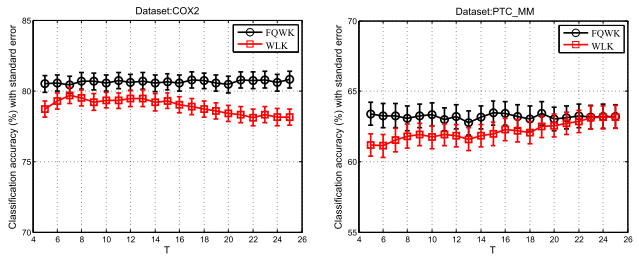
2) Test Result for Distinguishing Ability: Some similar and nonisomorphic graphs are usually difficult to distinguish via inexact graph comparison methods. Therefore, a graph kernel cannot be applied to some kinds of graphs. Here, the distinguishing ability for similar graphs is used to compare the applicability of these graph kernels. We utilize the failure rate as the applicability measurement for the graph kernels. Table V shows the failure rates of these graph kernels for distinguishing the similar graph pairs collected in Table II, including the cospectral graphs, regular graphs, and strong regular graphs.

TABLE V

FAILURE RATES (%) FOR DISTINGUISHING THE NONISOMORPHIC GRAPHS (- DENOTES THAT THE TEST CANNOT BE FINISHED BY THE KERNEL IN TEN DAYS). HERE, ONLY GRAPH KERNELS ARE CONSIDERED TO BE TESTED

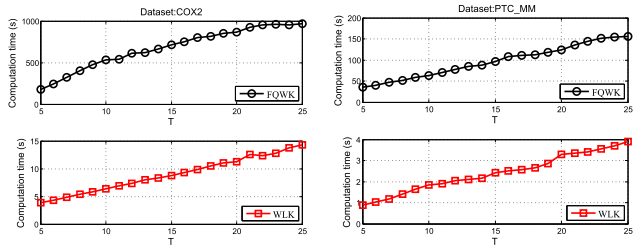
Kernel Name	CosGraph	RegGraph	SRGraph
RWK	100	100	100
WLK	1.66	100	100
GHK	18.82	0.12	100
AGK	5.96	4.87	3.82
ASK	33.16	1.14	95.96
QJSK	33.16	1.14	13.60
DQMK	**0**	0.11	-
FQWK	**0**	**0.02**	**0.0016**

RWK is the worst kernel, which cannot be used to distinguish these similar graphs. WLK can only locate the difference of the cospectral graphs but fails for regular graphs. Generally, compared with the traditional kernels, the quantum walk kernels achieve better distinguishing ability because the slight topological difference will be amplified by quantum interference. In particular, FQWK has the lowest failure rates for all the three kinds of nonisomorphic graphs and, thus, outperforms the other kernels. The powerful ability of FQWK for distinguishing nonisomorphic graphs, even including strong



(a) (b)

Fig. 8. (a) Classification accuracy (%) and standard error of using FQWK and WLK for the data set COX2 with different T's. (b) Classification accuracy (%) and standard error of using FQWK and WLK for the data set PTC_MM with different T's.



(a) (b)

Fig. 9. (a) Computation time (s) of using FQWK and WLK for the data set COX2 with different T's. (b) Computation time (s) of using FQWK and WLK for the data set PTC_MM with different T's.

regular graphs, comes from the fact that both the interquantum and intraquantum interferences of the neighborhood-pair substructures are included.

3) Test Result About the Parameter T: In the proposed algorithm, the maximum walk step T should be user-defined. This parameter is also used in QJSK, ASK, and some other kernels to restrict the walk steps. In essence, due to the fact that the amplitude propagation occurs in the neighborhood of nodes, the maximum walk step T actually limits the order of the neighborhood-pair substructures that are explored in the algorithm. In the former tests, we set $T = 10$.

In this part, the influence of T is under consideration. Our proposed kernel FQWK and WLK are tested, for example, using different T's for the data sets COX2 and PTC_MM. We change T from 5 to 25. The accuracy and computation time results are shown in Figs. 8 and 9, respectively. We find out that the accuracy keeps almost unchanged. However, the computation time has a nearly linear growth with the increase in T. The reason is that much of the structural information has already been contained in the low-order neighborhood of the node, which is quite discriminative for the graph classification based on the local structure matching. With the increase in T, there are barely isomorphic and large substructures between two graphs so that the classification accuracy nearly maintains. Therefore, it is reasonable that we choose the same parameter T for all graph kernels that extract features via node neighborhoods.

V. Conclusion

In this article, a novel R-convolution graph kernel FQWK is proposed based on the fast DTQW. Via the powerful quantum interference of the DTQW, more reliable location correspondences between the neighborhood-pair substructures are located so that FQWK can extract finer-grained structural features, which the traditional R-convolution kernels are deficient to. Extensive experiments demonstrate that the classification accuracy of FQWK outperforms that of the state-of-the-art graph kernels, and the distinguishing ability for nonisomorphic graphs is significantly improved.

