

This is a repository copy of *An R-convolution Graph Kernel based on Fast Discrete-Time Quantum Walk*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/166053/>

Version: Accepted Version

---

**Article:**

Zhang, Yi, Wang, Lulu, Wilson, Richard Charles orcid.org/0000-0001-7265-3033 et al. (1 more author) (Accepted: 2020) An R-convolution Graph Kernel based on Fast Discrete-Time Quantum Walk. IEEE Transactions on Neural Networks and Learning Systems. ISSN 2162-237X (In Press)

---

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

# 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

This is the author accepted version of a paper to appear in IEEE Transactions on Neural Networks and Learning Systems (C) IEEE 2020

**Abstract**—In this paper, 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 multi-step 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**—graph kernel, discrete-time quantum walk, R-convolution kernel, graph classification.

## 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 with other graph comparison methods such as graph edit distance [7] or

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 [9], subtrees for the Weisfeiler-Lehman kernel [10], and so on.

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 may be expensive to capture. 3) Some similar but non-isomorphic 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 has drawn much interest from both machine learning and quantum computation communities. The reasons why the discrete-time quantum walk is suitable to connect quantum mechanics and R-convolution kernels are twofold. 1) Owing to quantum interference, the discrete-time quantum walk can achieve better discrimination for graphic data, compared with traditional methods. 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 discrete-time quantum walk, which is crucial to design R-convolution kernels. However, the simulation complexity of the discrete-time quantum walk is quadratic to that of random walk because of the approximately quadratic state space, which becomes the fatal bottleneck for graph processing.

In this paper, a fast quantum walk kernel (FQWK) is proposed based on the discrete-time quantum walk. Via computing the quantum superposition amplitudes of discrete-time quantum walks 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 discrete-time quantum walk is proved 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.

L. Wang is the corresponding author. She is with the Artificial Intelligence Research Center, National Innovation Institute of Technology, Beijing, 100071 China. e-mail: wanglulunudt@163.com.

Y. Zhang is with the School of Computer Science, National University of Defense Technology and the Central Future Works Technology Co., Ltd.

K. Liu is with the School of Computer Science, National University of Defense Technology.

R. C. Wilson is with the Department of Computer Science, University of York, UK.

Manuscript received April 19, 2005; revised August 26, 2015.

## B. Related Works

1) *R-convolution Kernels*: The concept of R-convolution kernel was firstly proposed in [14]. A graph is firstly decomposed into some certain substructures, e.g. walks or subtrees, 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  $G_B$ ,  $\{\mathcal{G}_{A;1}, \dots, \mathcal{G}_{A;N_A}\}$  and  $\{\mathcal{G}_{B;1}, \dots, \mathcal{G}_{B;N_B}\}$  are the substructure sets of  $G_A$  and  $G_B$ , respectively.  $\mathcal{G}_{i;j}$  is the  $j^{\text{th}}$  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.

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

where  $\Delta$  denotes a Dirac kernel shown as follows,

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

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.

**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 track back 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 via 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 via counting all the sub-path similarities. Meanwhile, backtrackless walk kernel (BWK) was proposed in [12] using the zeta function which 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 random walk to improve the RWK, so that various node attributes can be effectively exploited.

**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 isomorphism subtree structures. Owing 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 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 datasets with typical tree structures, such as molecular structures.

**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 walk-based and subtree-based kernel can only capture limited structural features of graphs.

One of the classic subgraph-based kernel is the all graphlet kernel (AGK) [22]. A graphlet is a subgraph with only 3 to 5 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 show 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. L. Bai et al [33] improved the classification accuracy by modifying the kernels using 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 use 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 [35] and [11] respectively. The essence of these methods is using 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 paper are listed as follows.

- A novel R-convolution graph kernel FQWK is proposed based on the fast discrete-time quantum walk. The FQWK characterizes the neighborhood-pair substructure between two nodes via the superposition amplitude of all quantum walks of a particular length which join 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 datasets.
- To achieve the neighborhood-pair substructure matching effectively, a fast simulation method is proposed for discrete-time quantum walk. 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 multi-step discrete-time quantum walk. 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.
- The high dimensional structural information of graphs can be explored via quantum mechanism so that the slight structural differences between some similar but non-isomorphic 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 The Paper

The remainder of the paper is structured as follows. Firstly, the background of graph concept and discrete-time quantum walk is introduced in Section II. And then, in Section III, the R-convolution kernel based on fast discrete-time quantum walk 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.

