A Wave Packet Signature for Complex Networks

FURQAN AZIZ*,
Center for Excellence in Information Technology, IMSciences, Peshawar, Pakistan
*furqan.aziz@imsciences.edu.pk
RICHARD C. WILSON
Department of Computer Science, University of York, YO10 5GH UK
richard.wilson@york.ac.uk

AND

EDWIN R. HANCOCK
Department of Computer Science, University of York, YO10 5GH UK
edwin.hancock@york.ac.uk

[Received on 30 July 2018]

This paper explores the possibility of using a quantum graph representation to investigate information flow across a complex network in the form of wave propagation. The term quantum graph refers to a representation of a network with differential operators acting on functions defined over real valued intervals associated with the edges. These provide a convenient representation which allows differential operators from calculus to be generalized to graph or network structures, the simplest of which is the metric Laplacian. We present complete solutions of a wave equation on a quantum graph, obtained using the quantum graph Laplacian. The solutions obtained using the Laplacian are more complex than those obtained from the traditional vertex-based graph Laplacian counterpart due to the more complex structure of the eigenfunctions. Specifically, we provide explicit solutions for the wave equation where the initial condition is a Gaussian wave packet confined to a single edge of the graph. We use the new solutions to develop methods for characterising complex networks, and demonstrate their advantages over those obtained using the discrete graph Laplacian. The proposed method is highly robust in distinguishing networks that are co-spectral with respect to different graph representations. Moreover, the partial differential equations defined using the quantum graph Laplacian may be of great interest in the study of complex networks where distance, speed of propagation, and connectivity structure of the networks are important.

Keywords: Quantum graphs, Edge-based Laplacian, Graph characterization.

1. Introduction

In the last decade, the field of Complex Networks has emerged as an inter-disciplinary branch of research that combines concepts and ideas from various diverse fields including Mathematics, Physics, Social Science, and Computer Science [16, 40]. The focus of complex network theory is to represent and analyse complex systems that are often very large and dynamic. Such systems can be conveniently represented by means of a complex network where the components are represented by nodes while the interactions between them are represented by edges. This simple yet powerful representation can model a number of real world systems ranging from the very simple to the very complex. For example, a chemical compound can be represented with the help of a small network where vertices represent atoms and edges represent bonds between the atoms [59]. On the other hand, a text document can be represented as a large network, where paragraphs are represented as vertices and two vertices are connected whenever they share a minimum semantic content [27]. In each of these scenarios, the complex network representation helps us to uncover the properties of the underlying complex system. It also allows us to understand numerous phenomena associated with the system. Examples include the spread of disease in epidemiology and the nature of protein protein interactions in bioiformatics. The analysis of network structure has therefore become an important way not only of representing the structure of a system but also of understanding its function or changes in function too.

One of key elements in the study of complex network structure and function, is that of concise network characterization. The aim in here is to establish a method that can be used to compare and distinguish the structural detail of different networks. This can be achieved either by extracting salient features from the networks and then measuring the similarity of the features [3] or by directly computing a measure of similarity such as edit distance [12]. Once measures of similarity are to hand, then different classes of graph structure can be identified via graph clustering [62] [32]. One way to compute the similarity of a pair of networks is to apply subgraph isomorphism [49]. However subgraph isomorphism is an NP-complete problem, and hence the exact solution is computationally intractable. To cope with this situation, several approximate graph characterization algorithms have been developed [56]. Most of these procedures are based on the occurrence frequencies of substructures within the graph, and some well known techniques are random walks [33], backtrackless walks and prime cycles [3] and shortest paths [10].

The way information propagates in a network can reveal interesting properties about its structure. As a result random walks or equivalently diffusions have proved to provide convenient and powerful ways of understanding the properties of the graph [69]. Most of these properties can be understood through the analysis of the heat equation on the graph. So, for instance, Xiao et al. [68] have used the solution of heat equation to embed the nodes of the graphs into an Euclidean space and have used the resulting embedding coordinates as a geometric graph characterization. Escolano et al. [21] have used a heat diffusion process to gauge the complexity of a graph. Wang et al. [64] have explored the use of different thermodynamic characterizations to distinguish different network structures. Recently Sun et al. [60] have used the solution of a heat equation to define signature to analyse a three-dimensional shape and this is referred to as Heat Kernel Signature (HKS). Another physically motivated signature is the Wave Kernel Signature (WKS), which was proposed by Aubry et al. [1], and which uses wave like solutions. The WKS was proposed as a solution to the excessive sensitivity of the HKS to low frequency information. The WKS is the solution of Schrödinger equation [48, 55] and it represents the average probability of measuring a quantum mechanical particle at a specific location at different time intervals.

A critical issue concerning modelling of information propagation is the nature of the signal being studied [38]. Most of the work on the use of the heat equation, relies on studying the dissipation of a static initial distribution. In many problems though, a better model is of an information packet that propagates on a network. For instance, in the case of the wave equation, it would be interesting to study the propagation of wave-packet, rather than a single wave, since this may better model the transient nature of initial signal. Moreover, the study of wave packets also opens the interesting topic of what information the existence of solitons (i.e., the non dissipative solutions of the wave equation) of a network, might convey [41]. Finally, the ability study of both diffusion and wave propagation with inhomogeneous speed would also be interesting to models networks, where there is a differential rate of information propagation.

Unfortunately, the reason why these topics have not been studied in detail is that the requisite mathematical apparatus had not been available. To study the flow of information in a network, we need to implement partial differential equations that accurately model the propagation processes. Traditionally, for instance, diffusion processes have been implemented using the discrete Laplacian. The discrete Laplacian is an approximation to the continuous Laplacian that is appropriate when the data is to be sampled at finitely many points. In graph theory, the discrete Laplacian is an approximation of the continuous Laplacian over the vertices of the graph, and can be used to simulate a diffusion process on these vertices. It is defined as the diagonal degree matrix minus the adjacency matrix of the graph. As noted above, perhaps the most widely used example is the heat equation, which is ultimately related to the continuous time random walk on a graph [15]. However, this simple model does not capture features such as differential speed or packet propagation, since it is confined to the nodes and does not capture the details of propagation along the edges of a graph. Similar criticisms can be directed at the documented attempts to model wave propagation on a graph [1].

One way to overcome these problems is to use the apparatus furnished by quantum graphs [34, 37, 45, 46]. The term quantum graph refers to a metric graph, with a real-valued interval associated with each edge and a differential operator defined over these intervals. Functions may, therefore, exist on both the edges and the vertices. Quantum graph theory is a relatively new and exciting area of network science and has recently attracted significant attention [9, 24]. One of the interesting problems studied in this regard was the problem of community detection in quantum complex network by Faccin et al. [26] . They have demonstrated that certain quantum mechanical effects cannot be captured using current classical complex network tools and provide new methods that overcome these problems. They have tested their method on simple quantum networks and biological networks. In a related work [25] they have studied and analyzed quantum walks on complex networks which model network-based processes on complex networks. Cuquet et al. [17] have studied the effect of entanglement percolation as a means to establish long-distance interaction between arbitrary nodes of quantum complex networks.

The simplest differential operator on the edges of a quantum graph is the Laplace operator, which is simply the second derivative of the function with respect to the edge variables. This allows us to define a number of functions on graphs [53, 54]. This has numerous applications. As an example, it is possible to define a wave equation [30] which has a finite speed of propagation [44], in contrast to the node-based wave equation on a graph in which propagation along edges is instantaneous. The origin of quantum graphs can be traced back to the 1930's, and the analysis of free electrons in large organic molecules. They have been used in the analysis of a number of physical problems including quantum chaos and photonics waveguides, and most recently, in nanophysics. Whereas quantum graphs have been extensively studied in the analysis of physical systems, much attention has been focussed on spectral analysis [13, 23] and special cases. The detailed solution of differential equations and the analysis of their eigenfunctions has received less attention. In [29, 30], Friedman and Tillich provide partial solutions for the eigensystem of a quantum graph, but their analysis of the eigenfunctions is incomplete. However, in a recent paper [66] we have completed this analysis, and this opens up the possibility of solving a much richer family of differential equations on graphs.

In order to provide an initial proof of concept for the advantages of the edge-based Laplacian, in [2, 4] we proposed a graph signature based on the solution of a wave equation defined using the edge-based Laplacian. We have experimentally shown that the resulting signature is effective in characterizing both weighted and unweighted networks [2]. In this paper, on the other hand, we are proposing a similar measure that uses the solution of edge-based wave equation to characterize graphs. We explore the properties of the proposed measure and compare its performance with alternate techniques in the literature [1, 43, 56, 60, 63, 67]. We also show that the proposed method can be effectively used to distinguish between structurally modified graphs as well as non-isomorphic co-spectral graphs. This suggests that the work presented here has the potential to accurately characterize the shape of a graph [37]. Our novel contributions in the paper are as follows.

