This is a repository copy of *A Quantum-inspired Similarity Measure for the Analysis of Complete Weighted Graphs*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/145306/

Version: Accepted Version

Article:
Bai, Lu, Rossi, Luca, Cui, Lixin et al. (2 more authors) (2019) A Quantum-inspired Similarity Measure for the Analysis of Complete Weighted Graphs. IEEE Transactions on Cybernetics. pp. 1264-1277. ISSN: 2168-2267

https://doi.org/10.1109/TCYB.2019.2913038

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 including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

A Quantum-inspired Similarity Measure for the Analysis of Complete Weighted Graphs

Lu Bai, Luca Rossi, Lixin Cui, Jian Cheng, Edwin R. Hancock, *IEEE Fellow*

***Abstract*—We develop a novel method for measuring the similarity between complete weighted graphs, which are probed by means of discrete-time quantum walks. Directly probing complete graphs using discrete-time quantum walks is intractable due to the cost of simulating the quantum walk. We overcome this problem by extracting a commute-time minimum spanning tree from the complete weighted graph. The spanning tree is probed by a discrete time quantum walk which is initialised using a weighted version of the Perron-Frobenius operator. This naturally encapsulates the edge weight information for the spanning tree extracted from the original graph. For each pair of complete weighted graphs to be compared, we simulate a discrete-time quantum walk on each of the corresponding commute time minimum spanning trees, and then compute the associated density matrices for the quantum walks. The probability of the walk visiting each edge of the spanning tree is given by the diagonal elements of the density matrices. The similarity between each pair of graphs is then computed using either a) the inner product or b) the negative exponential of the Jensen-Shannon divergence between the probability distributions. We show that in both cases the resulting similarity measure is positive definite and therefore corresponds to a kernel on the graphs. We perform a series of experiments on publicly available graph datasets from a variety of different domains, together with time-varying financial networks extracted from data for the New York Stock Exchange. Our experiments demonstrate the effectiveness of the proposed similarity measures.**

***Index Terms*—Graph Similarity, Graph Kernels, Quantum Walks, Jensen-Shannon Divergence, Financial Networks.**

I. Introduction

Graph-based representations commonly arise in a wide variety of systems that are naturally described in terms of relations between their components. For instance, Wu et al. [1] have represented the texts inside a webpage as graphs, with vertices representing words and edges denoting relations between words. Li et al. [2] have represented each video frame as a graph structure with vertices representing superpixels and edges denoting relations between superpixels. Tang et al. [3] have looked at the local spectral descriptors of each image as points and constructed the weighted graph based on neighborhood relations. Other typical examples include representing a chemical molecular structure or a document as graphs [4]. One main problem of analyzing these structures is that of measuring the similarity between two graphs for classification or clustering [5], [6]. For example, in network science a common objective is to detect the extreme events that can significantly change the time-varying network structures [7], [8], [9], [10], [11], [12], [13] abstracted from vectorial time series [14]. Given a measure of similarity between graphs, one can then identify extreme events by looking at significant changes in the underlying network structures.

Consider a system that can be represented by a series of complete weighted graphs, where the number of vertices N is fixed but the edge weights change over time. Such a representation is a natural choice for those systems where we are interested in modelling the strength of the edges rather than their presence or absence. With this time-varying network to hand, one can look at how it evolves in order to detect anomalies in the system. An example of this type of graph is furnished by financial networks, where each vertex represents a stock and each edge weight measures the association between the time series of the corresponding stock, in terms of correlation [15], Granger causality [16] or transfer entropy [17]. In this domain, extreme events representing financial instability of different stock are of interest [18] and can be inferred by detecting the anomalies in the corresponding networks [15].

Existing methods aim to derive network characteristics based on connectivity structure, or statistics capturing connectivity structure [19], [20], [21]. These methods focus on capturing network substructures using clusters, hubs and communities. Moreover, an alternative principled approach is to characterize the networks using ideas from statistical physics [22], [23]. These methods use the partition function to describe the network, and the associated entropy, energy, and temperature measures can be computed through this function [15], [24], [25]. Unfortunately, the aforementioned methods tend to characterize network structures in a low dimensional pattern or vector space, and thus discard a lot of structural information. This drawback influences the effectiveness of existing approaches for time-varying network analysis. The aim of this paper is to address this shortcoming of existing methods by means of graph kernels.

A. Graph Kernels

In machine learning, graph kernels are important tools for analyzing structured data that are represented by graphs. This is because graph kernels not only allow us to map graph structures into a high dimensional space, but also provide a way of making the rapidly developing kernel machinery for vectorial data applicable to graphs. In essence, graph

Lu Bai is with Central University of Finance and Economics, Beijing, China. e-mail: bailucs@cufe.edu.cn; bailu69@hotmail.com.
Luca Rossi is with Southern University of Science and Technology, Guangdong, China. email: rossil@sustech.edu.cn (Co-Corresponding Author)
Lixin Cui is with Central University of Finance and Economics, Beijing, China. e-mail: cuilixin@cufe.edu.cn (Corresponding Author)
Jian Cheng is with National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
Edwin R. Hancock is with University of York, York, UK.

kernels are positive definite similarity measures between pairs of graphs [5], [26], [27], [28], [29], [30], [31], [32]. They allow rapidly developing kernelized algorithms (e.g., Support Vector Machines, kernel Principle Component Analysis, etc) for vectorial data associating with vectorial kernels [33], [34] directly applicable to graph structures.

One leading principle for defining kernels between a pair of graphs is to decompose the graphs into substructures and to measure the similarity between the input graphs by enumerating pairs of isomorphic substructures. Specifically, any available graph decomposition method can be adapted to develop a graph kernel, e.g., graph kernels based on counting pairs of isomorphic a) paths [26], b) walks [27], [28], and c) subgraphs or subtrees [29], [30], [31], [32]. Unfortunately, there are two common shortcomings arising with these graph kernels. First, they do not work well on complete weighted graphs, where each pair of vertices is linked by a weighted edge. This is due to the trivial structure of the complete graph, i.e., each vertex is adjacent to all the other vertices, whereas the weights may be quite different. Thus, it is difficult to decompose a complete weighted graph into substructures. On the other hand, identifying the isomorphism between weighted (sub)graphs tends to be computationally burdensome unless the weight information is discarded. Second, graph kernels cannot scale up to large structures. To overcome this issue, existing graph kernels usually compromise and use small sized substructures. However, measuring kernel values with small substructures only partially reflects the topological characteristics of a graph.

To address the restriction of R-convolution graph kernels to complete weighted graphs, a number of graph kernels [35], [36], [37] based on using the adjacency matrix to capture global graph characteristics have been developed. Since the adjacency matrix directly reflects the edge weight information, these kernels can naturally accommodate complete weighted graphs. For instance, Johansson [35] et al. have developed a family of global graph kernels based on the Lovasz number and its associated orthonormal representation through the adjacency matrix. Xu et al. [36] have proposed a local-global mixed reproducing kernel based on the approximated von Neumann entropy through the adjacency matrix. Bai et al. [37] have defined an information theoretic kernel based on the classical Jensen-Shannon divergence between the steady state random walk probability distributions obtained through the adjacency matrix. Recently, there has been increasing interests in continuous-time quantum walks [38] for the analysis of global graph structures. The continuous-time quantum walk is the quantum analogue of the classical continuous-time random walk. Unlike the classical random walk that is governed by a doubly stochastic matrix, the quantum walk is governed by a unitary matrix and is not dominated by the low frequencies of the Laplacian spectrum. Thus, the continuous-time quantum walk is able to better discriminate different graph structures.

There have been a number of graph kernels developed using the continuous-time quantum walk. For instance, Bai et al. [39] have developed a quantum kernel by measuring the similarity between two continuous-time quantum walks evolving on a pair of graphs. Specifically, they associate each graph with a mixed quantum state that represents the evolution of the quantum walk. The resulting kernel is computed by measuring the quantum Jensen-Shannon divergence between the associated density matrices. Rossi et al. [40] have developed a quantum kernel by exploiting the relation between the continuous-time quantum walk intereferences and the symmetries of a pair of graphs, in terms of the quantum Jensen-Shannon divergence. Both of these quantum kernels employ the Laplacian matrix as the required Hamiltonian operator, and thus they can naturally accommodate complete weighted graphs. Unfortunately, all of the aforementioned kernels, either the classical kernels defined through the adjacency matrix or the quantum kernels based on the Laplacian matrix as the Hamiltonian operator, are restricted to un-attributed graphs. Furthermore, both quantum kernels require a composite structure of each pair of graphs to compute an additional mixed state that describes a system having equal probability of being in the two original quantum states. Unless, we can make use of the transitive alignment information between the vertices of the two graphs, then neither of the quantum kernels are positive definite.

B. Contributions

In this paper, we aim to address the above mentioned drawbacks of existing state-of-the-art graph kernels by proposing a new kernel for complete weighted graphs. We test this similarity measure on graphs extracted from financial time series, as these are typically abstracted using sets of complete weighted graphs. We stress, however, that our kernel can also be applied to general graphs. In Section V we measure the performance of the new kernel on a series of graph embedding and classification tasks, showing that it significantly outperforms a number of widely used alternative kernels.

Given a set of graphs, our aim is to probe each of them using a discrete-time quantum walk [41]. The reasons for using a walk of this type are twofold. First, quantum walks possess several exotic properties not exhibited by their classical counterpart. These in turn are a consequence of the marked differences between the two types of walks. For instance, unlike the classical walks the evolution of which is governed by a stochastic matrix, the behaviour of quantum walks is governed by a unitary matrix. As a result of their unique properties, during their evolution, quantum walks produce destructive interference patterns which have been shown to lead to more powerful graph characterisations [41]. Second, unlike continuous-time quantum walks [42], the state space of a discrete-time quantum walk is the set of directed edges rather than the set of vertices. In particular, suppose G is a sample graph with vertex set V and edge set E. We replace each edge $\{u, v\} \in E$ with a pair of directed edges $e_d(u, v)$ and $e_d(v, u)$, and we denote this set as E_d. Since E_d is the state space of the discrete-time quantum walk and its cardinality is much larger than, or at least equal to, that of the vertex set V, the discrete-time quantum walk can capture the structural characteristics of the graph better than its continuous-time counter part.