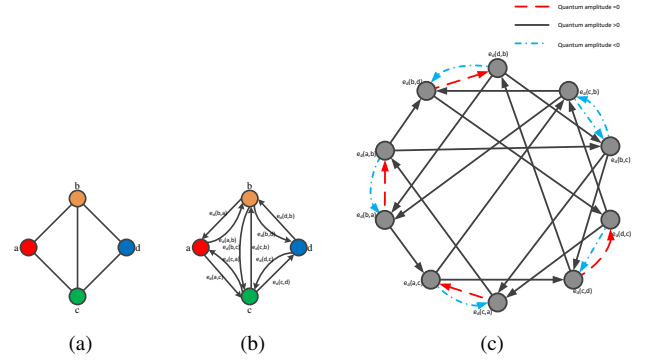


Fig. 1. (a) A demo graph with 4 nodes; (b) The demo graph in (a) whose edges are all replaced by the directed edge set  $E_d$ ; (c) The directed line graph  $G_L(V_L, E_L)$ . Note that discrete-time quantum walk based on the Grover diffusion matrix can be regarded as a walk on  $G_L$ . The red dashed arcs, black solid arcs, and blue dash-dot arcs represent that the quantum transition amplitudes are 0, positive and negative, respectively.

**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 paper, 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  $w = (v_0, v_1, \dots, v_k)$  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  $k$  which 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  $f : V \rightarrow V'$  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 discrete-time quantum walk (DTQW) is the quantum counterpart of the discrete-time classical random walk [36]. In discrete-time quantum walk, 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 discrete-time quantum walk 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.

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

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. And the node set and edge set are defined as follows,

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

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 discrete-time quantum walk, 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 with the classical random walk, the discrete-time quantum walk 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 discrete-time quantum walk is:

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

where the quantum amplitude  $\alpha_{uv}$  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  $\alpha_{uv}^*$  is the complex conjugate of  $\alpha_{uv}$ .

At each step, the evolution of the discrete-time quantum walk 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:

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

Here  $U_{im;nj}$  shows the quantum amplitude for the transition  $e_d(i, m) \rightarrow e_d(n, j)$ ,  $d_m$  denotes the node degree for node  $m$  and  $\delta_{ij}$  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 discrete-time quantum walk

is quantum amplitude. Given a state  $|im\rangle$ , the Grover matrix assigns the same amplitude to all transitions  $|im\rangle \rightarrow |mj\rangle$ , and a different amplitude to the transition  $|im\rangle \rightarrow |mi\rangle$ , 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 dash-dot 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 random walk will be settled via the amplitude penalties for back tracks.

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 discrete-time quantum walk 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 discrete-time quantum walk 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 than 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 paper, 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  $S_{ab}^{(k)}$ .** In a graph, for each node pair  $a$  and  $b$ , the  $k$ -level neighborhood-pair substructure  $S_{ab}^{(k)}$  is constructed which contains all the  $k$ -length walks between  $a$  and  $b$ .

$$S_{ab}^{(k)} = \{w \in W^{(k)} \mid v_0 = a, v_k = b\} \quad (7)$$

Fig. 2(a) illustrates that in the demo graph in Fig. 1(a), there are totally 5 3-length walks from node  $a$  to node  $b$ , which make up the auxiliary substructure  $S_{ab}^{(3)}$ . All the walk-based kernels can be regarded as counting over the auxiliary substructure  $S_{ab}^{(k)}$ . For example, there are 5 walks contributing to the RWK. The SPK and GHK compare only one walk which is the non-repeating path ( $acdb$ ). Similarly, the BWK counts the single



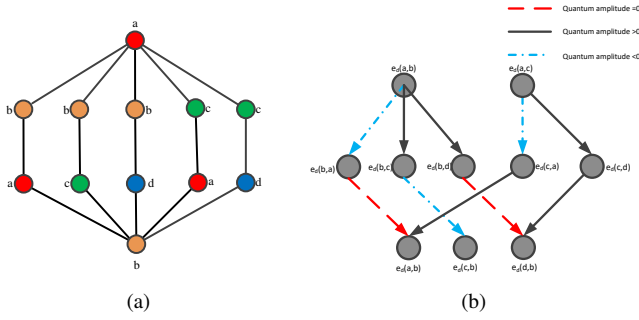


Fig. 2. (a) The 5 3-length walks from node  $a$  to node  $b$ ; (b) The corresponding 5 2-step quantum walks on the directed line graph  $G_L$ .