- We present complete solutions to the wave equation defined using the **edge-based Laplacian** for different types of graphs including trees, cycles and bipartite graphs. We assume that the initial condition is a single Gaussian wave packet residing on the edge with highest betweenness centrality [28] and define a novel signature based on the amplitudes of the waves over time.
- We also highlight the problems associated with infinite propagation speed for wave equation defined using the discrete Laplacian. We show why wave equation defined using the edge-based Laplacian on a quantum graph provides a more powerful tool for characterising graphs.
- In order to investigate the properties of the proposed shape signature, we apply it to different types of randomly generated complex networks. This allows us to study how the shape of the proposed signature varies for different types of complex networks. We also demonstrate that the WPS is robust to structural modifications to the graph.
- Finally, we compare the proposed method with some state-of-the-art methods including the heat kernel signature and the wave kernel signature on different datasets. We report the resulting classification accuracies.

One of the advantages of our method is that it can resolve differences between co-spectral graphs which are structurally different, yet have the same Laplacian spectrum. This is is a long-standing problem not only in the graph and network theory, but has links with Kac's classical work "Can one hear the Shape of A Drum". This latter problem remained open until 1993, when Gordon, Webb and Wolpert [36] used spectral geometry [61] to show that different shapes can have identical eigenvalues. More recently, interest has focussed on quantum graphs and under what conditions different graphs give rise to Schrödinger operators which share the same spectrum [5, 6, 37]. To that end, Gutkin [37] et al. have shown that the spectrum of a Schrödinger operator on a quantum graph determines uniquely the connectivity matrix and the bond lengths. Similarly, Kurasov [48] has studied the relation between the spectrum of a Schrödinger operator of a Quantum graph and geometric properties of the graph. Our empirical work here suggests that the wave packet signatures explored in this paper can distinguish structures that are co-spectral under the Laplacian. This may suggest a new line of theoretical work aimed at understanding more deeply how wave-packets propagate on quantum graphs and their links to isospectality.

From the perspective of complex networks our work has a number of potential uses. Our new framework allows the propagation of wave-packets on a graph to be modelled and characterised. From the complex network perspective, this can lead to new methods for the characterisation of network centrality and communicability on networks [8, 69]. For instance, recently there has been interest in how to characterise network centrality using quantum walks[58] rather than classical ones [22]. This is based on the complex wave equation (Schrödinger's equation). Our analysis allows these ideas to be extended to where there is weight or distance function on the edges, extending the method to geometric graphs. From the perspective of communicability our method offers a number of potential extensions of the existing theory. First, it allows the propagation and time evolution of information packets of finite width to be analysed, and their transmission on different types of structure understood. It also allows the propagation of information on geometric graphs to be modelled. This latter point may allow more detailed models to be constructed in the epidemiology and communication network domains.

The remainder of the paper is organized as follows. In Section 2, we introduce both graphs and quantum graphs together with some basic definitions that will be used throughout the paper. For the sake of completeness, in Section 3, we define the eigensystem of the edge-based Laplacian and present method for its computation. Section 4 provides a general solution of a wave equation for different types of graphs, initialized with a Gaussian wave packet confined to a single edge of the graph. Section 5 compares the discrete and the continuous solutions on a simple graph, and demonstrates the limitations of the discrete solutions. In Section 6, we define signatures to characterize graphs and explore some of their properties on different type of complex networks. In section 7, we have applied the proposed method to real-world graph datasets and we report the resulting accuracies. Finally in Section 8, we give some conclusions of our work and suggest future directions for research.

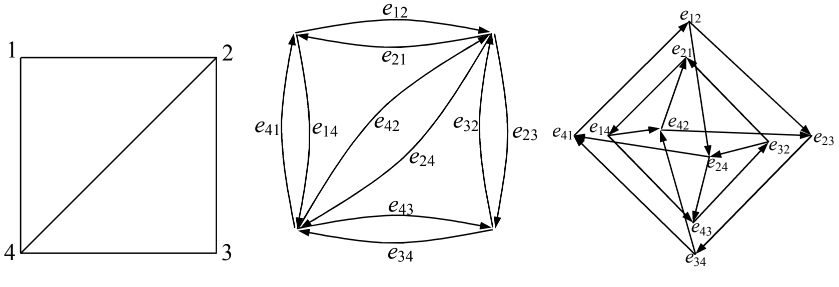
2. Graphs

A *graph* $G=(V,E)$ consists of a finite nonempty set V of *vertices* and a finite set E of unordered pairs of vertices, called *edges*. A *directed graph* or a *digraph* $D=(V_D,E_D)$ consists of a finite nonempty set V_D of vertices and a finite set E_D of ordered pairs of vertices, called *arcs*. So a digraph is a graph with an orientation on each edge. A digraph D is called *symmetric* if whenever (u,v) is an arc of D, then (v,u) is also an arc of D. There is a one-to-one correspondence between the set of symmetric digraphs and the set of graphs, given by identifying an edge of the graph with an arc and its inverse arc on the digraph on the same set of vertices. We denote by $D(G)$ the symmetric digraph associated with the graph G.

The *oriented line graph* $OL(G)=(V_O;E_O)$, where V_O and E_O are respectively the set of vertices and edges of the oriented line graph, is constructed by replacing each arc of $D(G)$ by a vertex. These vertices are connected if the head of one arc meets the tail of another except that reverse pairs of arcs are not connected, i.e. $((u,v),(v,u))$ is not an edge. The vertex and edge sets of $OL(G)$ are therefore

$$\begin{aligned} V_O &= \{(u,v)\in D(G)\}, \\ E_O &= \{((u,v),(v,w)) : (u,v),(v,w)\in D(G), u\neq w\} \end{aligned}$$

Figure 1(a) shows a simple graph, 1(b) its digraph, and 1(c) the corresponding oriented line graph.



(a) Graph (b) Digraph (c) Oriented line graph

FIG. 1. Graph, its digraph, and its oriented line graph.

A *walk* w in a graph is a sequence of vertices $v_1,v_2,...v_k$ where $v_i\in V$ such that v_i and v_{i+1} are adjacent. A walk has *backtracking* if $v_{i-1}=v_{i+1}$, for some i, $2\leqslant i\leqslant k-1$, where k is the length of the walk. A walk is *backtrackless* if it has no backtracking. A random walk on the vertices of the oriented line graph represents a backtrackless walk.

2.1 *Graph representations*

A graph $G = (V,E)$ is usually represented by a $|V| \times |V|$ adjacency matrix, whose elements are given as

$$A(i,j) = \begin{cases} 1, & \text{if } (i,j) \in E; \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Laplacian matrix is another useful representation of a graph. It is defined as $L = D - A$, where D is the diagonal matrix whose i^{th} diagonal entry represents the degrees of the i^{th} vetex. The Laplacian matrix can be used to find many useful properties of a graph. It is sometimes also referred to as discrete Laplacian, since it is an approximation of the continuous Laplacian to the vertices of a graph. Other useful graph representations include the *normalized Laplacian* which is defined as $\hat{L} = D^{-1/2}LD^{-1/2}$.

2.2 *Metric graph*

A graph $G = (V,E)$, with a (possibly empty) boundary set $\partial G \subset G$, is said to be a *metric graph*, if it has a geometric realization $\mathscr{G}$ [47]. A geometric realization is the metric space consisting of vertices V with a closed interval of length l_e associated with each edge $e \in E$. We associate an edge variable x_e with each edge $e = (u,v)$ that represents the standard coordinate on that edge with $x_e(u) = 0$ and $x_e(v) = 1$. The start and end vertices are determined by assigning an arbitrary orientation to each edge. Note that in this paper we will focus on metric graphs in which the edge lengths are all equal. The results can naturally be extended to graphs with any rational edge lengths by simple subdivision of the edges thereby adding nodes.

Definition 1 (Kuchment [45]) A **quantum graph** is a metric graph equipped with the operator $\mathscr{H}$ (Hamiltonian) that acts as the negative second order derivative along edges and satisfies some suitable boundary conditions at the vertices.

Let f be a square integrable function defined on the graph (on both edges and vertices). We take $f(u)$ to mean the value of f at vertex u and $f(e,x_e)$ to mean the value of f at position x_e along edge e. Here we choose the Hamiltonian to be the usual continuous Laplacian $-\Delta f$ along the edges.

Definition 2 **Neumann-Kirchhoff boundary conditions:** A function satisfies the Neumann-Kirchhoff boundary conditions if

$$\sum_{e \ni v} (-1)^{1-x_{e,v}} \nabla f(e,x_{e,v}) = 0, \forall v, \tag{2.2}$$

which means that the sum of the outward-pointing gradients must be zero. Here $x_{e,v}$ represents the end point of the edge e incident on vertex v.