Unfortunately, directly simulating the discrete-time quantum walk on a complete weighted graph tends to be a challenging problem. The time complexity of simulating the evolution of

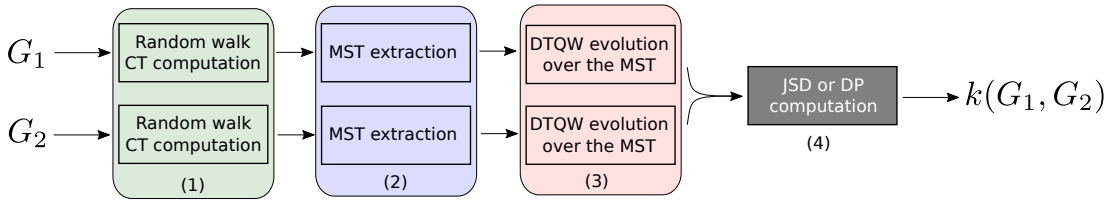


Fig. 1. The proposed framework to compute the similarity between two complete weighted graphs. Given two input graphs, for each of them (1) we compute the random walk commute time matrix, and (2) we extract the corresponding commute time spanning tree. (3) We probe the structure of each tree using discrete-time quantum walks and computing the time-average probability of visiting each arc residing on the edge. Finally, (4) the kernel between the original input graphs is defined as the similarity between the corresponding time-averaged probability distributions.

a discrete-time quantum walk is cubic in the size of the state space. For a complete weighted graph having n vertices, there are $n(n-1)$ directed edges. Thus, performing the discrete-time quantum walk on a complete weighted graph has an associated time complexity of $O(n^6)$. Consequently, the use of discrete-time quantum walks for the analysis of complete weighted graphs is computationally expensive and unadvisable. Most importantly, two complete weighted graphs over the same set of vertices are clearly structurally equivalent. In other words, unless we consider the edge weighted information, these graphs are virtually indistinguishable.

One way to overcome these issues is to compute sparser versions of the original graphs. There have been a number of alternative approaches to extract sparse structures from complete or dense weighted graphs. For instance, Ye et al. [15] and Silva et al. [18] take the widely used threshold-based approach. That is, they preserve only those edges whose weights fall into the larger 10% of weights. Unfortunately, these methods usually lead to a significant loss of information, since many weighted edges are discarded. Moreover, the resulting structure clearly depends on the choice of the threshold, and it is generally unclear how this threshold should be selected. This in turn results in very unstable and potentially disjoint graphs. Mantegna and Stanley [43] have extracted the minimum spanning tree over the original weighted adjacency matrix. This removes the need of selecting a threshold and the resulting instability, but it still suffers from the information loss problem of threshold-based methods.

To address the aforementioned problems, in this paper we propose to sparsify the original graph using the minimum spanning tree through the commute time matrix rather than the original weighted adjacency matrix. The commute time averages the time taken for a random walk to travel between a pair of vertices over all connecting paths, and is robust to the deletion of individual edges or paths (i.e., structural noise) unless these form bridges between connected components of a graph [44]. Thus, the resulting spanning tree structure through the commute time matrix not only retains salient structural characteristics of the graph, but also encapsulates proximity information residing on the discarded edges. In other words, we minimize the edge number in the original graph while preserving most of its path structure information. This, as shown in [45], can lead to a significant reduction of the computational complexity when applied to dense graphs. Most importantly, the commute time can easily accommodate weighted information residing on edges. As a result, the commute time not only represents an ideal candidate for sparsifying the structure of the original graph [46], but also allows us to separate otherwise structurally indistinguishable complete graphs.

The aim of this paper is to develop a new kernel for complete weighted graphs associated with discrete-time quantum walks. To this end, we propose a new framework of computing this kernel and proceed as follows. Given the commute time spanning tree representations of the original pair of graphs, we first simulate the evolution of a discrete-time quantum walk on each of the trees, where we make use of a novel weighted version of the Perron-Frobenius operator [47]. This in turn allows us to encode the weights on the edges of the commute time spanning tree in the initial state of the walk. Then, for each discrete-time quantum walk, we compute the associated time-averaged density matrix. Density matrices are matrices that describe quantum systems that are in a statistical mixture of quantum states, and they play a fundamental role in the quantum observation process. In our case, the time-averaged density matrix describes a statistical ensemble of quantum states encapsulating the time-evolution of a quantum walk. The diagonal of this matrix corresponds to the time-averaged probability distribution of the walk visiting each arc residing on the edge of the underlying graph. With a pair of density matrices to hand, the kernel between the original graphs is computed as the similarity between the associated time-averaged probability distributions. We show that the similarity between these distributions can be computed either as the negative exponential of their classical Jensen-Shannon divergence [48] or as their dot product. Both approaches lead to the definition of a positive definite kernel measure. Fig.1 illustrates the structure of the proposed framework to compute the kernel based similarity between two complete weighted graphs. Experiments on financial networks datasets as well as standard graph datasets abstracted from the bioinformatics domain demonstrate the effectiveness of the new kernel.

Finally, note that the proposed kernel is closely related to the kernel we introduced in [45], where we proposed to simplify the structure of the input graphs through commute time, to then compare them using discrete-time quantum walks. However, the proposed kernel significantly differs from [45] and has a number of important theoretical advantages. First, the computation of the initial quantum state for the proposed kernel is based on the newly introduced weighted Perron-Frobenius operator. As a result, only the proposed kernel can encapsulate the edge weight information of the original graphs. Second, unlike [45], where the similarity between the input graphs is computed using the quantum Jensen-

Shannon divergence between the density matrices associated with the graphs, here we look at the classical Jensen-Shannon divergence between the probability distributions associated with these density matrices. In particular, we show that these probability distributions can be easily transformed to distributions over the space of directed edge labels, defined as the union of the original edge and vertex labels, thus allowing this new kernel to incorporate both vertex and edge labels of the original graphs. Finally, in order to compute the divergence between the full density matrices, the kernel in [45] also requires the computation of an additional mixed density matrix for each pair of input graphs. This does not take into account the correspondences between the nodes of the input graphs and thus does not guarantee permutation invariance. On the other hand, we overcome this problem by computing the divergence between probability distributions over a common state space, i.e., the space of directed edge labels. As a result, the proposed kernel is both permutation invariant and positive definite.

The remainder of the paper is organised as follows. Section II introduces the necessary quantum mechanical background, while Section III reviews the concept of commute time and shows how to sparsify a graph using the commute time spanning tree. Section IV introduces the proposed kernel, which is extensively evaluated in Section V. Finally, Section VI concludes the paper.

II. Quantum Mechanical Background

A. Discrete-time Quantum Walks

In quantum mechanics, discrete-time quantum walks are defined as the quantum counterparts of classical discrete-time random walks [41]. Quantum processes are reversible, so in quantum walks the states need to specify both the current and the previous locations of the walk. Let us replace each edge $e(u,v) \in E$ with a pair of directed edges $e_d(u,v)$ and $e_d(v,u)$, and denote the new set as E_d. The state space of the discrete-time quantum walk is E_d and we represent the state of the quantum walker at $e_d(u,v)$ as $|uv\rangle$. That is, $|uv\rangle$ denotes the state in which the walk is at vertex v having previously been at vertex u. A general state of the walk is

$$|\psi\rangle = \sum_{e_d(u,v)\in E_d} \alpha_{uv}|uv\rangle. \tag{1}$$

where the quantum amplitudes α_{uv} are complex. The probability that the walk is in 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 time step, the quantum walk evolution is governed by the transition matrix $\boldsymbol{U}$, the entries of which indicate the transition probabilities between states, i.e.,

$$|\psi_{t+1}\rangle = \boldsymbol{U}|\psi_t\rangle. \tag{2}$$

Since the walk evolution is linear and conserves probability, $\boldsymbol{U}$ must be an unitary matrix, i.e., the inverse of $\boldsymbol{U}$ is equal to its Hermitian transpose $\boldsymbol{U}^\dagger$. A typical choice is to choose the Grover matrix [49] as the transition matrix, i.e.,

$$U_{(u,v),(w,x)} = \begin{cases} \frac{2}{d_x} - \delta_{ux}, & \text{v=w;} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where d_x corresponds to the vertex degree of vertex x, $U_{(u,v),(w,x)}$ indicates the quantum amplitude of the transition $e_d(u,v) \rightarrow e_d(w,x)$, and δ_{ux} is the Kronecker delta, i.e., $\delta_{ux} = 1$ if $u = x$ and 0 otherwise. For each state $|u_1v\rangle$, $\boldsymbol{U}$ assigns the same amplitude to all transitions $|u_1v\rangle \rightarrow |vu_i\rangle$, and a different amplitude to the transition $|u_1v\rangle \rightarrow |vu_1\rangle$, where u_i is a neighbour vertex of v. Note that the elements of $\boldsymbol{U}$ are real numbers, and they can be either positive or negative. This indicates that Eq.(3) allows *destructive interference* to take place as a consequence of negative quantum amplitudes appearing during the walk evolution.

B. The Weighted Perron-Frobenius Operator

In [47], it was shown that there exist an important link between the Perron-Frobenius operator and the transition matrix of discrete-time quantum walks. To illustrate this linkage, we commence by introducing the concepts of directed line graph and positive support for a matrix.

Definition 1 Let $G(V,E)$ be a given graph, the directed line graph $G_D(V_D,E_D)$ of G is a dual representation. To obtain G_D, we replace each edge $\{u,w\} \in E$ with a pair of directed edges $e_d(u,w)$ and $e_d(w,u)$ for vertices $u,w \in V$, and we denote this set as E_d. The directed line graph $G_D(V_D,E_D)$ is an oriented graph with vertex set V_D and edge set E_D as

$$\begin{cases} V_D = E_d, \\ E_D = \{(e_d(u,v), e_d(v,w)) \in E_d \times E_d\}. \end{cases} \tag{4}$$

where vertices $v,u,w \in V$ and $w \neq u$. Based on [47], the adjacency matrix $\boldsymbol{T} = [T_{i,j}]_{|V_D|\times|V_D|}$ of $G_D(V_D,E_D)$ is the Perron-Frobenius operator. □

Definition 2 Assume $\boldsymbol{M} = [M_{x,y}]_{m\times n}$ is a $m \times n$ matrix. Its positive support is a $m \times n$ matrix $\mathrm{S}^+(\boldsymbol{M}) = [s_{x,y}]_{m\times n}$ with

$$s_{x,y} = \begin{cases} 1, & M_{x,y} > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $1 \leq x \leq m$ and $1 \leq y \leq n$. □

If $\boldsymbol{U}$ is the unitary matrix of a discrete-time quantum walk on the graph $G(V,E)$, then, based on [47], the Perron-Frobenius operator $\boldsymbol{T}$ of $G(V,E)$ can be constructed from the positive support of $\boldsymbol{U}$, i.e.,

$$\boldsymbol{T} = \boldsymbol{S}^+(\boldsymbol{U}^\top). \tag{6}$$

For the directed line graph $G_D(V_D,E_D)$, each vertex $v_d \in V_D$ corresponds to a unique directed edge, thus the vertex set V_D of G_D essentially corresponds to the state space of the quantum walk. Furthermore, if there exists a directed edge from vertex $v_d \in V_D$ to vertex $u_d \in V_D$, the quantum walk on G allows a transition from the directed edge representing v_d to the directed edge representing u_d, and vice versa. These observations indicate that the discrete-time quantum walk on the original graph G can be seen as a walk evolved on the corresponding directed line graph G_D, where the transitions of the walk are constrained by the directed edges of G_D.