In addition, a novel and fast simulation method is proposed for computing the transition matrices of the DTQW so that FQWK can achieve the highest computation speed among all the existing kernels based on the quantum walk.

Acknowledgment

The authors appreciate the kind comments and professional criticisms of the anonymous reviewers, which greatly enhanced the overall quality of this article and opened numerous perspectives geared toward improving the work.

References

[1] V. T. Ta, O. Lézoray, A. Elmoataz, and S. Schüpp, "Graph-based tools for microscopic cellular image segmentation," *Pattern Recognit.*, vol. 42, no. 6, pp. 1113–1125, 2009.

[2] J. W. Raymond and P. Willett, "Maximum common subgraph isomorphism algorithms for the matching of chemical structures," *J. Comput. Aided Mol. Des.*, vol. 16, no. 7, pp. 521–533, 2002.

[3] W. Fan, "Graph pattern matching revised for social network analysis," in *Int. Conf. Database Theory*, 2012, pp. 1–5.

[4] A. C. N. Ngomo and F. Schumacher, "BorderFlow: A local graph clustering algorithm for natural language processing," in *Proc. Int. Conf. Comput. Linguistics Intell. Text Process.*, 2009, pp. 1–8.

[5] P. Mahé and J. P. Vert, "Graph kernels based on tree patterns for molecules," *Mach. Learn.*, vol. 75, no. 1, pp. 3–35, 2009.

[6] N. M. Kriege, M. Neumann, C. Morris, K. Kersting, and P. Mutzel, "A unifying view of explicit and implicit feature maps for structured data: Systematic studies of graph kernels," *CoRR*, vol. abs/1703.00676, pp. 1–5, Oct. 2017.

[7] H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recognit. Lett.*, vol. 18, no. 8, pp. 689–694, Aug. 1997.

[8] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, no. 4, pp. 1705–1749, 2005.

[9] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Learning Theory Kernel Machine*. Berlin, Germany: Springer-Verlag, 2003, pp. 129–143.

[10] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman graph kernels," *J. Mach. Learn. Res.*, vol. 12, pp. 2539–2561, Sep. 2011.

[11] L. Bai, L. Rossi, and E. Hancock, "An aligned subtree kernel for weighted graphs," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 30–39.

[12] F. Aziz, R. C. Wilson, and E. R. Hancock, "Backtrackless walks on a graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 977–989, Jun. 2013.

[13] Y. Aharonov, L. Davidovich, and N. Zagury, "Quantum random walks," *Phys. Rev. A, Gen. Phys.*, vol. 48, p. 1687, Aug. 1993.

[14] D. Haussler, "Convolution kernels on discrete structures," Dept. Comput. Sci., Univ. California, Oakland, CA, USA, Tech. Rep. UCS-CRL-99-10, 1999.

[15] L. Bai, "Information theoretic graph kernels," Ph.D. dissertation, Dept. Comput. Sci., Univ. York, York, U.K., 2014.

[16] K. M. Borgwardt and H. Kriegel, "Shortest-path kernels on graphs," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2005, p. 8.