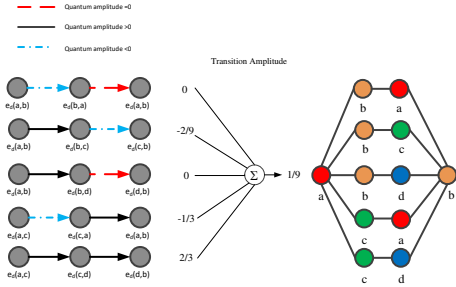


Fig. 3. Quantum transition amplitudes of the 5 3-length walks and the neighborhood-pair substructure  $S_{ab}^{(3)}$ .

non-backtracking walk ( $acdb$ ). However, the BWK, GHK and SPK do not account for the edges  $(ab)$  and  $(bc)$  and the RWK treats all 5 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 3-length walks correspond to the 5 2-step discrete-time quantum walks 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 3-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:

$$M_{ab} = \sum_{m=1}^N \sum_{n=1}^N U_{am;nb} \quad (8)$$

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

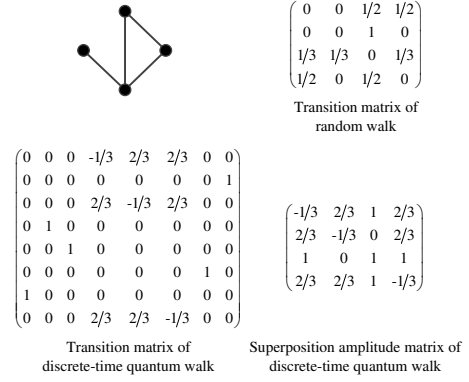


Fig. 4. A demo graph and the comparisons of its three transition matrices.

amplitudes over all possible first and last steps:

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

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  $t^{th}$  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. Firstly, it is more compact than the DTQW, since it is only of size  $|V| \times |V|$ . Secondly 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 discrete-time quantum walk, the superposition amplitude matrix  $M^{(t)}$  is computed as,

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

Obviously, 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  $O(|V|^6)$ , which is potentially quadratic to the transition

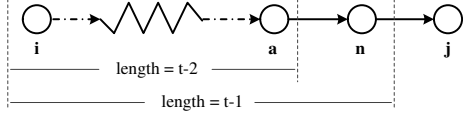


Fig. 5. A  $t$ -step quantum walk from node  $i$  to node  $j$ . Node  $n$  is the previous step of node  $j$ . Node  $a$  is the previous step of node  $n$ .

matrix of random walk. Therefore, it is not time-acceptable to perform graph processing directly.

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

Therefore, we need to find a fast method to simulate the discrete-time quantum walk and compute the matrix  $M^{(t)}$ . In Fig.5, an arbitrary  $t$ -step discrete-time quantum walk from node  $i$  to node  $j$  is shown as an example. This walk is consisted of a  $(t-1)$ -step quantum walk from  $i$  to  $n$  and a 1-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 discrete-time quantum walk.

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

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

*Proof.* (1) When  $t = 1$ , for every item in matrix  $M$ ,

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

(2) When  $t = 2$ ,

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

(3) When  $t \geq 3$ ,

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

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

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

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

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 discrete-time quantum walk, named FQWK.

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

$$\mathcal{K}_{FQWK}(G_A, G_B) = \sum_t \mathcal{K}_t(G_A, G_B). \quad (16)$$

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

$$\mathcal{K}_t(G_A, G_B) = \sum_{m,n \in G_A} \sum_{u,v \in G_B} \Delta(S_{mn}^{(t)}, S_{uv}^{(t)}), \quad (17)$$

where  $\Delta$  is a Dirac function as follows:

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

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 dataset:* In the above definition, the computation of the simulation of the DTQW is the most time-consuming procedure. Therefore for a graph dataset with many graphs, this procedure needs to be performed only once on each graph actually, before the pairwise kernel computation.

Suppose that a graph dataset includes  $K$  graphs and each one has  $N$  unattributed nodes.