The directed line graph has a number of interesting properties which in turn highlight some advantages of discrete-time quantum walks over their classical counterparts. First, the directed line graph can represent the original graph in a higher

dimensional feature space. This is because the vertex set of the line graph corresponds to the set of directed edges of the original graph and its cardinality is usually greater than that of the vertex number of the original graph. Thus, compared to the continuous-time quantum walk [39] on the original graph, the discrete-time quantum walk on the directed line graph can capture richer structural characteristics. Second, the directed line graph is a backtrackless representation of the original graph structure, because the edges of the line graph are all directed. Since the transitions of the discrete-time quantum walk are constrained by the directed edges of the line graph, the walk cannot visit a vertex and then immediately return to the starting vertex through the same edge. As a result, the discrete-time quantum walk can significantly reduce the notorious tottering problem of the classical random walk [28]. Finally, since the discrete-time quantum walk and the directed line graph are related, the initial state of the quantum walk can be computed through the Perron-Frobenius operator T of the line graph [45]. Unfortunately, the operator T cannot reflect weight information residing on the edges of the original graph and thus its use leads to information loss. To overcome this shortcoming, we propose a new weighted Perron-Frobenius operator for the directed line graph. We compute the initial state by taking the square root of the sum of the out-degree and in-degree distributions from the new weighted operator.

Definition 3 (Initial State from Directed Line Graphs) Let $G(V,E)$ be a graph with weighted adjacency matrix A, E_d the set of directed edges for G, and $G_D(V_D,E_D)$ $(E_d = V_D)$ the directed line graph of G. Each element of the weighted Perron-Frobenius operator $\boldsymbol{T}^W$ of G satisfies

$$\boldsymbol{T}^W_{(u,v),(w,x)} = \begin{cases} A_d(u,v) + A_d(w,x), & v = w; \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $u,v,x,w \in V$, $A_d(u,v) = A(u,v)$ and $A_d(w,x) = A(w,x)$ are the weights of $e_d(u,v) \in E_d$ and $e_d(v,w) \in E_d$, $E_d = V_D$, and $(e_d(u,v), e_d(w,x)) \in E_D$. The initial state $|\psi_0\rangle$ for G through G_D is

$$|\psi_0\rangle = \sum_{e_d(u,v)\in E_d} \alpha^0_{uv} |uv\rangle, \tag{8}$$

where

$$\alpha^0_{uv} = \sqrt{\frac{\sum_{e_d(w,x)\in V_L}\{\boldsymbol{T}^W_{(u,v),(w,x)} + \boldsymbol{T}^W_{(w,x),(u,v)}\}}{\sum_{e_d(w,x),e_d(u,v)\in V_L}\{\boldsymbol{T}^W_{(u,v),(w,x)} + \boldsymbol{T}^W_{(w,x),(u,v)}\}}}. \tag{9}$$

The initial state $|\psi_0\rangle$ not only preserves the structural information of G_D, but also encapsulates the edge weight information in the original graph G. □

C. From Quantum Walks to Density Matrices

In quantum mechanics [47], a quantum system can be in a statistical ensemble of pure quantum states, where each pure state is described as a single ket vector $|\psi_i\rangle$ and has an associated probability p_i. The density matrix of this quantum system is $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. Consider a sample graph $G(V,E)$ and let $|\psi_t\rangle$ be the pure state that corresponds to a discrete-time quantum walk evolved from time step $t = 0$ to time step $t = T$ on G. The time-averaged density matrix ρ^T_G for G associated with the quantum walk is defined as $\rho^T_G = \frac{1}{T+1}\sum_{t=0}^{T} |\psi_t\rangle\langle\psi_t|$, since the state $|\psi_t\rangle$ at time t can be computed by $|\psi_t\rangle = \boldsymbol{U}^t |\psi_0\rangle$, where $|\psi_0\rangle$ is the initial state of the quantum walk, and $\boldsymbol{U}$ is the transition matrix. Given the initial state $|\psi_0\rangle$, ρ^T_G can be re-written as

$$\rho^T_G = \frac{1}{T+1}\sum_{t=0}^{T} (\boldsymbol{U})^t |\psi_0\rangle\langle\psi_0| (\boldsymbol{U}^\top)^t, \tag{10}$$

where $|\psi_0\rangle$ is defined in Def.3 through the weighted Perron-Frobenius operator. ρ^T_G describes a quantum system that consists of a family of equally probable pure states, which are defined by the quantum walk evolution from time step $t = 0$ to $t = T$. Furthermore, we can compute the time-averaged probability $p\{e_d(u,v)\}$ of the quantum walk visiting $e_d(v,u)$

$$p\{e_d(u,v)\} = \rho^T_G((u,v),(u,v)), \tag{11}$$

where $v,u \in V$ and (u,v) indexes $e_d(u,v)$.

III. Graph Simplification through Commute Time

The aim of this paper is to develop a novel similarity measure between complete weighted graphs. Directly simulating the evolution of a discrete-time quantum walk on these graphs tends to be elusive, due to the high computational complexity. To overcome this problem, we propose to sparsify the original graphs through the commute time matrix [44].

We first review the concept of commute time. Let $\mathbb{G}$ be a set of complete weighted graphs. Assume $G(V,E)$ is a sample graph from $\mathbb{G}$ with an edge weight function $\mathrm{w} : V \times V \to \mathbb{R}^+$. If $\mathrm{w}\{u,v\} > 0$ or $\mathrm{w}\{v,u\} > 0$, there exists an undirected edge $\{u,v\} \in E$ between vertices $v \in V$ and $u \in V$, i.e., v and u are adjacent. Let A denote the adjacency matrix of $G(V,E)$, with entries $A(u,v) = A(v,u) = \mathrm{w}\{u,v\}$. The degree matrix D of G is a diagonal matrix, where each diagonal entry $D(u,u)$ is computed by summing the corresponding row or column of A, i.e., $D(u,u) = \sum_v A(u,v) = \sum_v A(v,u)$. Then, the graph Laplacian matrix L is computed by subtracting A from D, i.e., $L = D - A$. The spectral decomposition of L is defined as $L = \Phi\Lambda\Phi^T$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$ is a $|V| \times |V|$ diagonal matrix with ascending eigenvalues as elements, i.e., $0 = \lambda_1 \le \lambda_2 \le ... \le \lambda_{|V|}$, and Φ is a $|V| \times |V|$ matrix $\Phi = (\phi_1|\phi_2|...|\phi_{|V|})$ with the corresponding ordered eigenvectors as columns. The hitting time $H(u,v)$ between a pair of vertices $v \in V$ and $u \in V$ of $G(V,E)$ is defined as the expected number of steps of a random walk starting from u to v. Similarly, the commute time $C(u,v)$ is computed as the expected number of steps of the walk starting from u to v, and then returning to u, i.e., $C(u,v) = H(u,v) + H(v,u)$. The commute time $C(u,v)$ can be computed in terms of the unnormalized Laplacian eigendecomposition [44]

$$C(u,v) = \sum_{u\in V} D(u,u) \sum_{j=2}^{|V|} \frac{1}{\lambda_j}(\phi_j(u) - \phi_j(v))^2. \tag{12}$$

Similarly to [46], we propose to simplify the graph structure by computing a modified commute time spanning tree and reduce the number of edges to $n-1$. More specifically, for

a complete weighted graph $G(V,E) \in \mathbb{G}$ and its weighted adjacency matrix A, we first compute its commute time matrix C and its associated modified commute time matrix Q, where $Q = C \bigodot A$. We then use Q as the new adjacency matrix of $G(V,E)$. Based on Kruskal's method [50], we compute the minimum spanning tree $\mathcal{S}(\mathcal{V},\mathcal{E})$ over Q, where $\mathcal{V} = V$.

Discussion The commute time provides a number of theoretical advantages. **First**, the commute time amplifies the affinity between pairwise vertices [51] and is robust under the perturbation of the graph structure [44]. Thus, the minimum spanning tree $\mathcal{S}$ constructed on the modified commute time matrix can reflect the dominant structural information of the original graph G, while yielding a sparser structure [46]. **Second**, as we have stated, the minimum spanning tree $\mathcal{S}$ can reduce the edge number of G to $n-1$. There will be $2(n-1)$ directed edges in $\mathcal{S}$, thus evolving the discrete-time quantum walk on the spanning tree has time complexity $O(n^3)$. This is because the computation of the commute time matrix is based on the eigendecomposition of the normalized Laplacian. Note that this represents a considerable improvement from the original time complexity $O(n^6)$ of simulating the quantum walk on the original graph G. **Third**, the commute time can accommodate the weighted information residing on the graph edges, since it can be computed in terms of the unnormalized Laplacian matrix through the weighted adjacency matrix [44]. **Fourth**, the spanning tree is a simplified structure of the original graph and some weighted edges of the original graph are not encapsulated in the spanning trees. However, since the commute time is defined as the expected step number of the random walk departing from one vertex and returning to the same vertex [52], i.e., the commute time reflects the integrated effectiveness of all possible paths between pairwise vertices of the original graph [44]. Thus, the weighted edges of the spanning tree from the modified commute time matrix can also reflect the weighted information of deleted edges in the original graph. In summary, the minimum spanning tree provides an elegant way to probe the structure of complete weighted graphs using discrete-time quantum walks.

IV. A Kernel for Complete Weighted Graphs

In this section we introduce the proposed graph kernel. Given a set of complete weighted graphs, we show that we can associate with each graph a probability distribution over the directed edge labels, which is induced by the time-average probability distribution of the discrete-time quantum walk defined in Section II. Then, we define two kernels between a pair of graphs in terms of the similarity between the corresponding distributions over the directed edge labels. An instance of computing the proposed kernels between pairwise graphs are exhibited in Fig.2.

A. The Jensen-Shannon Divergence

The Jensen-Shannon divergence is a non-extensive mutual information measure defined between probability distributions [53]. Let $\mathcal{P} = (p_1,\ldots,p_m,\ldots,p_M)$ and $\mathcal{Q} = (q_1,\ldots,q_m,\ldots,q_M)$ be a pair of probability distributions, then the divergence measure between the distributions is

$$\begin{aligned} D_{JS}(\mathcal{P},\mathcal{Q}) &= H_S(\frac{\mathcal{P}+\mathcal{Q}}{2}) - \frac{1}{2}H_S(\mathcal{P}) - \frac{1}{2}H_S(\mathcal{Q}) \\ &= -\sum_{m=1}^{M}\frac{p_m+q_m}{2}\log\frac{p_m+q_m}{2} \\ &\quad + \sum_{m=1}^{M} p_m \log p_m + \sum_{m=1}^{M} q_m \log q_m. \end{aligned} \tag{13}$$

where $H_S(\mathcal{P}) = \sum_{m=1}^{M} p_m \log p_m$ is the Shannon entropy associated with $\mathcal{P}$. D_{JS} is always negative definite, symmetric, well defined, and bounded, i.e., $0 \leq D_{JS} \leq 1$.