We call such a setup *edge-based* since there is no contribution to the Laplacian from the vertices. In particular, the Laplacian defined on a metric graph satisfied Neumann-Kirchhoff boundary conditions is referred to as the edge-based Laplacian.

3. Edge-based Eigensystem

In this section we review what is known about the eigensystem of the edge-based Laplacian[30][66]. To commence, let $G = (V,E)$ be a graph with a boundary ∂G and let $\mathscr{G}$ be the geometric realization of G. The eigenpairs of the edge-based Laplacian are of two types, i.e., vertex supported eigenpairs and edge-interior eigenpairs.

3.1 *Vertex-supported Eigenfunctions*

The vertex-supported eigenpairs of the edge-based Laplacian can be expressed in terms of the eigenpairs of the normalized adjacency matrix of the graph. Let A be the adjacency matrix of the graph G, and $\tilde{A}$ be the row-normalized adjacency matrix, i.e., the $(i,j)^{\text{th}}$ entry of $\tilde{A}$ is given as

$$\tilde{A}(i,j) = \frac{A(i,j)}{\sum_{(k,j)\in E} A(k,j)}.$$

Let $(g(v),\lambda)$ be an eigenvector-eigenvalue pair for this matrix. Note that g is defined on vertices and may be extended along each edge to an edge-based eigenfunction. Let ω^2 and $\phi(e,x_e)$ denote the edge-based eigenvalue and eigenfunction respectively. Then the vertex-supported eigenpairs of the edge-based Laplacian are given as follows:

1. For each $(g(v),\lambda)$ with $\lambda \neq \pm 1$, we have an eigenvalue ω^2 with $\omega = \cos^{-1}\lambda$. Since there are multiple solutions to the equation $\omega = \cos^{-1}\lambda$, we obtain an infinite sequence of eigenfunctions. If $\omega_0 \in [0,\pi]$ is the principal solution, the eigenvalues are $\omega = \omega_0 + 2\pi n, n \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers. The corresponding eigenfunctions are $\phi(e,x_e) = \pm C(e,\omega)\cos(B(e,\omega)+\omega x_e)$ where

$$C(e,\omega)^2 = \frac{g(v)^2 + g(u)^2 - 2g(v)g(u)\cos(\omega)}{\sin^2(\omega)},$$

and

$$\tan(B(e,\omega)) = \frac{g(v)\cos(\omega) - g(u)}{g(v)\sin(\omega)}.$$

There are two solutions here, $\{C, B_0\}$ or $\{-C, B_0+\pi\}$ but both give the same eigenfunction. The sign of $C(e,\omega)$ must be chosen correctly to match the phase.

2. $\lambda = 1$ is always an eigenvalue of $\tilde{A}$. For this eigenvalue, we obtain a principal frequency $\omega = 0$, and since $\phi(e,x_e) = C\cos(2n\pi x_e)$, therefore $\phi(e,v) = \phi(e,u) = C$, which means that the eigenfunction is constant on the vertices.

3. $\lambda = -1$ is an eigenvalue of $\tilde{A}$, if and only if the graph is bipartite. For this eigenvalue, we obtain a principal frequency $\omega = \pi$, and since $\phi(e,x_e) = C_B\cos(\pi x_e + 2n\pi x_e)$, therefore $\phi(e,v) = C_B$ and $\phi(e,u) = -C_B$, which means that the eigenfunction is constant on the vertices, with an alternating sign on both sides of a bipartition.

3.2 *Edge-interior eigenfunctions*

The edge-interior eigenfunctions are those eigenfunctions which are zero on vertices and must therefore have a principal frequency of $\omega \in \{\pi, 2\pi\}$. These eigenfunctions can be determined from the eigenvectors of the adjacency matrix of the oriented line graph [66]. In particular

1. The eigenvector corresponding to the eigenvalue $\lambda = 1$ of the oriented line graph provides a solution in the case $\omega = 2\pi$, and we obtain $|E| - |V| + 1$ linearly independent solutions. Since there are potentially multiple solutions, we denote the constants associated with each solution as $C_{E_{2\pi}}(e,i)$.

2. Similarly, the eigenvector corresponding to the eigenvalue $\lambda = -1$ of the oriented line graph provides a solution in the case $\omega = \pi$. If the graph is bipartite, then we obtain $|E| - |V| + 1$ linearly independent solutions. If the graph is non-bipartite, then we obtain $|E| - |V|$ linearly independent solutions with constants $C_{E_\pi}(e,i)$.

Hence the structure of the edge-based eigenfunctions is captured both by the random walks and backtrackless random walks on a graph. This comprises all of the principal eigenpairs which are only supported on the edges.

3.3 *Normalization of the eigenfunctions*

Note that although these eigenfunctions are orthogonal, they are not normalized. To normalize the eigenfunctions we need to determine the normalization factor corresponding to each eigenvalue. Let $\rho(\omega)$ denote the normalization factor corresponding to the eigenvalue ω. Then

$$\rho^2(\omega) = \sum_{e \in E} \int_0^1 \phi^2(e,x_e)\,dx_e.$$

Evaluating the integral, we get

$$\rho(\omega) = \sqrt{\sum_{e \in \mathscr{E}} C(e,\omega)^2 \left[\frac{1}{2} + \frac{\sin(2\omega + 2B(e,\omega))}{4\omega} - \frac{\sin(2B(e,\omega))}{4\omega}\right]}.$$

Once we have the normalization factor in hand, we can compute a complete set of orthonormal basis by dividing each eigenfunction by the corresponding normalization factor.

Note that the constant eigenfunctions $\phi(e,x_e) = C\cos(2n\pi x_e)$ corresponding to the principal frequency $\omega = 0$ are different for the cases when $n = 0$ and when $n > 0$. When $n = 0$, these eigenfunctions are constant on the vertices as well as on the edges. Therefore, in this case

$$C = \sqrt{\frac{1}{|E|}},\ \text{since} \sum_E \int_0^1 \left(\frac{1}{\sqrt{E}}\right)^2 dx_e = 1.$$

On the other hand, when $n > 0$, the eigenfunctions corresponding to the principal frequency $\omega = 0$ are constant on the vertices, but not on edges. They take the form $\phi(e,x_e) = C\cos(2n\pi x_e)$ on the edges. These eigenfunctions must be normalized, so

$$C = \sqrt{\frac{2}{E}},\ \text{since} \sum_E \int_0^1 \left(\sqrt{\frac{2}{E}}\cos(2\pi n x_e)\right)^2 dx_e = 1.$$

Once normalized, these eigenfunctions form a complete set of orthonormal basis for $L^2(G,E)$ [30], where $L^2(G,E)$ is the set of all square integrable functions over the metric graph G.

4. Wave equation on a graph

In this section we provide a general solution of a wave equation on a graph, where the initial condition is a single wave packet normally distributed over an arbitrary edge of the graph. The edge-based wave equation on the graph is defined as

$$\frac{\partial^2}{\partial t^2} u(\mathscr{X},t) = \Delta_E u(\mathscr{X},t), \tag{4.1}$$

$u_1(\mathscr{X},t)$	$\sum_{\omega\in\Omega_a}\frac{C(e,\omega)C(f,\omega)}{2}\left(\exp\left[-a\mathscr{W}(x+t+\mu)^2\right]\cos\left[B(e,\omega)+B(f,\omega)+\omega\left\lfloor x+t+\mu+\frac{1}{2}\right\rfloor\right]\right.$ $\left.+\exp\left[-a\mathscr{W}(x-t-\mu)^2\right]\cos\left[B(e,\omega)-B(f,\omega)+\omega\left\lfloor x-t-\mu+\frac{1}{2}\right\rfloor\right]\right)$
$u_2(\mathscr{X},t)$	$\frac{1}{2\lvert E\rvert}\left(\exp\left[-a\mathscr{W}(x+t+\mu)^2\right]+\exp\left[-a\mathscr{W}(x-t-\mu)^2\right]\right)$
$u_3(\mathscr{X},t)$	$\frac{C_B^2}{4}\left((-1)^{\left\lfloor x-t-\mu+\frac{1}{2}\right\rfloor}\exp\left[-a\mathscr{W}(x-t-\mu)^2\right]+(-1)^{\left\lfloor x+t+\mu+\frac{1}{2}\right\rfloor}\exp\left[-a\mathscr{W}(x+t+\mu)^2\right]\right)$
$u_4(\mathscr{X},t)$	$\sum_i\frac{C_{E\pi}(e,i)C_{E\pi}(f,i)}{4}\left(\exp\left[-a\mathscr{W}(x-t-\mu)^2\right]-\exp\left[-a\mathscr{W}(x+t+\mu)^2\right]\right)$
$u_5(\mathscr{X},t)$	$\sum_i\frac{C_{E_{2\pi}}(e,i)C_{E_{2\pi}}(f,i)}{4}\left((-1)^{\left\lfloor x-t-\mu+\frac{1}{2}\right\rfloor}\exp\left[-a\mathscr{W}(x-t-\mu)^2\right]\right.$ $\left.-(-1)^{\left\lfloor x+t+\mu+\frac{1}{2}\right\rfloor}\exp\left[-a\mathscr{W}(x+t+\mu)^2\right]\right)$