Firstly, a  $T$ -step discrete-time quantum walk 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  $i^{th}$  matrices  $M^{(i)}$  of every graph  $G$ , the histogram of all the  $N^2$  items is constructed as the  $i^{th}$  feature  $F_G^{(i)}$  of graph  $G$ . Therefore, for each graph, a  $T$ -dimension feature vector can be extracted. The pseudo code for the implementation of graph feature extraction algorithm is shown in **Algorithm 1**.

And then, for each graph pair  $G_A$  and  $G_B$  in the dataset, 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 dataset, 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 1 Graph Feature Extraction Algorithm

---

**Require:** Graph  $G$ , with  $N$  nodes;  $d_i, i = 1, \dots, N$  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  $G$ ,  $t = 1, \dots, T$ .

```

1: function GraphFeature( $G, T$ )
2:   Get the adjacency matrix  $A$ 
3:    $D = \text{diag}\left(\frac{2}{d_1}, \frac{2}{d_2}, \dots, \frac{2}{d_N}\right)$ 
4:    $M \leftarrow ADA - 2D^{-1}$ 
5:    $F_G^{(1)} \leftarrow \text{histogram}(M)$ 
6:    $M^{(2)} \leftarrow MDA - A$ 
7:    $F_G^{(2)} \leftarrow \text{histogram}(M^{(2)})$ 
8:   for  $t = 3 \rightarrow T$  do
9:      $M^{(t)} \leftarrow M^{(t-1)}DA - M^{(t-2)}$ 
10:     $F_G^{(t)} \leftarrow \text{histogram}(M^{(t)})$ 
11:   end for
12: end function
```