B. The Proposed Graph Kernel Over Directed Edge Labels

Let $\mathbb{G} = \{G_1,\ldots,G_a,\ldots,G_b,\ldots,G_N\}$ denote a set of complete weighted graphs. For a sample graph $G(V,E) \in \mathbb{G}$, we first transform G into a minimum spanning tree $\mathcal{S}(V,\mathcal{E})$ over its modified commute time matrix, based on the method introduced in Section III. Note that the edges of the commute time spanning tree are attributed with the modified commute time between the corresponding pair of vertices. Let $\mathcal{E}_d$ be the directed edges set of $\mathcal{S}$. Based on Section II, we simulate the discrete-time quantum walk on the spanning tree $\mathcal{S}$, and we associate with each directed edge $e_d(u,v) \in \mathcal{E}_d$ $(v,u \in V)$ the time-average probability $p\{e_d(u,v)\}$ from the quantum walk. Let $L(v)$ be the discrete label associated with the vertex $v \in V$. The directed edge label $\mathcal{L}(e_d(u,v))$ of $e_d(u,v)$ is the label union of the start vertex u and end vertex v, i.e.,

$$\mathcal{L}(e_d(u,v)) = \{L(u), L(v)\}. \tag{14}$$

Note that, the resided edge $\{u,v\} \in E$ of $e_d(u,v) \in \mathcal{E}_d$ on the original graph $G(V,E)$ may also have discrete label $L(u,v)$. For this instance, $\mathcal{L}(e_d(u,v))$ can be re-written as

$$\mathcal{L}(e_d(u,v)) = \{L(u), L(u,v), L(v)\}. \tag{15}$$

Thus, we can simultaneously accommodate both discrete vertex and edge labels. Let $\mathbf{L}$ be the set of all possible directed edge labels and $l \in \mathbf{L}$ is a label. We assign l the probability

$$p(l) = \sum_{v,u \in V} p\{e_d(u,v)\}, \tag{16}$$

where each $e_d(u,v)$ satisfies $\mathcal{L}(e_d(u,v)) = l$. Note that, for a graph, if the edge label l does not exist, $p(l) = 0$.

Let $G_a(V_a,E_a) \in \mathbb{G}$ and $G_b(V_b,E_b) \in \mathbb{G}$ denote a pair of graphs, and $\mathcal{S}_a(V_a,\mathcal{E}_a)$ and $\mathcal{S}_b(V_b,\mathcal{E}_b)$ the associated minimum spanning trees. Based on Eq.(16), we compute the probability distributions over directed edge labels as

$$\mathcal{P}_a = \{p_a(1),\ldots,p_a(l),\ldots,p_a(|\mathbf{L}|)\},$$

and

$$\mathcal{P}_b = \{p_b(1),\ldots,p_b(l),\ldots,p_b(|\mathbf{L}|)\},$$

associated with the quantum walks on $\mathcal{S}_a$ and $\mathcal{S}_b$.

Given this setting, we propose to compute the kernel between G_a and G_b in terms of the similarity between $\mathcal{P}_a$ and $\mathcal{P}_b$. We consider two alternative ways of computing this similarity, both resulting in a positive definite kernel.

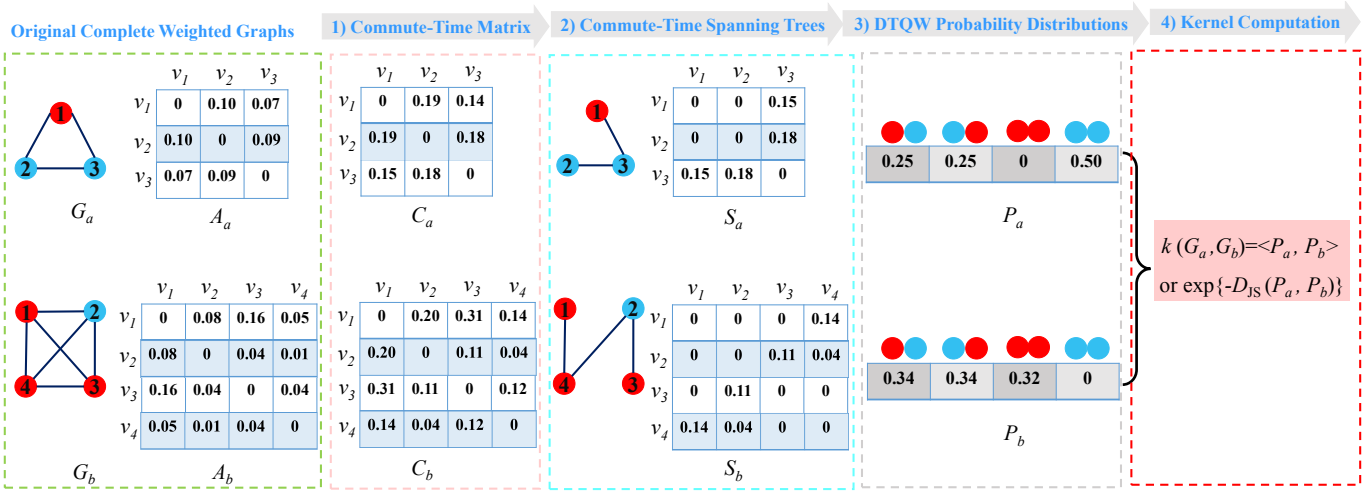


Fig. 2. An instance of computing the proposed kernels. Let G_a and G_b be two input complete weighted graphs, and A_a and A_b their original weighted vertex adjacency matrices. Note that each color residing on a vertex corresponds to a vertex label. The vertices have the same vertex label if their colors are the same. Specifically, the procedure of computing the proposed kernel between G_a and G_b consists of four steps. (1) The first step computes the commute time matrices, and employs their modified versions C_a and C_b as the new weighted adjacency matrices of G_a and G_b, respectively. (2) The second step computes the minimum spanning trees S_a and S_b of G_a and G_b over the modified commute time matrices C_a and C_b. Here, the structures of S_a and S_b are sparser than that of their original graphs G_a and G_b. (3) The third step probes the spanning trees S_a and S_b in terms of the discrete-time quantum walk, and computes the probability of the quantum walk visiting the directed edges residing on the original edges of the spanning trees. More specifically, this process computes the probability distribution vectors P_a and P_b for G_a and G_b, where each vector is spanned by different directed edge labels and each element of the vector corresponds to the sum of the probabilities of the quantum walks visiting the directed edges having the same directed edge label. Here, the combination of each pair of colors corresponds to a directed edge label, where the first and second colors of the combination correspond to the vertex labels of the start and end vertices of a directed edge. (4) The final step computes the kernel based similarity using either a) the dot product or b) the negative exponential of the Jensen-Shannon divergence between the probability distributions P_a and P_b.

1) Measuring Similarity Using the Dot Product: Recall that a graph kernel is a positive definite similarity measure which corresponds to the dot product between graphs in an implicit feature space. If we consider $\mathcal{P}_a$ and $\mathcal{P}_b$ as the feature space embeddings of $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$, then the kernel between them can be defined as

$$k_{DP}(G_a, G_b) = \langle \mathcal{P}_a, \mathcal{P}_b \rangle, \tag{17}$$

where $\langle . \rangle$ denotes the dot product. The positive definiteness of the kernel trivially follows from Eq.(17). Note that in theory one can employ any standard vector-based kernel measure between the probability distributions $\mathcal{P}_a$ and $\mathcal{P}_b$ as the similarity measure, e.g., the Radial Basis Function (RBF) kernel [54], Laplacian kernel [55], etc. However, these vectorial kernels are instances of parametric kernels, i.e., they introduce additional parameters that need to be adjusted in order to achieve the best performance. By contrast, the kernel defined in Eq.(17) is parameter-free and thus does not require the manual setup of additional parameters during the kernel computation

2) Measuring Similarity Using the JS Divergence: Based on Eq.(13), the graph kernel between $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$ associated with the Jensen-Shannon divergence is defined as

$$\begin{aligned} K_{JS}(G_a, G_b) &= \exp\{-D_{JS}(\mathcal{P}_a, \mathcal{P}_b)\} \\ &= \exp\{-H_S(\frac{\mathcal{P}_a + \mathcal{P}_b}{2}) + \frac{H_S(\mathcal{P}_a) + H_S(\mathcal{P}_b)}{2}\} \\ &= -\sum_{l \in \mathbf{L}} \frac{p_a(l) + p_b(l)}{2} \log \frac{p_a(l) + p_b(l)}{2} \\ &+ \sum_{l \in \mathbf{L}} p_a(l) \log p_a(l) + \sum_{l \in \mathbf{L}} p_b(l) \log p_b(l), \end{aligned} \tag{18}$$

where $\frac{\mathcal{P}_a + \mathcal{P}_b}{2}$ is a composite probability distribution and $\frac{p_a(l)+p_b(l)}{2}$ is computed over the same label $l \in \mathbf{L}$. In other words, the similarity between the input graphs is defined in terms of the similarity between the probability distributions over their labels. Note that since these probability distributions are computed over the space of directed edges labels (and not the directed edges themselves), we do not require the two graphs to have the same number of edges.

To see why Eq.(18) defines a positive definite kernel, recall that the Jensen-Shannon divergence between probability distributions is a symmetric dissimilarity measure [48]. Since the kernel is computed as the negative exponential of the divergence measure, it follows that it is positive definite.

The computational complexity of both kernels is the same. Let n be the maximum number of vertices in a pair of graphs. Then computing $\mathrm{K_{DP}}$ or $\mathrm{K_{JS}}$ between these graphs has time complexity $O(n^3)$. In fact, the cost of computing the kernels is dominated by simulating the discrete-time quantum walk on the spanning trees extracted from the original graphs, and this has time complexity $O(n^3)$, as explained in Section III.

Finally, note that the computation of $\mathrm{K_{DP}}$ discards any information on a label if this does not appear in both input graphs. For instance, given two graphs G_a and G_b, if the label l appears in G_a but not in G_b, we have $p_a(l) > 0$ and $p_b(l) = 0$. As a result, by taking the dot product between the probability vectors, the label l will not influence the resulting kernel value. By contrast, in the computation of $\mathrm{K_{JS}}$ we make use of all labels, including those that appear in only one of the two input graphs. This in turn suggests that $\mathrm{K_{JS}}$ may potentially reflect richer graph characteristics than $\mathrm{K_{DP}}$ for some instances.