Table 1. LIST OF SOLUTIONS

where Δ_E is the edge-based Laplacian. To obtain a general solution of the above equation, we need to solve it for all possible cases, i.e., for all values of ω and n. A complete solution for all these cases is given in the Table 1. Here Ω_a represents the set of vertex-supported eigenvalues. The second term, $u_2(\mathscr{X},t)$, comes from the constant eigenvalue $\omega=0$, while the third term, $u_3(\mathscr{X},t)$, corresponds to the constant eigenvalue $\omega=\pi$. The last two terms, i.e., $u_4(\mathscr{X},t)$ and $u_5(\mathscr{X},t)$, correspond to the edge-interior eigenpairs. See Appendix for a complete analytical solution of the wave equation.

The exact solution of the wave equation depends both upon whether the graph being considered is bipartite or not, and upon the number of edges and vertices in the graph. Here we give solutions for some of the special cases of graph structures.

1. For a bipartite graph with $|E|>|V|$, the solution is

$$\begin{aligned}u(\mathscr{X},t)=u_1(\mathscr{X},t)+u_2(\mathscr{X},t)+u_3(\mathscr{X},t)\\+u_4(\mathscr{X},t)+u_5(\mathscr{X},t).\end{aligned}$$

   Here u_3 is due to the constant eigenfunction corresponding to the principal frequency $\omega=\pi$.

2. For a non-bipartite graph with $|E|>|V|$, the solution is

$$u(\mathscr{X},t)=u_1(\mathscr{X},t)+u_2(\mathscr{X},t)+u_4(\mathscr{X},t)+u_5(\mathscr{X},t),$$

   and so the term u_3 is omitted, because a non-bipartite graph has no principal frequency $\omega=\pi$ and $B\neq 0$.

3. Note that the presence of the terms u_4 and u_5 depends on the number of edges and vertices in the graph, and also upon whether the graph is bipartite or not. So, for a graph which is a cycle of odd length, the solution is

$$u(\mathscr{X},t)=u_1(\mathscr{X},t)+u_2(\mathscr{X},t)+u_5(\mathscr{X},t),$$

   while for a cycle of even length, the solution is

$$\begin{aligned}u(\mathscr{X},t)=u_1(\mathscr{X},t)+u_2(\mathscr{X},t)+u_3(\mathscr{X},t)\\+u_4(\mathscr{X},t)+u_5(\mathscr{X},t).\end{aligned}$$

4. For a tree (which is always bipartite), we have

$$u(\mathscr{X},t) = u_1(\mathscr{X},t) + u_2(\mathscr{X},t) + u_3(\mathscr{X},t), \tag{4.2}$$

i.e., both u_4 and u_5 are omitted. This is because for a tree $|E| = |V| - 1$, and so the multiplicity of the edge-interior eigenfunctions is zero.

5. Discrete vs. continuous solutions

In this section we demonstrate the limitations of the wave equation that is defined using the discrete Laplacian. We will show why the partial differential equations defined using the edge-based Laplacian are more powerful for simulating a diffusion process on a graph than those resulting from the discrete Laplacian. For this purpose, we study the discrete version of the wave equation. The vertex-based wave equation is defined as

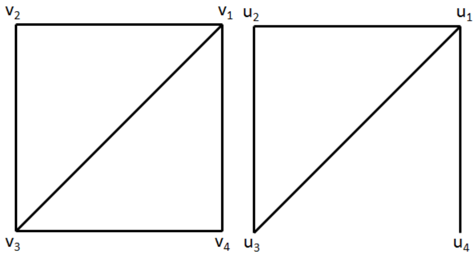
$$\frac{\partial^2}{\partial t^2} u(x,t) = -\Delta_V u(x,t), \tag{5.1}$$

For any edgewise linear function f, the solution of the discrete wave equation is

$$u(.,t) = \cos\left(t\sqrt{\Delta_V}\right) f = f - t^2 \Delta_V f/2! + t^4 \Delta_V^2 f/4! - ...,$$

which satisfies the wave equation with $u(x,0) = f(x)$.

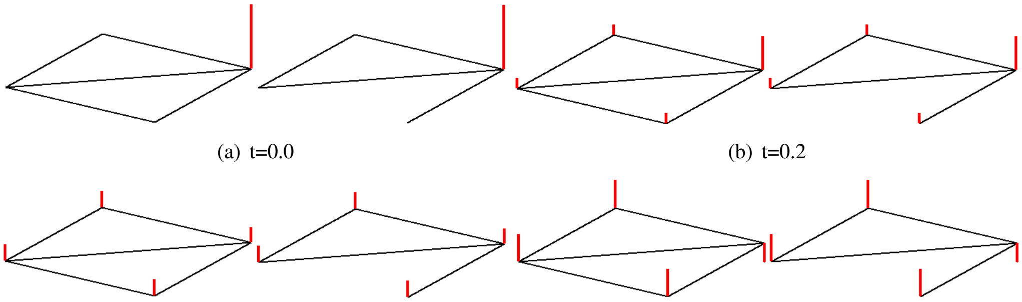
As mentioned earlier, one of the problems with partial differential equations defined using the discrete Laplacian is the speed of propagation. For example, the wave equation defined using the discrete Laplacian does not have finite speed of propagation [30]. This reduces the power of the partial differential equation to distinguish between graphs with different structures. To demonstrate this, we choose two different graphs G_1 and G_2, with 4 vertices each. The number of edges in G_1 and G_2 are 5 and 4 respectively. Figure 2 shows these graphs.



(a) G1 (b) G2

FIG. 2. Two simple graphs with same number of vertices but different number of edges.

We simulate the evolution of a discrete wave equation. Here we assume the initial condition is a unit wave on a single vertex of the graph (in this case v_1 for G_1 and u_1 for G_2). Figure 3 shows the evolution of wave amplitudes with time for both G_1 and G_2. The figure shows that the wave equation fails to distinguish the two graphs.



(a) t=0.0

(b) t=0.2

(c) t=0.4

(d) t=1.2

FIG. 3. Discrete wave equation for two different graphs.

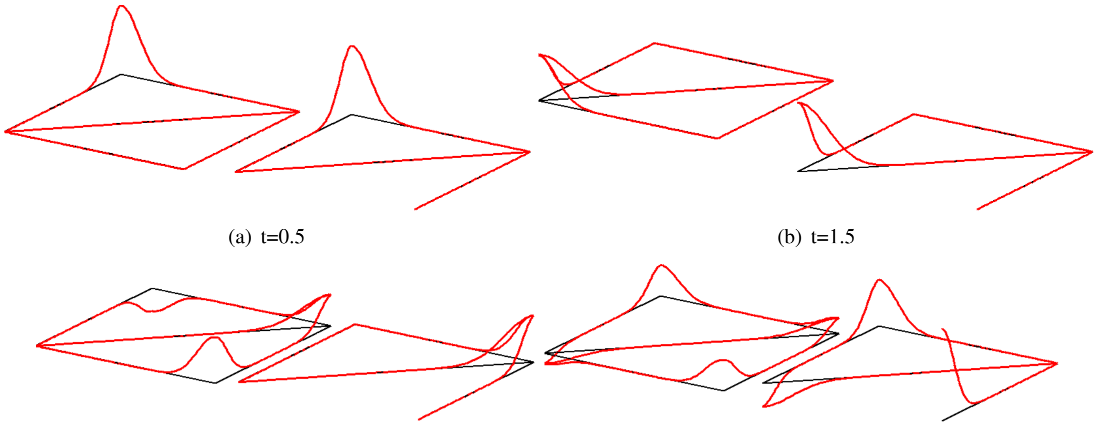
Table 2. Amplitude of waves on different vertices of the graphs.

t	v_1	v_2	v_3	v_4
0.0	1.0000	0.0000	0.0000	0.0000
0.2	0.9408	0.0197	0.0197	0.0197
0.4	0.7725	0.0758	0.0758	0.0758
0.6	0.5218	0.1594	0.1594	0.1594
0.8	0.2281	0.2573	0.2573	0.2573
1.0	-0.0621	0.3540	0.3540	0.3540
1.2	-0.3030	0.4343	0.4343	0.4343
1.4	-0.4567	0.4856	0.4856	0.4856
1.6	-0.4987	0.4996	0.4996	0.4996
1.8	-0.4226	0.4742	0.4742	0.4742
2.0	-0.2402	0.4134	0.4134	0.4134

We observed that the values of the amplitudes of waves on all the four vertices of both graphs. These values are shown in Table 2.