[17] C. Morris, N. M. Kriege, K. Kersting, and P. Mutzel, "Faster kernels for graphs with continuous attributes via hashing," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 216–224.
[18] Z. Zhang, M. Wang, Y. Xiang, Y. Huang, and A. Nehorai, "RetGK: Graph kernels based on return probabilities of random walks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3964–3974.
[19] R. Kondor and H. Pan, "The multiscale Laplacian graph kernel," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2990–2998.
[20] L. Bai, Z. Zhang, C. Wang, X. Bai, and E. R. Hancock, "A graph kernel based on the Jensen-Shannon representation alignment," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–5.
[21] G. Nikolentzos, P. Meladianos, S. Limnios, and M. Vazirgiannis, "A degeneracy framework for graph similarity," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2595–2601.
[22] N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Proc. Artif. Intell. Statist.*, 2009, pp. 488–495.
[23] P. Yanardag and S. V. N. Vishwanathan, "Deep graph kernels," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1365–1374.
[24] N. Kriege and P. Mutzel, "Subgraph matching kernels for attributed graphs," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 291–298.
[25] P. Wang *et al.*, "MOSS-5: A fast method of approximating counts of 5-node graphlets in large graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 1, pp. 73–86, Jan. 2018.
[26] N. M. Kriege, P. L. Giscard, and R. C. Wilson, "On valid optimal assignment kernels and applications to graph classification," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1623–1631.
[27] F. Orsini, P. Frasconi, and L. De Raedt, "Graph invariant kernels," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2015, pp. 678–689.
[28] G. Minello, L. Rossi, and A. Torsello, "Can a quantum walk tell which is which—A study of quantum walk-based graph similarity," *Entropy*, vol. 21, no. 3, p. 328, Mar. 2019.
[29] L. Bai, E. R. Hancock, A. Torsello, and L. Rossi, "A quantum Jensen-Shannon graph kernel using the continuous-time quantum walk," in *Proc. Int. Workshop Graph-Based Represent. Pattern Recognit.*, 2013, pp. 121–131.
[30] L. Rossi, A. Torsello, and E. R. Hancock, "A continuous-time quantum walk kernel for unattributed graphs," in *Proc. Int. Workshop Graph-Based Represent. Pattern Recognit.*, 2013, pp. 101–110.
[31] L. Rossi, A. Torsello, and E. R. Hancock, "Measuring graph similarity through continuous-time quantum walks and the quantum Jensen-Shannon divergence," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 91, no. 2, Feb. 2015, Art. no. 022815.
[32] L. Bai *et al.*, "Quantum kernels for unattributed graphs using discrete-time quantum walks," *Pattern Recognit. Lett.*, vol. 87, pp. 96–103, Feb. 2017.
[33] L. Bai, L. Rossi, H. Bunke, and E. R. Hancock, "Attributed graph kernels using the Jensen-Tsallis q-differences," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2014, pp. 99–114.
[34] D. Emms, R. C. Wilson, and E. R. Hancock, "Graph matching using the interference of continuous-time quantum walks," *Pattern Recognit.*, vol. 42, no. 5, pp. 985–1002, May 2009.
[35] L. Bai, Z. Zhang, P. Ren, L. Rossi, and E. R. Hancock, "An edge-based matching kernel through discrete-time quantum walks," in *Proc. Int. Conf. Image Anal. Process.*, 2015, pp. 27–38.
[36] D. Emms, S. Severini, R. C. Wilson, and E. R. Hancock, "Coined quantum walks lift the cospectrality of graphs and trees," *Pattern Recognit.*, vol. 42, no. 9, pp. 1988–2002, Sep. 2009.
[37] L. Bai, P. Ren, and E. R. Hancock, "A hypergraph kernel from isomorphism tests," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 3880–3885.
[38] P. Ren, T. Aleksiá, D. Emms, R. C. Wilson, and E. R. Hancock, "Quantum walks, Ihara zeta functions and cospectrality in regular graphs," *Quantum Inf. Process.*, vol. 10, no. 3, pp. 405–417, Jun. 2011.
[39] J. C. Ferreira and V. A. Menegatto, "Eigenvalues of integral operators defined by smooth positive definite kernels," *Integral Equ. Operator Theory*, vol. 64, no. 1, pp. 61–81, May 2009.
[40] S. V. N. Vishwanathan, K. M. Borgwardt, and N. N. Schraudolph, "Fast computation of graph kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 3880–3885.
[41] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. ICML*, 2016, pp. 2014–2023.
[42] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. AAAI*, 2018, pp. 4438–4445.
[43] K. Kersting, N. M. Kriege, C. Morris, P. Mutzel, and M. Neumann. (2016). *Benchmark data sets for graph kernels*. [Online]. Available: http://graphkernels.cs.tu-dortmund.de
[44] Y. Zhang, L. Wang, and L. Wang, "A comprehensive evaluation of graph kernels for unattributed graphs," *Entropy*, vol. 20, no. 12, p. 984, Dec. 2018.
[45] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

Yi Zhang was born in Nanyang, China, in 1987. He received the B.Eng. and Ph.D. degrees in computer science from the National University of Defense Technology, Changsha, China, in 2009 and 2014, respectively.

He is currently an Associate Professor with the National University of Defense Technology, Changsha, China, and a Technical Manager with Central Future Works Technology Co., Ltd., Beijing, China. His research interests include quantum mechanics, machine learning, and graph classification.

Lulu Wang received the B.Eng. and Ph.D. degrees in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2009 and 2015, respectively.

She is currently an Assistant Professor with Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, China. Her research interests include cognitive radar, radar waveform optimization, and machine learning.

Richard C. Wilson (Senior Member, IEEE) received the B.A. degree in physics from the University of Oxford, Oxford, U.K., in 1992, and the D.Phil. degree from the University of York, York, U.K., in 1996.

He is currently a Professor with the Department of Computer Science, University of York. He has authored more than 200 articles in journals, edited books, and refereed conferences. His research interests include structural pattern recognition, graph methods for computer vision, and novel imaging systems.

Dr. Wilson is also a fellow of the International Association for Pattern Recognition (IAPR).

Kai Liu received the B.Eng. degree in computer science from the National University of Defense Technology, Changsha, China, in 2016, where he is currently pursuing the Ph.D. degree with the College of Computer Science.

His research interests are in the areas of quantum computation and data mining.