TABLE I
SUMMARY STATISTICS FOR THE SELECTED GRAPH DATASETS

Property	**The Proposed Kernels**	QJSK [42]	QJSKT [45]	QEMK [56]
Permutation Invariant	Yes	No	No	Yes
Time Complexity for N Graphs	$O(Nn^3 + N^2n)$	$O(N^2n^6)$	$O(N^2n^3)$	$O(Nn^6 + N^2n^4)$
Time Complexity for Pairwise Graphs	$O(n^3)$	$O(n^6)$	$O(n^3)$	$O(n^6)$
Positive Definite	Yes	Yes	Yes	No
Accommodate Attributed Graphs	Yes	No	No	No
Accommodate Edge Weights	Yes	No	No	No

C. Discussions and Related Works

The proposed graph kernels are related to the quantum Jensen-Shannon kernel (QJSK) [42] and its faster variant (QJSKT) [45], as well as the quantum edge-based matching kernel (QEMK) [56]. All these kernels use discrete-time quantum walks to probe the graph structure. Moreover, the proposed kernels are also similar to the QJSKT kernel, which is also based on the spanning trees extracted from original graph through the commute time. Thus, the QJSKT kernel for a pair of graphs both having n vertices also has time complexity $O(n^3)$. This is significantly more efficient than the QJSK kernel that requires time complexity $O(n^6)$. However, there are five significant theoretical differences between the proposed kernels and these kernels. These differences are listed in Table I and discussed as follows.

First, as we have stated in Section III, the minimum spanning tree, which is extracted from the original graph through the commute time, can reflect weight information residing on the edges of the original graph. Thus, unlike the QJSK kernel between original graphs, the proposed kernels between commute time spanning trees not only overcome the shortcoming of inefficiency, but also encapsulate the edge weight information of original graphs through the proposed weighted Perron-Frobenius operator.

Second, like the proposed kernels, the QJSKT kernel is also defined on the commute time spanning tree and requires the same time complexity $O(n^3)$ for a pair of graphs. The computation of the initial quantum state of the QJSKT kernel is based on the unweighted Perron-Frobenius operator rather then the weighted operator. This is similar to the QJSK kernel. Thus, unlike the proposed kernels, the QJSKT kernel ignores edge weight information of the spanning trees, i.e., it does not reflect the edge weight information of the original graphs.

Third, unlike the QJSKT and QJSK kernels, the proposed kernels are based on the dot product and the classical Jensen-Shannon divergence between probability distributions over directed edge labels, respectively. These distributions, on the other hand, correspond to the diagonals of the density operators of the discrete-time quantum walks. Since the labels for the proposed kernels are computed by taking the union of the corresponding vertex/edge labels, the proposed kernels can also accommodate vertex/edge attributed graphs. By contrast, the QJSKT and QJSK kernels are based on the quantum Jensen-Shannon divergence [45] between the density operators associated with the discrete-time quantum walks. This in turn requires computing their eigendecomposition, which has time complexity $O(n^3)$. On the other hand, both the dot product and the classical Jensen-Shannon divergence between a pair of probability distributions have time complexity $O(n)$. As a result, for a set of N graphs each having n vertices, the proposed kernels only have time complexity $O(Nn^3 + N^2n)$. This is significantly more efficient than the QJSKT kernel, which has time complexity $O(N^2n^3)$.

Fourth, both the QJSKT kernel and the QJSK kernel require computing a composite density operator from the pair of density operator under comparisons. However, when computing this composite density operator both kernels do not take into account the correspondences between the vertices of the directed line graphs, i.e., the directed edges residing on the original graph edges. As a result, neither kernel is permutation invariant. By contrast, in the proposed kernels we overcome the problem by comparing probability distributions over the space of the directed edge labels. The proposed kernels are thus permutation invariant, since permuting the directed edge order (i.e., the adjacency matrix of the corresponding directed line graph) does not change its set of directed edge labels. In other words, our kernels gives a more precise kernel based similarity measure than the QJSKT and QJSK kernels.

Finally, like the QJSK kernel, the QEMK kernel also involves evolving a discrete-time quantum walk, but on the original graph. Thus, the QEMK kernel also suffers from the inefficiency of simulating quantum walks on the original graph. Moreover, computing the QEMK kernel between a pair of graphs requires aligning their vertices through their depth-based representations [45]. Since this alignment step is not guaranteed to be transitive, the resulting kernel is not positive definite (***pd***) [57]. By contrast, the proposed kernels are ***pd***.

V. EXPERIMENTS

As an example of a complete weighted network, we consider the case of time-varying financial networks. The **NYSE dataset** consists of a series of networks for each trading day abstracted from the closing price of stocks from the New York Stock Exchange (NYSE) database [18], which consists of 3799 stocks and their associated daily prices. The stock prices were obtained from the Yahoo financial dataset (http://finance.yahoo.com) [15]. To extract the network representations, we select 347 stocks that were traded from January 1986 to February 2011, i.e., for a total of 6004 days. We select a period of 28 days as the time window to analyse the similarity of the closing prices of different stocks and we move this window along the 6004 trading days to construct a time varying sequence of stock prices. More specifically, each chronological window of the sequence encapsulates a time series of stock return values over a corresponding period of 28 days. For each time window, i.e., each trading day, we represent the trades between each pair of stock as a network where the connection weight between two stocks is the Euclidean distance between their time series. The resulting structure is a time-varying network on 347 vertices over 5976 days. Note

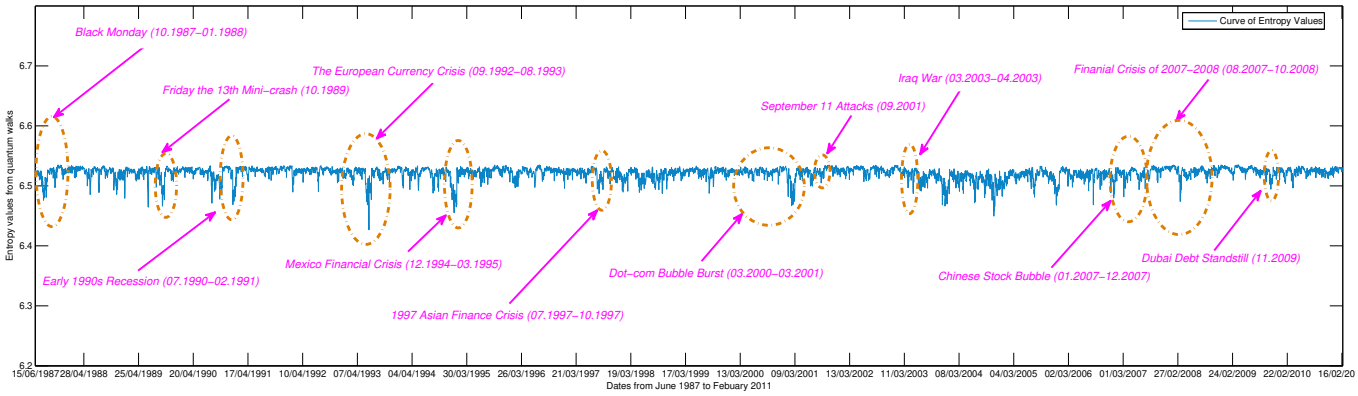
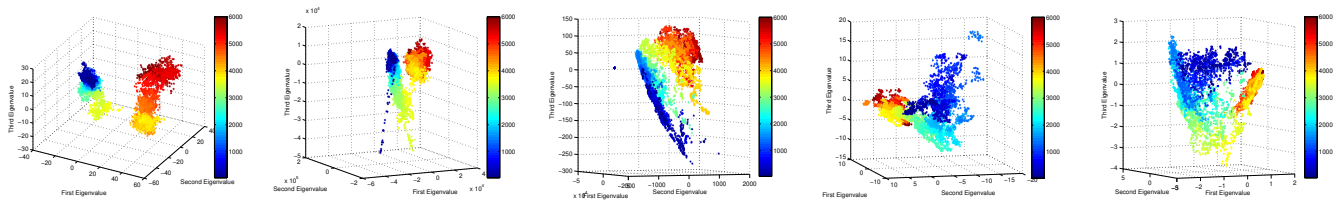


Fig. 3. Shannon entropy versus time for time-varying financial networks.



(a) QK kPCA Embeddings (b) WLSK kPCA Embeddings (c) JSGK kPCA Embeddings (d) QJSK kPCA Embeddings (e) FLGK kPCK Embeddings
Fig. 4. (Color online) Path of financial networks over 5976 days based on kPCAs of different graph kernels.

that each network is a complete weighted graph, and each vertex label corresponds to a corresponding stock name (the edges do not have discrete labels). To our knowledge, the aforementioned existing state-of-the-art R-convolution graph kernels cannot directly accommodate this kind of structures, since none of them can deal with complete weighted graphs.

A. Evaluations of the Network Entropy

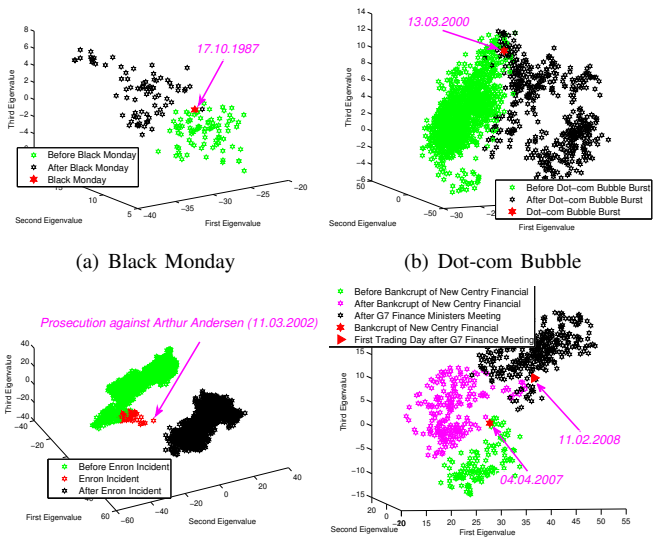
We commence by investigating the Shannon entropies associated with the directed edge label probability distributions induced by the discrete-time quantum walks. These entropies play a significant role in determining the kernel performance. Specifically, we explore the evolutionary behavior of the NYSE stock market by computing the entropy of the time varying financial networks at each time step. This allows us to investigate whether the evolutionary behaviour can be understood through the variation of the network entropy, i.e., we aim to analyze whether abrupt changes in network structure or different evolutionary epochs can be characterized by the entropy measure. Note that, we evolve the quantum walks from $t = 1$ to $t = 25$, i.e., we set the maximum t as $T = 25$, because the Shannon entropy computed through the quantum walk tends to reach a limiting value when $T \geq 25$. The results are shown in Fig. 3, where the x-axis represents the date (time) and the y-axis represents entropy values. Fig. 3 indicates that most of the significant fluctuations in the entropy time series correspond to different financial crises, e.g., Black Monday [58], the Friday the 13th minicrash [59], the Dot-com Bubble Burst, the Mexico Financial Crisis, the Iraq War, and the Subprime Crisis [60], etc. This is because the time-varying financial network experiences dramatic structural changes when a financial crisis occurs. For instance, some significant Internet companies that led to a rapid increase of both market confidence and stock prices were identified during the period of the Dot-com Bubble [61]. This noticeably modified the subsequent relations between stock, and this phenomenon can be captured by examining the variation of the Shannon entropy values.