Although the two graphs are structurally different (they have a different number of edges) the wave equation evolves similarly on both graphs. In fact, it can be shown that the discrete heat equation with the same initial condition also evolves similarly on these two graphs. This reduces the power of the diffusion process to distinguish between non-isomorphic graphs.

To compare these results, we have performed a similar experiment with the edge-based wave equation on the two graphs of Figure 2. Here the initial condition is a Gaussian wave packet on a single edge of the graph (in this case (v_1, v_2) for G_1, and (u_1, u_2) for G_2). The wave is initially travelling towards the vertex v_2 in G_1 and u_2 in G_2. Figure 4 illustrates the propagation of the wave packet for the two graphs at different times. Here the packet is sampled at the nodes, for comparison with Figure 3. We have commenced the simulation with the packet on part of the structure which occurs in both graphs, but as time evolves the packet propagates to regions where the structure is different. So although the amplitudes are similar at the beginning of the sequence, as time evolves the pattern of amplitudes is very different. Specifically, the differences are small in subfigures a) and b), in subfigures c) and d) they are significantly different. This means that the edge-based wave equation is potentially a more powerful tool to distinguish graphs when compared to the discrete wave equation.


(a) t=0.5

(b) t=1.5

(c) t=2.5

(d) t=3.5

FIG. 4. Continuous wave equation for two different graphs.

The above results suggest that a partial differential equation defined using the edge-based Laplacian provide a more powerful tool to simulate a wave propagation process on a graph when compared to one defined using the discrete Laplacian. It both admits a finite speed of propagation and is more sensitive to the differences in the non-isomorphic graph structure.

6. Gaussian wave packet signature

In this section we develop a signature based on the solution of an edge-based wave equation to characterize the structure of a graph. For this purpose, we assume that the initial condition is a Gaussian wave packet on a single edge of the graph. We select the initial edge as the one that maximizes the edge betweenness centrality. The notion of edge betweenness centrality was originally proposed by Girvan and Newman [35] to find the bottlenecks of a network. For an edge e, It is defined as the number of shortest paths between all pair of nodes of the graph that run along the edge e. This will ensure a quick spread of the wave packet over the entire network.

To define a signature to characterize graph structure, we allow the wave packet to spread over the network and use the amplitude of the waves on the edges of graph over time. For each time sample, we define a local weighted sub-graph, G_{t_i}, that consists of those edges of the graph having non-zero amplitude at that specific time. In other words, for each time sample, a local subgraph is defined as

$$G_{t_i} = \{e : u(\mathscr{X}, t_i) \neq 0, w(e) = (\mathscr{X}, t_i)\},$$

where t_i represents the ith time sample and $w(e)$ represents the weight of the edge e, i.e., the amplitude of the wave residing at e at time t_i. Given a graph G, we then define its **wave packet signature (WPS)** as

$$\mathrm{WPS}(G) = \mathrm{hist}\left(G_{t_1}, G_{t_2}, G_{t_3}, ..., G_{t_n}\right), \tag{6.1}$$

where $t_1, t_2, ..., t_n$ are time samples and $\mathrm{hist}(\cdot)$ is the histogram operator which bins the list of arguments.

We choose integer values for the time interval, since for integer values the parts of wave packet are fully contained on the edges and the nodes have zero amplitude. We select the time interval as twice the number of edges in the graph. i.e., $n = 2|E|$. With this choice, the WPS becomes

$$\mathrm{WPS}(G) = \mathrm{hist}\left(G_1, G_2, G_3, ..., G_{2|E|}\right). \tag{6.2}$$

This means that the wave packet signature is constructed by binning the edge weights for the subgraphs (and hence the amplitudes of the relevant part of the wave packet over time). Since the edge weights represent the amplitudes of the wave packet, the WPS therefore encodes information about the amplitudes of parts of the wave packet over time.

The selection of the edge with highest betweenness centrality allows the wave packet to quickly propagate over the entire network. In the case where we have more than one edges with the highest edge betweenness centrality value, we randomly select among these edges. We have observed that the method is insensitive to the selection of initial edge, as long as we choose from among those with highest betweenness centrality. Also, a random selection of the initial edge may slightly reduce the performance of the signature. Note that, in the experimental evaluation, we have chosen the number of bins as 100. We also observed that the performance of the WPS does not change significantly as long as the bin size is not too large or too small.

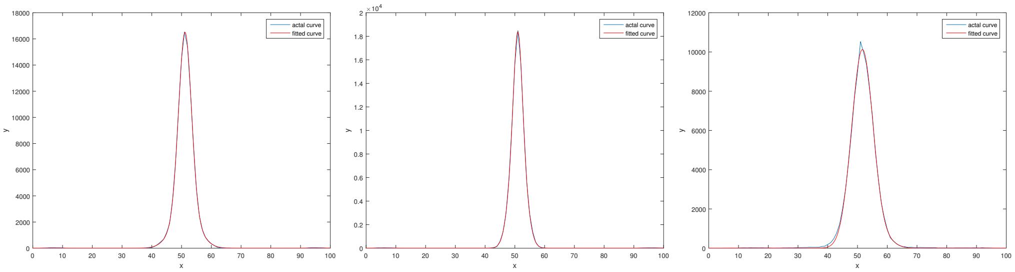
We now explore some of the properties of WPS for different types of graphs. For this purpose, we have generated random graphs and have analyzed the shape of the WPS for each of these graphs. Specifically, we have generated random graphs according to the following three different types of models.

Erdős-Rényi model (ER) [20]: An *ER* graph $G(n, p)$ is constructed by connecting n vertices randomly with probability p. i.e., each edge is included in the graph with probability p independent from every other edge. These models are also called *r*andom networks.

Watts and Strogatz model (WS) [65]: A *WS* graph $G(n, k, p)$ is constructed in the following way. First we construct a regular ring lattice, i.e., a graph with n vertices and each vertex is connected to the k nearest vertices, $k/2$ of them on either side. Then for every vertex, we take each connecting edge and rewire it with probability p. These models are also called *s*mall-world networks.

Barabási-Albert model (BA) [7]: A *BA* graph $G(n, n_0, m)$ is constructed from an initial fully connected graph with n_0 vertices. New vertices are added to the graph one at a time. Each new vertex is connected to m existing vertices with a probability that is proportional to the number of links that the existing nodes already have. These models are also called *s*cale-free networks.

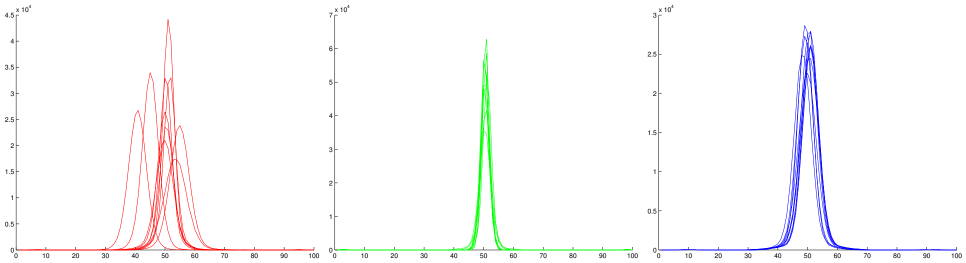
For each of these three models, we randomly generate a graph with 150 vertices and compute the WPS for each of these graphs. The resulting signatures are plotted in Figure 5.



(a) ER (b) BA (c) WS

FIG. 5. Gaussian fit to random graphs

The figure shows that the distribution in the histogram of WPS closely follows an unnormalized Gaussian distribution. This means that the WPS can be characterized by the mean, width and the amplitude of the resulting distribution, computed from the histogram bin contents. To explore these properties of the WPS, we randomly generate ten networks for each of the three graph models presented above. The number of nodes for each model varies from 151 to 160. For each model, we have chosen the parameters in such a way that all three types of network with the same number of vertices have approximately the same number of edges. For the ER model we choose $p = 10/n$, for the WS model we choose $p = 0.25$ and $k = 8$, and for the BA model we choose $n_0 = 5$ and $k = 4$. We have computed the WPS for each of these graphs, and plotted the resulting histograms in Figure 6.