---



---

### Algorithm 2 FQWK Algorithm

---

**Require:** Graph dataset  $\{G_1, G_2, \dots, G_K\}$ , 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:    $F_{G_i}^{(t)} \leftarrow \text{GraphFeature}(G_i, T)$ ,  $t = 1, \dots, T$ 
3: end for
4: for each graph pair  $G_i$  and  $G_j$  do
5:   for  $t = 1 \rightarrow T$  do
6:      $\mathbf{f}_{G_i}^{(t)}, \mathbf{f}_{G_j}^{(t)} = \text{alignment}(F_{G_i}^{(t)}, F_{G_j}^{(t)})$ 
7:      $\mathcal{K}_t(G_i, G_j) \leftarrow \langle \mathbf{f}_{G_i}^{(t)}, \mathbf{f}_{G_j}^{(t)} \rangle$ 
8:   end for
9:    $\mathcal{K}_{FQWK}(G_i, G_j) = \sum_{t=1}^T \mathcal{K}_t(G_i, G_j)$ 
10: end for
```

---

Finally, we will evaluate the time complexity of calculating the kernel matrix. For each graph  $G$  of the dataset, a  $T$ -step discrete-time quantum walk needs to be performed firstly to extract the  $T$ -dimension feature vector  $F_G$ . It will cost no more than  $\mathcal{O}(KTN^3 + KTN^2 \log N)$  for line 1-3 in **Algorithm 2**. And in line 4-9, to compute the kernel matrix, for all the graph pairs, the neighborhood-pair matching procedures are performed which cost about  $\mathcal{O}(K^2TN^2)$ . Overall, the total time complexity of N<sup>2</sup>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 semi-definite (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.

- Symmetric.

Based on the definition of FQWK in (16), it is obvious that  $\mathcal{K}_{FQWK}(G_A, G_B) = \mathcal{K}_{FQWK}(G_B, G_A)$ .



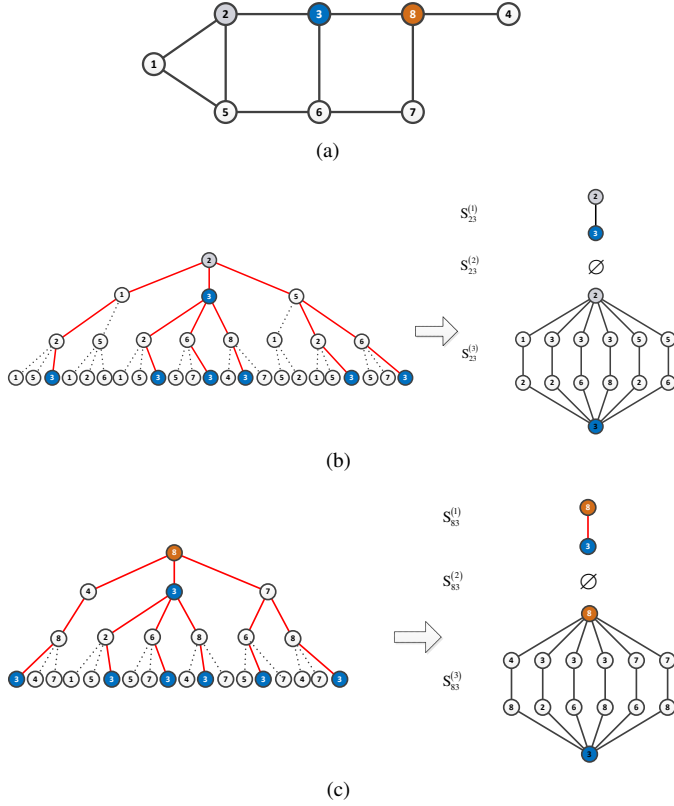


Fig. 6. (a) An example graph with node 8; (b) The 3-level neighborhood of node  $v_2$  and the substructures  $S_{23}^{(1)}$ ,  $S_{23}^{(2)}$  and  $S_{23}^{(3)}$ ; (c) The neighborhood of  $v_8$  and the substructures  $S_{83}^{(1)}$ ,  $S_{83}^{(2)}$  and  $S_{83}^{(3)}$ .

- p.s.d.

In the definition of FQWK, after the  $t^{th}$  step of quantum walk, a matching sub-kernel  $\mathcal{K}_t(G_A, G_B)$  will be performed to count all the  $t^{th}$ -level isomorphic neighborhood-pair substructures. It is known that the summation kernel of some p.s.d. ones is still p.s.d. Because Dirac function is p.s.d, the sub-kernel  $\mathcal{K}_t(G_A, G_B)$  is p.s.d. Therefore, FQWK is p.s.d.  $\square$

2) *Theoretical analysis of FQWK*: From the above definition, FQWK seems an instance of walk-based graph kernels. It is close to RWK and BWK because we only use discrete-time quantum walk 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) shows an example graph with 8 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  $v_8$  as well as the 1 to 3 level neighborhood-pair substructures are shown in Fig.6(b) and Fig.6(c), respectively. Obviously, both  $S_{23}^{(1)}$  and  $S_{83}^{(1)}$  have only 1 walk (path).  $S_{23}^{(2)}$  and  $S_{83}^{(2)}$  are both empty. Meanwhile there are 6 3-length walks included in  $S_{23}^{(3)}$  and

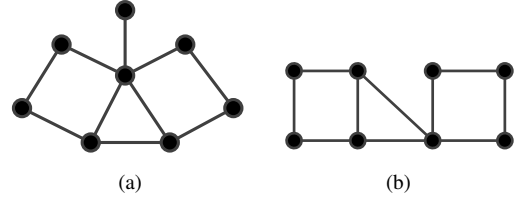


Fig. 7. Two example graphs with node 8.

$S_{83}^{(3)}$ , and only 1 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 kind of structures exist. If the proposed FQWK is used, the superposition amplitude of the substructure  $S_{v_2v_3}^{(3)}$  is  $-1/9$ , which is different with the substructure  $S_{v_8v_3}^{(3)}$  with amplitude 1, so the two substructures don't 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.

Actually, 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 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, 1 triangle and 2 squares are included in all the three example graphs, but the relative position among these 3 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 are 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:

- The inter interference among all the quantum walks which are included in the neighborhood-pair substructure. In Fig.6(b) and Fig.6(c), the red solid lines denote these included quantum walks. The quantum interference among them will affect the superposition amplitude of the substructure  $S_{ab}^{(k)}$ .
- The intra interference from the adjacent neighborhood-pair substructures. In Fig.6(b) and Fig.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 of the location relationships of local substructures.

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

#### IV. EXPERIMENTS

In this section, the newly proposed kernel FQWK is evaluated on classification problems for unattributed graphs. Here we perform a fast discrete-time quantum walk 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. And we also compare FQWK with several other popular graph classification methods as follows.

- Random walk kernel (RWK). The version of [40] is used to test.