Note that, although Fig. 3 demonstrates that the entropy is effective in detecting the extreme events in the financial network evolution, the time series is one dimensional and hence overlooks information concerning detailed changes in network structure. By contrast, the proposed quantum walk kernels can map the network structures into a high dimensional space by kernelizing the entropy and better preserve structural information contained in the networks.

B. Quantum Kernel Embeddings from kPCA

In this subsection, we explore the effectiveness of the proposed quantum kernels on the NYSE dataset. To this end, we apply the new kernels to the time-varying financial networks with the objective of analyzing whether abrupt changes in network evolution can be distinguished. Note that in the remainder of this subsection we choose to show only the results for the kernel based on the dot product, as we observed that these were very similar to the ones computed using the Jensen-Shannon divergence and the computation is faster.

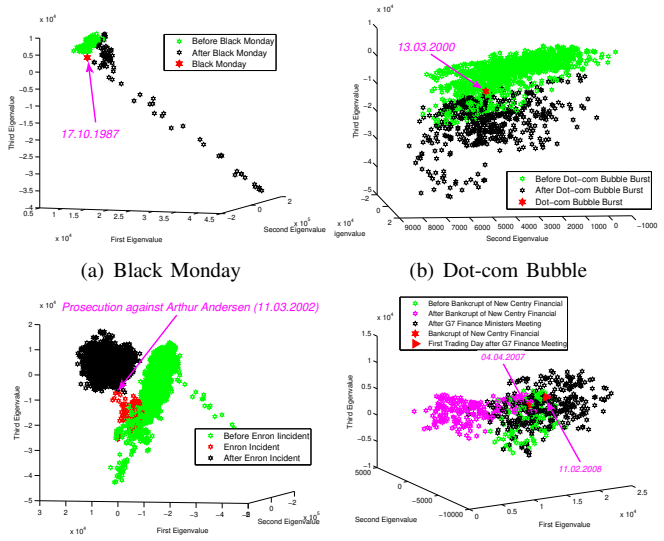
We commence by setting $T = 25$ for the evolution of the required discrete-time quantum walks. We perform kernel Principle Component Analysis (kPCA) [62] on the kernel matrix of the financial networks (from January 1986 to February 2011) using the proposed kernel, and we embed the networks into a 3-dimensional component space. The results from the kPCA are visualized using the first three principal components and



(a) Black Monday (b) Dot-com Bubble

(c) Enron Incident (d) Subprime Crisis

Fig. 5. The 3D embedding of the financial networks during different finance crises based on kPCA from QK (for Commute Time Spanning Trees).



(a) Black Monday (b) Dot-com Bubble

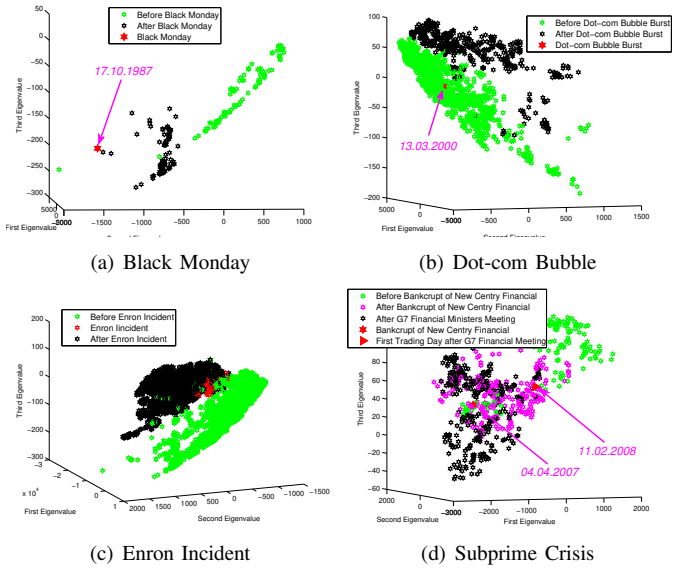
(c) Enron Incident (d) Subprime Crisis

Fig. 6. The 3D embedding of the financial Networks during Different Finance Crises based on kPCA from WLSK.

are exhibited in Fig. 4(a). Moreover, we compare the proposed quantum kernel with four state-of-the-art kernels, i.e., the Weisfeiler-Lehman subtree kernel (WLSK) [30], the Jensen-Shannon graph kernel (JSGK) [37], the quantum Jensen-Shannon kernel (QJS) [39] and the feature space Laplacian graph kernel (FLGK) [63]. Note that, the WLSK is an instance of the R-convolution kernel, and it cannot accommodate either complete weighted graphs or weighted graphs. Thus, we apply the WLSK kernel to the transformed commute time spanning trees and ignore the weight information on the tree edges. The JSGK kernel is a mutual information kernel associated with the steady state random walk, the QJS kernel is a quantum kernel associated with the continuous-time quantum walk, and the FLGK kernel is a global kernel associated with the Laplacian matrix. Since all the JSGK, QJS and FLGK kernels can accommodate edge weight, we directly apply these kernels to the original financial networks. Moreover, since each vertex label appears just once for each network, we establish the required correspondences between a pair of networks through the vertex labels for the JSGK and QJS kernels. We also perform kPCA on the resulting kernel matrices and embed the networks into a 3-dimensional principal component space. The embedding results for the WLSK, JSGK, QJS and FLGK kernels are visualized in Fig. 4(b), 4(c), 4(d) and 4(e). The plots in Fig. 4 indicate the paths of the time-varying financial networks in the different kernel spaces over 5976 trading days. The color bar beside each plot represents the date in the time series. It is clear that the embedding given by the proposed kernel shows a better manifold structure. Moreover, an interesting phenomenon in Figs.4(a) and 4(b) is that networks from Jan 1986 to Feb 2011 are well divided into two clusters. The clusters found by our kernel are better separated.

To take this study one step further, we show in detail the kPCA embeddings during four different financial crisis periods. Specifically, Fig. 5(a) corresponds to the Black Monday period (***from 15th Jun 1987 to 17th Feb 1988***), Fig. 5(b) to the Dot-com Bubble period (***from 3rd Jan 1995 to 31st Dec 2001***), Fig. 5(c) to the Enron Incident period (***the red points, from 16th Oct 2001 to 11th Mar 2002***), and Fig. 5(d) to the Subprime Crisis period (***from 2nd Jan 2006 to 1st Jul 2009***). Fig. 5(a) indicates that Black Monday (***17th Oct, 1987***) is a crucial financial event, as the network embedding points before and after the event are divided into two clusters. Similarly, Fig. 5 indicates that the Dot-com Bubble Burst (***13rd Mar, 2000***) in Fig. 5(b), the Enron Incident period (***from 2nd Dec 2001 to 11th Mar 2002***) in Fig. 5(c), the New Century Financial crisis (***4th April, 2007***) and the First Trading Day after G7 Finance Meeting (***2nd Nov, 2008***) in Fig. 5(d) are also crucial financial events. The network embedding points before and after these events are separated into distinct clusters. Moreover, points corresponding to the crucial events are midway between the two clusters. Another interesting feature in Fig. 5(c) is that the networks between 1986 and 2011 are separated by the Prosecution against Arthur Andersen (***3rd Nov, 2002***). The prosecution is closely related to the Enron Incident. As a result, the Enron Incident can be seen as a watershed at the beginning of 21st century, that significantly distinguishes the financial networks of the 21st and 20th centuries.

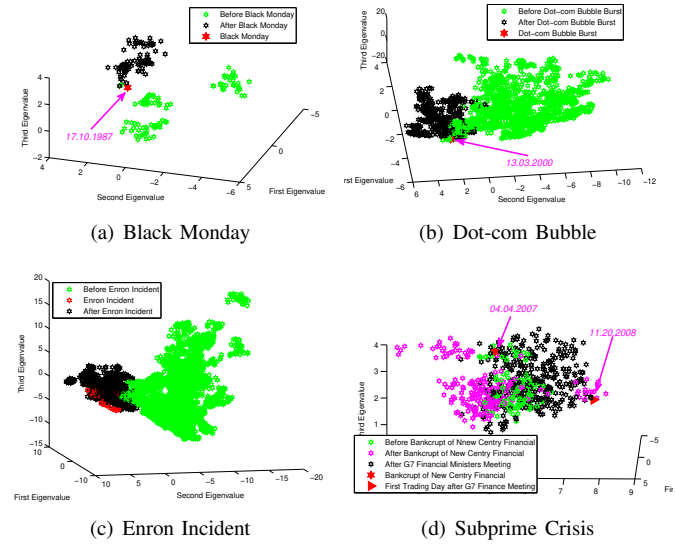
Similar embeddings are also found for a) the WLSK kernel in Fig. 6, b) the JSGK kernel in Fig. 7, c) the QJS kernel in Fig. 8, and d) the FLGK kernel in Fig. 9. We observe that the proposed kernel outperforms the WLSK, JSGK, QJS and FLGK kernels in terms of the distributions of networks. More specifically, for the proposed quantum kernel, the boundaries between clusters are clear and the clusters are tighter. The reasons for this are threefold. First, unlike the proposed kernel, the WLSK kernel cannot encapsulate weight information residing on the edges. As a result, the WLSK kernel may lose important information from the financial networks. Second, the state space of the discrete-time quantum walk is larger than that of the steady state random walk and the continuous-time quantum walk, and thus can reflect more topological information. Moreover, the discrete-time quantum walk can



(a) Black Monday (b) Dot-com Bubble

(c) Enron Incident (d) Subprime Crisis

Fig. 7. The 3D embedding of the financial networks during different finance crises based on kPCA from JSGK (for original complete weight graphs).