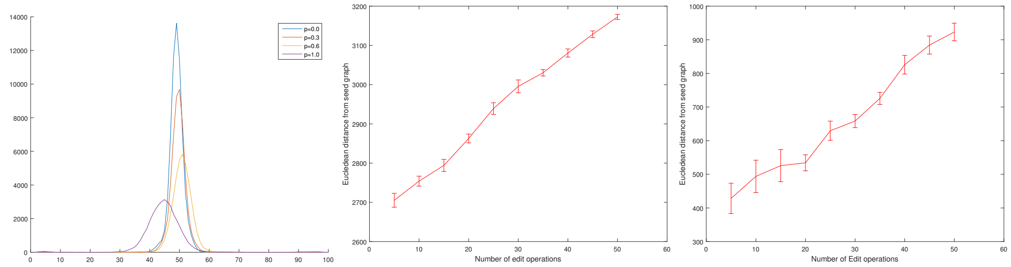
(a) ER (b) BA (c) WS

FIG. 6. Histogram of graphs generated according to different graph models

Note that due to their similar structures, the WPS of the networks generated using the WS and BA models show very small variations in mean, amplitude and width for different graphs. On the other hand, due to their random structure, the networks generated using the ER model show large variance for the mean, amplitude and the width. To further investigate the shape of the WPS under structural modification, we generate a random graph according to the WS model. We choose the parameters as $n = 30$, $k = 12$ and $p = 0$. This will produce a regular ring lattice with 30 vertices, where each vertex is connected to the 12 nearest vertices. Next, we generated three different graphs by rewiring the edges with probabilities $p = 0.3$, $p = 0.6$ and $p = 1.0$ respectively. We have computed the WPS for each graph. Figure 7(a) shows the resulting histograms for all of these graphs.

Figure 7(a) shows that, as the graph becomes more irregular, the amplitude of the histogram decreases. The histogram of the regular ring lattice has a sharp peak and small width. On the other hand, the histogram of the random graph has a lower amplitude and higher width.

In order to demonstrate the stability of WPS under controlled structure modifications, we study the relationship between the graph edit distance and the Euclidean distance between the WPS of a graph and its modified graph.We choose a seed graph according to the WS model with parameters as $n = 30$, $k = 12$ and $p = 0.1$. Next we generate 10 different graphs by randomly deleting 5, 10, ..., 50 edges from the seed graph respectively. The edit distance between the seed graph and the newly generated graph is then equal to the number of edges deleted. For each of these graphs, we compute its WPS and find its Euclidean distance from the WPS of the seed graph. We repeat this experiment 20 times. Figure 7(b) shows the average Euclidean distance along with standard errors. The standard error is calculated as $\sigma/\sqrt{n}$, where σ represents the standard deviation and n the number of samples.



(a) WS (b) WS (c) BA

FIG. 7. Effect of structural modification

Figure 7(c) shows similar results for a graph that is generated using the *BA* model. In this case, the seed graph was generated using parameters $n = 30$, $n_0 = 10$ and $k = 8$. The results shown in Figure 7 illustrate that the WPS can be used to capture the regularity structure of a graph. It is interesting to note that the *WS* networks have low standard errors as compared to the *BA* networks. This is due to the small world property of the *WS* networks, where the removal of an edge will have little effect on the propagation of wave packet.

Finally, to demonstrate the effectiveness of the WPS in distinguishing between different types of complex networks, we have generated 50 instances of each of the three random graph models discussed above, i.e., the *ER*, *WS* and *BA* networks. The number of nodes for each model varies from 151 to 200. For each graph we compute the wave packet signature. Next, we apply Principal Component Analysis (PCA) on feature vectors consisting of the histogram bin contents (where the feature vector components correspond to individual bin contents). PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data lies in the first principal component direction, the second greatest variance on the second principal component direction, and so on [42]. We embed the transformed data in to a three-dimensional space spawned by the leading three principal components. Figure 8 shows the resulting embedding of feature vectors.

The embedding results in Figure 8 show that the WPS can be effectively used to distinguish different complex network models. Note that for the WS and BA models, the graphs generated using the same model are clustered close to each other and the embedding also provides a clear separation between the two classes of the networks, i.e., WS and BA. The ER graphs, on the other hand, show high inter-class variation due to their random structures.

Running time Analysis: The local subgraph G_{t_i} can be computed in $O(|E|^2)$ time, once all the eigenfunctions are computed. This is because the solution of the wave equation for each time sample will take $O(|E|)$ time, and there are total $2|E|$ time samples. Therefore WPS can be computed in $O(|E|^3)$ time. The vertex supported eigenfunctions can be computed in $O(|V|^3)$ time while the edge-interior eigenfunctions will require $O(|E|^3)$ time. Therefore the total running time of the WPS is $O(|E|^3)$ in worst case.

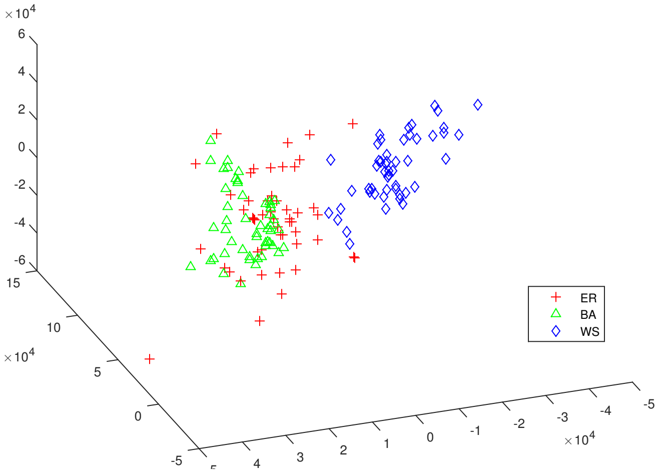


FIG. 8. Embedding of Random Graphs

7. Experiments

In this section, we test the proposed method on both real-world data and synthetic graphs. We also compare the performance of the proposed method with following state-of-the-art methods and report the results.

Heat kernel signature[60]: The HKS is based on analysing the heat diffusion process defined using the classical Laplacian over different time samples. The HKS is a local descriptor defined for every vertex of the graph. A globlal descriptor is obtained by binning the contents of the local descriptors. In our experiments, we have chosen a bin size of 100.

Wave kernel signature [1]: The WKS represents the average probability of measuring a quantum mechanical particle at a specific location at different time intervals. Like HKS, WKS is also a local signature. A global signature is constructed by binning the local signatures by keeping a bin size of 100.

Random walk kernel [63]: Random walk kernel is state-of-the-art graph kernel used to compare graphs. It gauges the similarity between two graphs by counting the frequencies of matching random walks in the two input graphs. Rather than decomposing the graph into walks of different length, it can be efficiently computed using the product graph formalism [63].

Graphlet count [43]: This method decomposes a graph into a set of common graphlets, which are small connected non-isomorphic induced subgraphs of a large graph. Here we set the size of the graphlet to 4.

Shape-DNA [57]: This method uses the first few smallest eigenvalues of the normalized Laplacian matrix of the graph and the oriented line graph, which are treated as the components of a feature vector. For our experiments, we tried vectors of different length and reported the results that give the best performance.

Heat Kernel Trace [67]: Heat kernel trace is a function of time and is given by the sum of the Laplacian eigenvalues exponentiated with time. It was used by Xiao et al. [67] to characterize the structure of a graph.

Ihara coefficients [56]: This method uses a feature-vector that records prime cycle frequencies in a graph. These cycle frequencies are computed using first few coefficients of the Ihara zeta function of the graph. The Ihara zeta function is simply the reciprocal of the characteristic polynomial of the oriented line graph, extracted from the graph in hand.

To evaluate the performance of the proposed method and compare it to alternative methods, we have used *k-fold cross-validation*. In k-fold cross validation the original data is partitioned into k subsamples of equal size. Of these k subsamples a single subsample is retained as the validation data for testing the model, and the remaining $k-1$ subsamples are used as training data. This process of cross-validation is then repeated k times. The classification accuracies are estimated using *k-nearest neighbours algorithm (k-*NN*)*. k-NN is non-parametric method that classifies each object based on k closest training examples in the feature space, i.e., an object is assigned the label that obtains the maximum votes amongst the k nearest neighbours.

7.1 *Bioinformatics datasets*

We commence by experimenting our method on graphs extracted from bioinformatics datasets. For this purpose we use the following datasets.

MUTAG [59]: The Mutagenesis dataset consists of a set of chemical compounds. The data consists of two classes, one which produces mutagenic activity and one which does not. The goal, from the point of view of classification, is to identify the mutation-causing molecules from their structure. There are 125 chemicals in the active class and 63 in the inactive class. The average number of nodes and the edges of these graphs is 17.93 and 19.79 respectively.