- Weisfeiler-Lehman kernel (WLK) [10]. For the sake of fairness, the highest dimension of the Weisfeiler-Lehman isomorphism is set to be 10.
- GraphHopper kernel (GHK) [17]. The Dirac kernel is used to be the base node kernel.
- All graphlet kernel (AGK) [22]. The graphlet size is chosen to be 4.
- Aligned subtree kernel (ASK) [11]. The density matrix of continuous-time quantum walk is used to enhance the node attributes. And 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.
- Quantum Jensen-Shannon kernel (QJSK) [32]. The density matrix of 10-step discrete-time quantum walk is used for the computation of Von Neumann entropy.
- Edge-based matching kernel based on discrete-time quantum walk (DQMK) [35]. A 10-step discrete-time quantum walk is used to perform edge matching and the highest layer of the depth-based representation is set to be 10.
- PATCHY-SAN convolutional neural network(PSCN) [41]. Similar with 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 use the graph canonization tool NAUTY to apply CNNs on these patches. Here we set the receptive field size as 10.
- 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 learning rate as 0.0001. Both PSCN and DGCNN are trained with epochs 150 and batch size 25.

TABLE I  
THE DETAILED INFORMATION OF THE REAL-WORLD DATASETS

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 NON-ISOMORPHIC GRAPH DATASETS. IN COSGRAPH, EVERY COSPECTRAL GRAPH PAIR IS USED AS A TEST. IN REGGRAPH AND SRGRAPH, PAIR-WISE 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

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. Datasets

Both real-world and synthetic datasets are used to evaluate the graph kernels.

**The real-world datasets.** To evaluate the classification accuracy, 7 chemical datasets 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 2,000 elements totally (1,600 inactive elements and 400 active elements), which represent molecules with activity against HIV or not. COX2 has 467 cyclooxygenase-2 inhibitors has been assembled in this dataset. DHFR is consisted 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 6 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 non-small cell lung cancer cell lines. The Predictive Toxicology Challenge (PTC) dataset 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 shows the statistical information of these chemical datasets.

**The synthetic datasets.** In order to further evaluate the distinguishing ability of the kernels, some special datasets are chosen [44] as shown in Table II. CosGraph includes

TABLE III  
THE AVERAGE ACCURACY (IN %  $\pm$  STANDARD ERROR) ON GRAPH CLASSIFICATION BENCHMARK DATASETS

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

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)

Datasets	AIDS	COX2	DHFR	ENZYMES	MUTAG	NCI1	PTC_MM
Methods							
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"
<b>FQWK</b>	38'52"	8'47"	13'57"	5'15"	27"	269'50"	58"

5048 pairs of 10-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 datasets in Table I.

For each dataset, 10-fold 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 Table III and Table IV.

Table III reports the average classification accuracy and the relative standard error of each method and dataset. For the datasets 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 slight better performance than FQWK respectively. Although the classification for unattributed graphs is quite difficult, FQWK turns out to be

the best competitor in terms of accuracy on most of these benchmark datasets.

The reasons for the effectiveness are threefold.

- a) Owing 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.
- b) Compared with the information theoretic kernel QJSK which 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 discrete-time quantum walk.
- c) 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 dataset. FQWK achieves a moderate performance. However, compared with ASK, QJSK and DQMK, FQWK is the fastest one among all the kernels based on 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 non-isomorphic 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

TABLE V

THE FAILURE RATES (%) FOR DISTINGUISHING THE NON-ISOMORPHIC GRAPHS (- DENOTES THE TEST CANNOT BE FINISHED BY THE KERNEL IN 10 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	-
<b>FQWK</b>	<b>0</b>	<b>0.02</b>	<b>0.0016</b>

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.