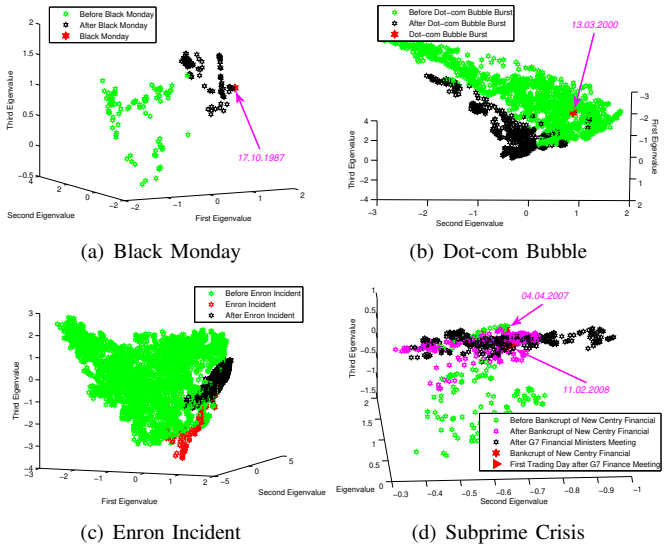
(a) Black Monday (b) Dot-com Bubble

(c) Enron Incident (d) Subprime Crisis

Fig. 8. The 3D embedding of the financial networks during different finance crises based on kPCA from QJS (for original complete weighted graphs).

limit the tottering problem arising in the classical random walk. As a result, the proposed quantum kernel can reflect richer characteristics than the JSGK and QJS kernels. Third, the discrete-time quantum walk can reflect more information of a network structure than its simple Laplacian matrix. As a result, the proposed kernel can better discriminate different financial network structures than the FLGK kernel. The above observations demonstrate that our kernel can distinguish different types of network evolution for time-varying financial networks, and outperforms state-of-the-art methods.

Since the commute time minimum spanning tree plays an important role in determining the performance of the proposed quantum kernel, we also provide experiments to demonstrate the advantage of using this kind of tree structures. Our experiments demonstrate that when the minimum spanning trees are abstracted through the commute time matrix, they can better preserve the graph characteristics than those abstracted through



(a) Black Monday (b) Dot-com Bubble

(c) Enron Incident (d) Subprime Crisis

Fig. 9. The 3D embeddings of the financial networks during different crises based on kPCA from FLGK (for original complete weighted graphs).

the original graph adjacency matrix. Moreover, to further demonstrate the effectiveness of the proposed kernel, we also evaluate its performance on time-varying financial networks where the edges are weighted according to the correlation instead of the Euclidean distance between the corresponding time series. The experimental results demonstrate again that the proposed kernel can outperform alternative methods on this kind of financial networks. Indeed, only the proposed kernel can simultaneously preserve the network characteristics in the Hilbert space and accommodate the edge weights. Due to the limit space of this manuscript, the results of these experimental evaluations are included in the supplementary materials.

C. Experiments on General Graphs

We conclude our experiments by showing the classification performance of the proposed kernels on more general standard graph datasets abstracted from the field of bioinformatics. These datasets include the MUTAG, PPIs(Proteobacteria40 PPIs and Acidobacteria46 PPIs), PTC(MR) and NCI1 datasets and their information is shown in Table II. More introduction of these datasets can be found in [45]. Note that, unlike the time-varying networks, the graphs from these datasets are neither edge weighted nor completed graphs. To accommodate these graphs using the proposed kernels, we proceed as follows. For each graph, we first compute its commute time matrix, and assign each edge a weight using the commute time value between the pair of vertices connected by the edge, i.e., we transform each original graph into a weighted graph by assigning each original edge a weight using the corresponding commute time value. If the ratio between the edge and vertex number of the graph is larger than 1.5 (i.e., the graph is dense), we compute the weighted minimum spanning tree over its weighted adjacency matrix as the new graph structure. We measure the kernel values using the proposed kernels between the original or sparsified structures. Furthermore we also strengthen the labels using the Weisfeiler-Lehman (WL) method [64] based on different iterations h. Recall that the

TABLE II
SUMMARY STATISTICS FOR THE GRAPH DATASETS.

Datasets	MUTAG	PPIs	PTC	NCI1
Max # vertices	28	232	109	111
Min # vertices	10	3	2	3
Avg # vertices	17.93	109.60	25.60	29.87
Max # edges	33	1503	108	119
Min # edges	10	2	1	2
Avg # edges	19.79	432.18	25.96	32.30
# graphs	188	86	344	4110
# classes	2	2	2	2
Avg # edges/Avg # vertices	1.10	3.94	1.01	1.08

TABLE III
ACCURACY (IN % ± STANDARD ERROR) AND RUNTIME (IN SECOND).

Datasets	MUTAG	PPIs	PTC	NCI1
$\mathrm{K_{DP}}$	82.38 ± 0.55	88.62 ± 0.86	58.62 ± 0.69	85.03 ± 0.12
$\mathrm{K_{JS}}$	**85.44** ± 0.58	**90.62** ± 0.90	**60.70** ± 0.62	**85.18** ± 0.10
WLSK	82.05 ± 0.57	78.50 ± 1.40	56.05 ± 0.51	80.68 ± 0.27
QJSK	83.83 ± 0.49	70.57 ± 1.20	58.23 ± 0.80	67.40 ± 0.20
QJSKT	81.55 ± 0.53	68.12 ± 0.84	57.44 ± 0.36	67.00 ± 0.15
SPGK	83.38 ± 0.81	61.12 ± 1.09	56.55 ± 0.53	74.21 ± 0.30
JSGK	83.11 ± 0.80	57.87 ± 1.36	57.29 ± 0.41	62.50 ± 0.33
BRWK	77.50 ± 0.75	53.50 ± 1.47	53.97 ± 0.31	60.34 ± 0.17

Datasets	MUTAG	PPIs	PTC	NCI1
$\mathrm{K_{DP}}$	$2.7 \cdot 10^1$	$5.5 \cdot 10^1$	$2.7 \cdot 10^1$	$5.4 \cdot 10^2$
$\mathrm{K_{JS}}$	$1.2 \cdot 10^1$	$1.7 \cdot 10^2$	$2.7 \cdot 10^2$	$4.1 \cdot 10^4$
WLSK	$0.4 \cdot 10^1$	$1.3 \cdot 10^1$	$1.1 \cdot 10^1$	$1.5 \cdot 10^2$
QJSK	$1.2 \cdot 10^1$	$1.4 \cdot 10^4$	$1.1 \cdot 10^2$	$1.6 \cdot 10^4$
QJSKT	$2.9 \cdot 10^1$	$1.5 \cdot 10^2$	$1.7 \cdot 10^2$	$1.4 \cdot 10^4$
SPGK	$0.1 \cdot 10^1$	$0.7 \cdot 10^1$	$0.1 \cdot 10^1$	$8.3 \cdot 10^1$
JSGK	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$
BRWK	$0.1 \cdot 10^1$	$8.6 \cdot 10^2$	$0.3 \cdot 10^1$	$4.1 \cdot 10^2$

WL method consists of the repeated propagation of the label information of a vertex to its neighbours, and when $h = 0$ the strengthened vertex labels are the original vertex labels. In our experiments, we set the largest value of h as 3, and compute the kernel matrix using each of the proposed kernels associated with all strengthened vertex labels by varying h from 0 to 3. Note that, when no vertex labels are available, each vertex is first labelled with its degree before applying the WL strengthening approach. Finally, note that since each graph of the MUTAG dataset has an original discrete edge label, the required directed edge label is computed as in Eq.(15). For the remaining datasets, the required directed edge label is computed as in Eq.(14).

We compare the performance of the proposed kernels $\mathrm{K_{DP}}$ and $\mathrm{K_{JS}}$ with that of several alternative state-of-the-art graph kernels. In addition to the kernels considered in the previous subsections, i.e., the WLSK [64], the JSGK [37], the QJSK [42], and the QJSKT [45], we consider the widely adopted shortest path graph kernel (SPGK) [26], and the backtrackless version of the random walk kernel (BRWK) [65]. For each dataset, we report the average classification accuracies ($\pm$ standard error) and the time usage of computing the kernel matrices for each kernel in Table III. The results are computed by performing a 10-fold cross-validation using a C-Support Vector Machine (C-SVM) to evaluate the classification accuracies of the different kernels. For each class, we used 90% of the samples for training and the remaining 10% for testing. The parameters of the C-SVMs are optimized separately on the training set for each dataset.

Table III shows that the proposed kernels outperform the alternative kernels on any dataset. Although our kernel measures are faster to compute than both the QJSK, the QJSKT, and the BRWK, the proposed kernel $\mathrm{K_{JS}}$ through the Jensen-Shannon divergence has a significantly higher runtime than the WLSK kernel and the JSGK kernel. However our kernels can still complete the computation in polynomial time, while yielding a better classification performance. This effectiveness is due to the fact that only the proposed kernels can accommodate both the original edge and vertex labels, e.g, on the MUTAG dataset. Moreover, only the proposed kernels can accommodate the edge weights through the commute time and reflect more graph characteristics than the alternative kernels.

VI. CONCLUSION

In this paper, we have proposed a new kernel measure for complete weighted graphs based on discrete-time quantum walks. Unlike existing state-of-the-art graph kernels, our kernel can accommodate complete weighted graphs, while at the same time overcoming the inefficiency and ineffectiveness of discrete-time quantum walks on these graphs. Experiments on time-varying financial networks abstracted from the New York Stock Exchange database as well as standard bionformatics graph datasets demonstrate the effectiveness of our kernel.

Our future work aims to extend the discrete-time quantum walk kernel for hypergraph-based time-varying financial networks. Ren et al. [66] have explored the use of the directed line graph representations for hypergraphs that can reflect richer high-order information than graphs. As we have stated, the discrete-time quantum walk can be seen as a walk evolving on directed line graphs. Thus, it would be interesting to extend these works by comparing the quantum walks on the directed line graphs associated with a pair of hypergraph-based time-varying financial networks.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant no. 61602535 and 61503422), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), and the program for innovation research in Central University of Finance and Economics. Corresponding author: Lixin Cui, E-mail: cuilixin@cufe.edu.cn.

REFERENCES

[1] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Trans. Cybernetics*, vol. 45, no. 3, pp. 430–443, 2015.
[2] X. Li, Z. Han, L. Wang, and H. Lu, "Visual tracking via random walks on graph model," *IEEE Trans. Cybernetics*, vol. 46, no. 9, pp. 2144–2155, 2016.
[3] J. Tang, L. Shao, X. Li, and K. Lu, "A local structural descriptor for image matching via normalized graph laplacian embedding," *IEEE Trans. Cybernetics*, vol. 46, no. 2, pp. 410–420, 2016.
[4] J. Xuan, J. Lu, G. Zhang, and X. Luo, "Topic model for graph mining," *IEEE Trans. Cybernetics*, vol. 45, no. 12, pp. 2792–2803, 2015.
[5] K. Riesen and H. Bunke, "Graph classification by means of lipschitz embedding," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 39, no. 6, pp. 1472–1483, 2009.
[6] A. Sanfeliu and K. Fu, "A distance measure between attributed relational graphs for pattern recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 13, no. 3, pp. 353–362, 1983.
[7] J. Zhang and M. Small, "Complex network from pseudoperiodic time series: Topology versus dynamics," *Physical Review Letters*, vol. 96, p. 238701, 2006.
[8] G. Nicolis, A. G. Cantu, and C. Nicolis, "Dynamical aspects of interaction networks," *International Journal of Bifurcation and Chaos*, vol. 15, p. 3467, 2005.
[9] Y. Shimada, T. Kimura, and T. Ikeguchi, "Analysis of chaotic dynamics using measures of the complex network theory," in *Proceedings of ICANN*, 2008, pp. 61–70.
[10] H. Bunke, P. J. Dickinson, A. Humm, C. Irniger, and M. Kraetzl, "Graph sequence visualisation and its application to computer network monitoring and abnormal event detection," in *Applied Graph Theory in Computer Vision and Pattern Recognition*, 2007, pp. 227–245.