PROTEINS [19]: In this dataset each protein is represented as a graph, where the nodes correspond to secondary structure elements. Two nodes are connected whenever they are neighbours either in the amino acid sequence or in the 3D space of the protein tertiary structure [11]. The task is to distinguish between enzymes and non-enzymes. There are 1113 graphs in the dataset. The minimum number of nodes of the graphs is 4 while the maximum number is 620. The average number of nodes is 39.06.

We have computed the WPS for both the datasets and measured their performance using 10-fold cross validation. We have also performed similar experiments with some other stat-of-the art methods including the heat kernel signature, the wave kernel signature, the random walk kernel, the coefficients of the Ihara zeta function, the graphlet count and the normalized Laplacian of both the graph and the oriented line graph. Table 3 compares the resulting accuracies.

Table 3. Performance Comparison.

Method	MUTAG	PROTEINS
Wave Packet Signature	**86.84**	**70.01**
Heat Kernel Signature	85.79	66.27
Wave Kernel Signature	81.91	67.39
Random Walk	85.79	68.64
Graphlet Count	86.31	64.41
Heat Kernel Trace	84.21	62.88
Ihara Coefficients	84.21	66.21
Normalized Laplacian of graph	85.78	65.85
Normalized Laplacian of OLG	74.21	60.63

These results suggest that the WPS can be used on bioinformatics datasets with higher accuracy.

Note that both MUTAG and PROTEINS are labelled datasets, where the nodes/edges are annotated with unique labels. However, in our experiments we have ignored these labels as currently our method works only on unlabelled simple graphs.

7.2 *Graphs extracted from Images*

We now perform experiments on graphs that are extracted from the images in the Columbia object image library (COIL) dataset [51]. This dataset contains views of 3D objects under controlled viewer and lighting conditions. For each object in the database there are 72 equally spaced views taken at 5 degree intervals, as the object is rotated on a turntable. The objective here is to cluster different views of the same object onto the same class. To establish a graph on the images of the different object views, we first extract feature points from the image. For this purpose, we use Harris corner detector[39]. We then construct the following types of the graphs using the selected feature points as vertices.
Delaunay triangulation [18]: A Delaunay triangulation (DT) for a set P of points in a Euclidean space is a triangulation, $DT(P)$, such that no point in P is inside the circumcircle of any triangle in $DT(P)$.
Gabriel graphs [31]: The Gabriel graph (GG) for a set of n points is a subset of Delaunay triangulation, which connects two data points v_i and v_j for which there is no other point v_k inside the open ball whose diameter is the edge (v_i, v_j).

The purpose of performing experiments on Gabriel graphs is to investigate the performance of WPS under controlled structured modifications. To evaluate the performance of WPS we select four different object with all their 72 views. We extract the Delaunay triangulation and Gabriel graph for each view and compute its WPS. The average number of nodes of these graphs is 89, and the average number of edges of the Delaunay triangulation is 245, while the average number of edges of the Gabriel graph is 175. Next we measure the performance of the proposed method using 9-fold cross validation. The reason for choosing 9 -fold is that it nicely splits the set into 9 sets each with 32 objects. The resultant accuracies are reported in Table 4. To compare the performance of the proposed method with other state-of-the-art methods, we have performed a similar experiment with the heat kernel signature, the wave kernel signature, the random walk kernel, the coefficients of the Ihara zeta function, the graphlet count and the normalized Laplacian of both the graph and the oriented line graph. These results are also shown in Table 4. As with the bioinformatics datasets, in all cases, we have applied PCA to the resulting signatures and have used the first ten principal components in each case.

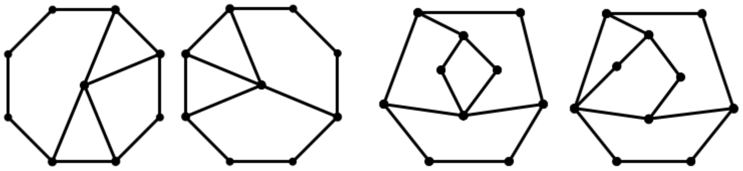
Table 4. Performance Comparison.

Method	DT	GG
Wave Packet Signature	**99.65**	98.61
Heat Kernel Signature	99.31	**98.96**
Wave Kernel Signature	96.86	96.53
Random Walk	99.31	97.91
Graphlet Count	**99.65**	93.05
Heat Kernel Trace	**99.65**	**98.96**
Ihara Coefficients	**99.65**	93.75
Normalized Laplacian of graph	**99.65**	89.58
Normalized Laplacian of OLG	93.75	60.63

From Table 4, the WPS gives the same performance on Delaunay triangulations as some of the alternate state-of-the art methods. However on Gabriel graphs the proposed method gives higher accuracy compared to the alternate methods, except the heat kernel trace and the heat kernel signature, which give slightly higher accuracy. The above results suggest that WPS can be a powerful tool to characterize graphs. Although its performance is comparable to some state-of-the-art methods on regular graphs, such as Delaunay triangulations and Gabriel graphs, it outperforms most state-of-the-art methods on irregular graphs such as bioinformatics data. In the next section we will show that the WPS is also a very powerful tool for distinguishing between cospectral graphs.

7.3 *Cospectral graphs:*

One of the advantages of using the solution of wave equation defined using the edge-based Laplacian is that it is less prone to the problem of failing to distinguish graphs due to cospectrality of the Laplacian or adjacency matrices. To demonstrate this we have selected pairs of graphs that are the cospectral with respect to their different matrix representations. Figure 9(a) and Figure 9(b) show two pairs of cospectral graphs with respect to both their adjacency matrices and the adjacency matrices of their compliments. We have computed the WPS of these two graphs and applied Principal Component Analysis (PCA) [42] on the resultant feature vectors. Figure 12 shows the first two principal eigenvectors. The embedding results show that the WPS can be used to distinguished non-isomorphic graphs that are cospectral with respect to their adjacency matrix representations.


(a) G_1 and G_2.

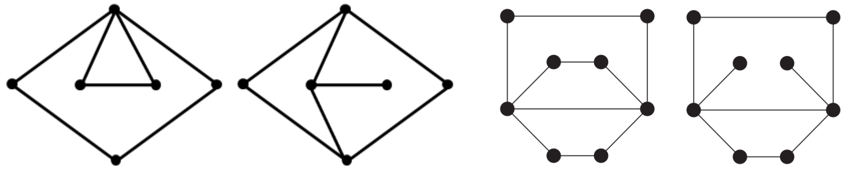
(b) G_3 and G_4.

FIG. 9. Examples of cospectral graphs with respect to adjacency matrices of graphs and adjacency matrices of their compliment graphs

Next we perform a similar experiment with graphs that are cospectral with respect to their Laplacian matrix representation. Figure 10(a) shows a pair of non-isomorphic graphs that are cospectral with respect to their Laplacian matrix representation, while Figure 10(b) shows a pair of graphs that are cospectral with respect to their normalized Laplacian matrix representation. We have computed WPS for both pairs and applied PCA on the resultant feature vectors. The resultant embedding is shown in Figure 12.

To demonstrate the power of WPS to distinguish graphs that are cospectral with respect to other matrix representations, we select a pair of non-isomorphic graphs with same number of nodes and edges and with same degree distribution (see Figure 11). These graphs are cospectral with respect to their adjacency matrix, normalized adjacency matrix, Laplacian matrix, Normalized Laplacian matrix and signless Laplacian matrix.

We have computed the WPS for these two graphs and embed the resulting feature vectors in a two-



(a) G_5 and G_6.

(b) G_7 and G_8.

FIG. 10. Examples of cospectral graphs with respect to Laplacian matrix (left) and normalized Laplacian matrix (right).

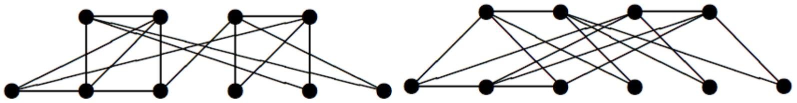


FIG. 11. Non-isomorphic graphs G_9 and G_{10} that are cospectral with respect to their adjacency, normalized adjacency, Laplacian, normalized Laplacian, and signless Laplacian matrices.

dimensional space, shown in Figure 12. Although the graphs were embedded very close in feature space, the proposed method was still able to distinguished between the two graphs.