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 non-isomorphic graphs, and thus outperforms the other kernels. The powerful ability of FQWK for distinguishing non-isomorphic graphs, even including strong regular graphs, comes from the fact that both the inter and intra quantum interference of the neighborhood-pair substructures is 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 is 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  $T$  for the datasets COX2 and PTC\_MM. We change  $T$  from 5 to 25. The accuracy and computation time results are shown in Fig. 8 and Fig. 9, respectively. We find out that the accuracy keeps almost unchanged. However, the computation time has a nearly linear growth with the increase of  $T$ . 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 increasing of  $T$ , 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 which extract features via node neighborhoods.

## V. CONCLUSION

In this paper, a novel R-convolution graph kernel FQWK is proposed based on the fast discrete-time quantum walk. Via the

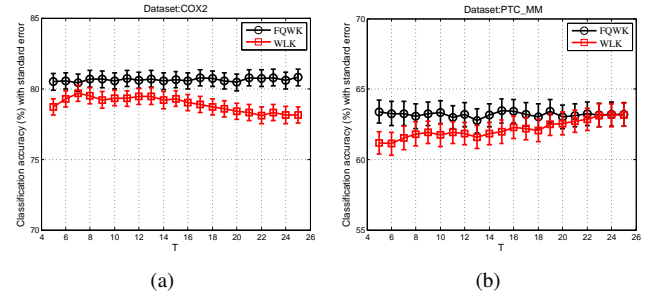


Fig. 8. (a) The classification accuracy (%) and standard error of using FQWK and WLK for the dataset COX2 with different  $T$ ; (b) The classification accuracy (%) and standard error of using FQWK and WLK for the dataset PTC\_MM with different  $T$ .

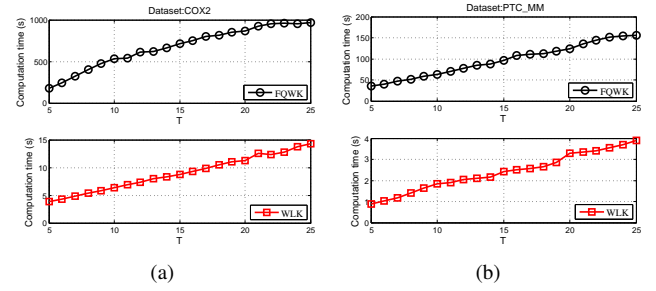


Fig. 9. (a) The computation time (s) of using FQWK and WLK for the dataset COX2 with different  $T$ ; (b) The computation time(s) of using FQWK and WLK for the dataset PTC\_MM with different  $T$ .

powerful quantum interference of the discrete-time quantum walk, 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 the classification accuracy of FQWK outperforms that of the state-of-the-art graph kernels and the distinguishing ability for non-isomorphic graphs is significantly improved.

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

## ACKNOWLEDGMENT

This work is funded by Natural Science Foundation of Hunan Province, grant number 2019JJ40340, and National Natural Science Foundation of China, grant number 61701502.

The authors appreciate the kind comments and professional criticisms of the anonymous reviewers. They have greatly enhanced the overall quality of the manuscript and opened numerous perspectives geared toward improving the work.

## REFERENCES

- [1] V. T. Ta, O. L  zoray, A. Elmoataz, and S. Sch  upp, "Graph-based tools for microscopic cellular image segmentation," *Pattern Recognition*, 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 *International Conference on Database Theory*, 2012.
- [4] A. C. N. Ngomo and F. Schumacher, "BorderFlow: A local graph clustering algorithm for natural language processing," in *International Conference on Computational Linguistics and Intelligent Text Processing*, 2009.
- [5] P. Mahé and J. P. Vert, "Graph kernels based on tree patterns for molecules," *Machine Learning*, 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, 2017.
- [7] H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 689–694, 1997.
- [8] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, 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 and kernel machines*. Springer, 2003, pp. 129–143.
- [10] N. Shervashidze, P. Schweitzer, E. J. V. Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman graph kernels," *Journal of Machine Learning Research*, vol. 12, no. 9, pp. 2539–2561, 2011.
- [11] L. Bai, L. Rossi, and E. Hancock, "An aligned subtree kernel for weighted graphs," in *International Conference on Machine Learning*, 2015, pp. 30–39.
- [12] F. Aziz, R. C. Wilson, and E. R. Hancock, "Backtrackless walks on a graph," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 6, pp. 977–989, 2013.
- [13] Y. Aharonov, L. Davidovich, and N. Zagury, "Quantum random walks," *Physical Review A*, vol. 48, no. 2, p. 1687, 1993.
- [14] D. Haussler, "Convolution kernels on discrete structures," Technical report, Department of Computer Science, University of California, Tech. Rep., 1999.
- [15] L. Bai, "Information theoretic graph kernels," Ph.D. dissertation, University of York, 2014.
- [16] K. M. Borgwardt and H. P. Kriegl, "Shortest-path kernels on graphs," in *International Conference on Data Mining*, 2005, p. 8.
- [17] A. Feragen, N. Kasenburg, J. Petersen, M. de Bruijne, and K. Borgwardt, "Scalable kernels for graphs with continuous attributes," in *Annual Conference on Neural Information Processing Systems*, 2013, 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 *Annual Conference on Neural Information Processing Systems*, 2018.
- [19] R. Kondor and H. Pan, "The multiscale Laplacian graph kernel," in *Annual Conference on Neural Information Processing Systems*, 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 *International Joint Conference on Artificial Intelligence*, 2015.