[11] P. Shoubridge, M. Kraetzl, W. D. Wallis, and H. Bunke, "Detection of abnormal change in a time series of graphs," *Journal of Interconnection Networks*, vol. 3, no. 1-2, pp. 85–101, 2002.

[12] H. Bunke, P. J. Dickinson, and M. Kraetzl, "Comparison of two different prediction schemes for the analysis of time series of graphs," in *Proceedings of IbPRIA II*, 2005, pp. 99–106.

[13] H. Bunke, P. J. Dickinson, C. Irniger, and M. Kraetzl, "Analysis of time series of graphs: Prediction of node presence by means of decision tree learning," in *Proceedings of MLDM*, 2005, pp. 366–375.

[14] E. Bullmore and O. Sporns, "Complex brain networks: Graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.

[15] C. Ye, C. H. Comin, T. K. Peron, F. N. Silva, F. A. Rodrigues, L. da F. Costa, A. Torsello, and E. R. Hancock, "Thermodynamic characterization of networks using graph polynomials," *Physical Review E*, vol. 92, no. 3, p. 032810, 2015.

[16] T. Vỳrost, Š. Lyócsa, and E. Baumöhl, "Granger causality stock market networks: Temporal proximity and preferential attachment," *Physica A: Statistical Mechanics and its Applications*, vol. 427, pp. 262–276, 2015.

[17] L. Sandoval, "Structure of a global network of financial companies based on transfer entropy," *Entropy*, vol. 16, no. 8, pp. 4443–4482, 2014.

[18] F. N. Silva, C. H. Comin, T. K. Peron, F. A. Rodrigues, C. Ye, R. C. Wilson, E. R. Hancock, and L. da F. Costa, "Modular dynamics of financial market networks," *arXiv preprint arXiv:1501.05040*, 2015.

[19] D. P. Feldman and J. P. Crutchfield, "Measures of statistical complexity: Why?" *Physics Letters A*, vol. 238, no. 4, pp. 244–252, 1998.

[20] K. Anand, G. Bianconi, and S. Severini, "Shannon and von neumann entropy of random networks with heterogeneous expected degree," *Physical Review E*, vol. 83, no. 3, p. 036109, 2011.

[21] K. Anand, D. Krioukov, and G. Bianconi, "Entropy distribution and condensation in random networks with a given degree distribution," *Physical Review E*, vol. 89, no. 6, p. 062807, 2014.

[22] K. Huang, *Statistical Mechanic*. Wiley, New York, 1987.

[23] M. A. Javarone and G. Armano, "Quantum-classical transitions in complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2013, no. 04, p. 04019, 2013.

[24] J.-C. Delvenne and A.-S. Libert, "Centrality measures and thermodynamic formalism for complex networks," *Physical Review E*, vol. 83, no. 4, p. 046117, 2011.

[25] A. Fronczak, P. Fronczak, and J. A. Hołyst, "Thermodynamic forces, flows, and onsager coefficients in complex networks," *Physical Review E*, vol. 76, no. 6, p. 061106, 2007.

[26] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *Proceedings of the IEEE International Conference on Data Mining*, 2005, pp. 74–81.

[27] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Proceedings of COLT*, 2003, pp. 129–143.

[28] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," in *Proceedings of ICML*, 2003, pp. 321–328.

[29] L. Bai and E. R. Hancock, "Fast depth-based subgraph kernels for unattributed graphs," *Pattern Recognition*, vol. 50, pp. 233–245, 2016.

[30] N. Shervashidze, S. Vishwanathan, K. M. T. Petri, and K. M. Borgwardt, "Efficient graphlet kernels for large graph comparison," *Journal of Machine Learning Research*, vol. 5, pp. 488–495, 2009.

[31] L. Bai, L. Rossi, Z. Zhang, and E. R. Hancock, "An aligned subtree kernel for weighted graphs," in *Proceedings of ICML*, 2015, pp. 30–39.

[32] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," in *Proceedings of CVPR*, 2007.

[33] Y. Han, K. Yang, Y. Ma, and G. Liu, "Localized multiple kernel learning via sample-wise alternating optimization," *IEEE Trans. Cybernetics*, vol. 44, no. 1, pp. 137–148, 2014.

[34] X. Liu, L. Wang, J. Yin, E. Zhu, and J. Zhang, "An efficient approach to integrating radius information into multiple kernel learning," *IEEE Trans. Cybernetics*, vol. 43, no. 2, pp. 557–569, 2013.

[35] F. D. Johansson, V. Jethava, D. P. Dubhashi, and C. Bhattacharyya, "Global graph kernels using geometric embeddings," in *Proceedings of ICML*, 2014, pp. 694–702.

[36] L. Xu, X. Niu, J. Xie, A. Abel, and B. Luo, "A local–global mixed kernel with reproducing property," *Neurocomputing*, vol. 168, pp. 190–199, 2015.

[37] L. Bai and E. R. Hancock, "Graph kernels from the jensen-shannon divergence," *Journal of Mathematical Imaging and Vision*, vol. 47, no. 1-2, pp. 60–69, 2013.

[38] E. Farhi and S. Gutmann, "Quantum computation and decision trees," *Physical Review A*, vol. 58, p. 915, 1998.

[39] L. Bai, L. Rossi, A. Torsello, and E. R. Hancock, "A quantum jensen-shannon graph kernel for unattributed graphs," *Pattern Recognition*, vol. 48, no. 2, pp. 344–355, 2015.

[40] L. Rossi, A. Torsello, and E. R. Hancock, "Measuring graph similarity through continuous-time quantum walks and the quantum jensen-shannon divergence," *Physical Review E*, vol. 91, no. 2, p. 022815, 2015.

[41] 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.

[42] L. Bai, L. Rossi, P. Ren, Z. Zhang, and E. R. Hancock, "A quantum jensen-shannon graph kernel using discrete-time quantum walks," in *Proceedings of GbRPR*, 2015, pp. 252–261.

[43] R. N. Mantegna and H. E. Stanley, *Introduction to econophysics: correlations and complexity in finance*. Cambridge university press, 1999.

[44] H. Qiu and E. R. Hancock, "Clustering and embedding using commute times," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1873–1890, 2007.

[45] L. Bai, L. Rossi, L. Cui, Z. Zhang, P. Ren, X. Bai, and E. R. Hancock, "Quantum kernels for unattributed graphs using discrete-time quantum walks," *Pattern Recognition Letters*, vol. 87, pp. 96–103, 2017.

[46] H. Qiu and E. R. Hancock, "Graph simplification and matching using commute times," *Pattern Recognition*, vol. 40, no. 10, pp. 2874–2889, 2007.

[47] P. Ren, T. Aleksic, D. Emms, R. C. Wilson, and E. R. Hancock, "Quantum walks, ihara zeta functions and cospectrality in regular graphs," *Quantum Information Process*, vol. 10, pp. 405–417, 2011.

[48] A. Majtey, P. Lamberti, and D. Prato, "Jensen-shannon divergence as a measure of distinguishability between mixed quantum states," *Physical Review A*, vol. 72, p. 052310, 2005.

[49] L. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of ACM Symposium on the Theory of Computation*, 1996, pp. 212–219.

[50] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," in *Proceedings of the American Mathematical Society*, 1956, pp. 7:48–50.

[51] I. Fischer and I. Fischer, "Amplifying the block matrix structure for spectral clustering," in *In Proceedings of IDSIA*, 2005, pp. 21–28.

[52] D. Levin, Y. Peres, and y. . . E.L. Wilmer title = Markov Chains and Mixing Times, journal = American Mathematical Society.

[53] A. F. Martins, N. A. Smith, E. P. Xing, P. M. Aguiar, and M. A. Figueiredo, "Nonextensive information theoretic kernels on measures," *Journal of Machine Learning Research*, vol. 10, pp. 935–975, 2009.

[54] K. Chung, W. Kao, C. Sun, L. Wang, and C. Lin, "Radius margin bounds for support vector machines with the RBF kernel," *Neural Computation*, vol. 15, no. 11, pp. 2643–2681, 2003.

[55] M. R. Hajiaboli, M. O. Ahmad, and C. Wang, "An edge-adapting laplacian kernel for nonlinear diffusion filters," *IEEE Trans. Image Processing*, vol. 21, no. 4, pp. 1561–1572, 2012.

[56] L. Bai, Z. Zhang, P. Ren, L. Rossi, and E. R. Hancock, "An edge-based matching kernel through discrete-time quantum walks," in *Processings of ICIAP I*, 2015, pp. 27–38.

[57] H. Fröhlich, J. K. Wegner, F. Sieker, and A. Zell, "Optimal assignment kernels for attributed molecular graphs," in *Proceedings of ICML*, 2005, pp. 225–232.

[58] E. S. Browning, "Exorcising ghosts of octobers past," *The Wall Street Journal*, pp. C1–C2, 2007.

[59] D. Jenkins, *Handbook of Airline Economics*. Aviation Week A Division of McGraw-Hill, New York, 2002.

[60] C. Mollenkamp, S. Craig, S. Ng, and A. Lucchetti, "Lehman files for bankruptcy, merrill sold, aig seeks cash," *The Wall Street Journal*, 2008.

[61] K. Anderson, C. Brooks, and A. Katsaris, "Speculative bubbles in the s&p 500: Was the tech bubble confined to the tech sector?" *Journal of empirical finance*, vol. 17, no. 3, pp. 345–361, 2010.

[62] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[63] R. Kondor and H. Pan, "The multiscale laplacian graph kernel," in *Proceedings of NIPS*, 2016, pp. 2982–2990.

[64] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2010.

[65] 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.

[66] P. Ren, T. Aleksic, R. C. Wilson, and E. R. Hancock, "A polynomial characterization of hypergraphs using the ihara zeta function," *Pattern Recognition*, vol. 44, no. 9, pp. 1941–1957, 2011.