- [21] G. Nikolentzos, P. Meladinos, S. Linnios, and M. Vazirgiannis, "A degeneracy framework for graph similarity," in *International Joint Conference on Artificial Intelligence*, 2018.
- [22] N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Artificial Intelligence and Statistics*, 2009, pp. 488–495.
- [23] P. Yanardag and S. V. N. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1365–1374.
- [24] N. Kriege and P. Mutzel, "Subgraph matching kernels for attributed graphs," in *International Conference on Machine Learning*, 2012, pp. 291–298.
- [25] P. Wang, J. Zhao, X. Zhang, Z. Li, J. Cheng, J. C. S. Lui, D. Towsley, J. Tao, and X. Guan, "MOSS-5: A fast method of approximating counts of 5-node graphlets in large graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 1, pp. 73–86, 2018.
- [26] N. M. Kriege, P. L. Giscard, and R. C. Wilson, "On valid optimal assignment kernels and applications to graph classification," in *Annual Conference on Neural Information Processing Systems*, 2016.
- [27] F. Orsini, P. Frasconi, and L. De Raedt, "Graph invariant kernels," in *Proceedings of the AAAI Conference on Artificial Intelligence (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, 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 *International Workshop on Graph-Based Representations in Pattern Recognition*, 2013, pp. 121–131.
- [30] L. Rossi, A. Torsello, and E. R. Hancock, "A continuous-time quantum walk kernel for unattributed graphs," in *International Workshop on Graph-Based Representations in Pattern Recognition*, 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 Nonlin Soft Matter Phys*, vol. 91, no. 2, p. 022815, 2015.
- [32] L. Bai, L. Rossi, L. Cui, Z. Zhang, P. Ren, X. Bai, and E. Hancock, "Quantum kernels for unattributed graphs using discrete-time quantum walks," *Pattern Recognition Letters*, vol. 87, pp. 96–103, 2017.
- [33] L. Bai, L. Rossi, H. Bunke, and E. R. Hancock, "Attributed graph kernels using the Jensen-Tsallis q-differences," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 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 Recognition*, vol. 42, no. 5, pp. 985–1002, 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 *International Conference on Image Analysis and Processing*. Springer, 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 Recognition*, vol. 42, no. 9, pp. 1988–2002, 2009.
- [37] L. Bai, P. Ren, and E. R. Hancock, "A hypergraph kernel from isomorphism tests," in *International Conference on Pattern Recognition*, 2014.
- [38] R. Peng, T. Aleksic, D. Emms, R. C. Wilson, and E. R. Hancock, "Quantum walks, Ihara zeta functions and cospectrality in regular graphs," *Quantum Information Processing*, vol. 10, no. 3, pp. 405–417, 2011.
- [39] J. C. Ferreira and V. A. Menegatto, "Eigenvalues of integral operators defined by smooth positive definite kernels," *Archiv Der Mathematik*, vol. 64, no. 1, pp. 61–81, 2009.
- [40] S. V. N. Vishwanathan, K. M. Borgwardt, and N. N. Schraudolph, "Fast computation of graph kernels," in *Advances in Neural Information Processing Systems 19*, 2007.
- [41] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *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 *AAAI*, 2018, pp. 4438–4445.
- [43] K. Kersting, N. M. Kriege, C. Morris, P. Mutzel, and M. Neumann, "Benchmark data sets for graph kernels," 2016. [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, 2018.
- [45] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM transactions on intelligent systems and technology*, vol. 2, no. 3, p. 27, 2011.



**Yi Zhang** was born in Nanyang, China, in 1987. He received the B.Eng. and the Ph.D. degree in Computer Science from National University of Defense Technology in 2009 and 2014, respectively. His research interests include quantum mechanics, machine learning and graph classification.





**Lulu Wang** received the B.Eng. and the Ph.D. degrees in Information and Communication Engineering from 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 Wilson** 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 papers in journals, edited books, and refereed conferences. His research interests include structural pattern recognition, graph methods for computer vision and novel imaging systems. He is a fellow of the IAPR and a senior member of IEEE.



**Kai Liu** received the B.Eng. degree in Computer Science from National University of Defense Technology in 2016. He is currently a doctoral student in College of Computer Science, National University of Defense Technology, Changsha, China. His research interests are in the areas of quantum computation and data mining.