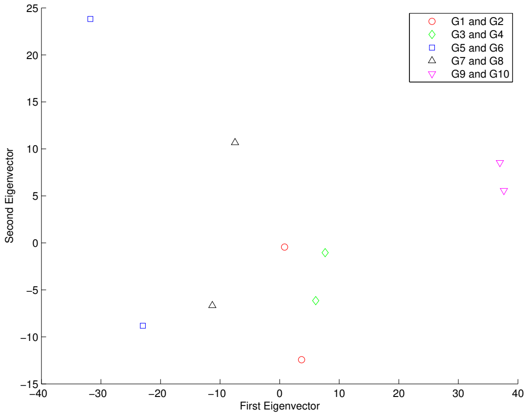


FIG. 12. Embedding of cospectral graphs.

The above results suggest that WPS is a powerful tool for distinguishing between graphs that are cospectral with respect to their different matrix representations including adjacency, normalized adjacency, Laplacian, normalized Laplacian, and signless Laplacian matrices. Note that some of these pair of graphs cannot be distinguished by some of the graph characterization methods. For example in [3], we have shown that graphs of Figure 9 cannot be distinguished by random walks on graphs. This is due to the fact that the structure of random walk is determined by the power of adjacency matrix and these graphs are cospectral with respect to their adjacency matrices. Similarly the graphs of Figure 10(a) cannot be distinguished by any method that is based on the Laplacian spectrum, and these include the heat kernel trace [67]. Note that these two graphs can not be distinguished by Ihara coefficients which are considered as a powerful tool to distinguish between non-isomorphic cospectral graphs [3]. This suggests that the proposed method is a powerful tool to distinguish cospectral graphs.

8. Conclusion

In this paper we have developed a complete solution of the wave equation on a graph defined using the edge-based Laplacian of a graph and used the resulting solution to define a signature for characterizing graphs. We assume the initial condition to be a Gaussian wave packet on a single edge of the graph. The advantage of using the edge-based Laplacian over the vertex-based Laplacian is that it allows the direct application of many results from analysis to the graph theoretic domain. Another advantage of using edge-based Laplacian is that it allows finite speed of propagation of a signal on a graph. This allows for the implementation of many partial differential equations on a graph including the wave equation, which does not have finite speed of propagation if defined using the discrete Laplacian. This may be of potential utility in the study of networks where distance and speed of propagation are important. We have also experimentally demonstrated that the partial differential equations defined using the discrete Laplacian may fail to capture the connectivity structure of a network. The current analysis is limited to the case of uniform edge lengths.

There are a number of directions in which the work reported here can be developed. In terms of immediate developments, it would be interesting to focus on the case where edge lengths may vary, or equivalently the speed of propagation between nodes is non-uniform. From a methodological point of view, it would also be interesting to explore the solutions to alternative differential equations such as the Fokker-Plank and relativistic wave (Dirac) equation. Finally, it would be interesting to use the wave packet approach to study non-dispersive solutions or solitons on graphs, and to explore how they can be used to categorize different types of graph.

As far as applications are concerned there are many potential avenues that can be explored. For instance, in computer vision it would be interesting to study whether the packet-solution of the wave equation can be applied to three-dimensional shapes, and whether they can give better result to the state of the art heat kernel signature [60]. As far as network science is concerned, there are a numerous potential applications. First, it would be interesting to study whether the application to the packet-solutions of the wave equation can be applied to epidemiological data [14, 40], and whether they can furnish better models of the spread of epidemics. It would also be interesting to explore if the proposed framework can be applied to citation networks in order to capture the dynamic growth of publications of researchers [50, 71]. Second, the new models presented in this paper may provide alternative ways of studying properties such as network centrality and communicability, as well as an alternative means of modelling network evolution via processes such as preferential attachment [8, 52].

A. Solution of Wave Equation

In this section we provide a general solution of a wave equation on a graph. Let the graph coordinate $\mathscr{X}$ define an edge e and a value of the standard coordinate on that edge x. The eigenfunctions of the edge-based Laplacian, corresponding to eigenvalue $\omega + 2n\pi$, are

$$\phi_{\omega,n}(\mathscr{X}) = C(e,\omega)\cos\left[B(e,\omega) + \omega x + 2\pi n x\right].$$

The edge-based wave equation on the graph is

$$\frac{\partial^2}{\partial t^2}u(\mathscr{X},t) = \Delta u(\mathscr{X},t). \tag{A.1}$$

To solve the above equation, we seek separable solutions of the form $u(\mathscr{X},t) = \phi_{\omega,n}(\mathscr{X})g(t)$. Substituting this separable solution into the above equation, we obtain

$$\phi_{\omega,n}(\mathscr{X})\frac{\partial^2}{\partial t^2}g(t) = -g(t)\left(\omega+2\pi n\right)^2\phi_{\omega,n}(\mathscr{X}),$$

which implies that,

$$\frac{\partial^2}{\partial t^2}g(t) = -g(t)\left(\omega+2\pi n\right)^2,$$

This is a second order partial differential equation whose characteristic equation has complex roots at $\pm i(\omega+2\pi n)$. Therefore the time-based part of the solution becomes

$$g(t) = \alpha_{\omega,n}\cos\left[(\omega+2\pi n)t\right] + \beta_{\omega,n}\sin\left[(\omega+2\pi n)t\right].$$

From the *principal of superposition*, the general solution is the sum of above solutions, for all the possible values of ω and n. Therefore the general solution takes the form

$$u(\mathscr{X},t) \quad = \textstyle\sum_\omega\sum_n C(e,\omega)\cos\left[B(e,\omega)+\omega x+2\pi nx\right]\left\{\alpha_{\omega,n}\cos\left[(\omega+2\pi n)t\right]+\beta_{\omega,n}\sin\left[(\omega+2\pi n)t\right]\right\}. \tag{A.2}$$

Here the coefficients $\alpha_{\omega,n}$ and $\beta_{\omega,n}$ depend on the initial conditions of the wave equation.

A.1 *Initial conditions*

Since the wave equation is a second order partial differential equation, we can impose initial conditions on both the position and speed of the wave. Hence we write

$$u(\mathscr{X},0) = p(\mathscr{X}) \tag{A.3}$$

and

$$\frac{\partial u}{\partial t}(\mathscr{X},0) = q(\mathscr{X}), \tag{A.4}$$

where $p(\mathscr{X})$ and $q(\mathscr{X})$ are initial conditions on position and speed respectively. To find $p(\mathscr{X})$, we take the value of $u(\mathscr{X},t=0)$ from Equation(A.2) and equate this with Equation(A.3). Therefore, we obtain

$$p(\mathscr{X}) = \sum_\omega\sum_n \alpha_{\omega,n}C(e,\omega)\cos\left[B(e,\omega)+\omega x+2\pi nx\right], \tag{A.5}$$

Now differentiating Equation(A.2), taking the value at $t=0$, and equating this with Equation(A.4), we obtain

$$q(\mathscr{X}) \quad = \quad \sum_\omega\sum_n \beta_{\omega,n}(\omega+2\pi n)\times C(e,\omega)\cos\left[B(e,\omega)+\omega x+2\pi nx\right]. \tag{A.6}$$

We can determine the coefficients $\alpha_{\omega,n}$ and $\beta_{\omega,n}$ using the orthogonality of the eigenfunctions. So from Equation (A.5), we obtain

$$\alpha_{\omega,n} = \sum_e C(e,\omega)\frac{1}{2}\left[F_{e,\omega,n} + F^*_{e,\omega,n}\right], \tag{A.7}$$

where

$$F_{e,\omega,n} = \exp\left[iB(e,\omega)\right]\int_0^1 dxp(e,x)\exp\left[i\omega x\right]\exp\left[i2\pi nx\right].$$

We can determine $\beta_{\omega,n}$, in a similar manner, using Equation A.6, to obtain

$$\beta_{\omega,n}(\omega+2\pi n) = \sum_e C(e,\omega)\frac{1}{2}\left[G_{e,\omega,n} + G^*_{e,\omega,n}\right],$$

where

$$G_{e,\omega,n} = \exp\left[iB(e,\omega)\right]\int_0^1 dxq(x,e)\exp\left[i(\omega+2\pi n)x\right]$$

The values of $F_{e,\omega,n}$ and $G_{e,\omega,n}$ depend on the initial position and the speed of the wave.

A.2 *Gaussian wave packet*

In this section, we obtain a complete solution of the wave equation, where the initial condition is a Gaussian wave packet confined to an edge of the graph. For this purpose, we assume that the initial position of the wave equation is the Gaussian wave packet

$$p(e,x) = \exp\left[-a(x-\mu)^2\right]$$

on one particular edge and zero everywhere else. In practice, this can be done by taking the mean of the wave packet to be the centre of the edge (i.e., $\mu = 0.5$) and keeping the variance sufficiently small. Since the variance or the width of the wave packet is very small compared to the length of the edge, we assume the energy of the wave packet outside the edge is negligible. We therefore have

$$\begin{aligned} F_{e,\omega,n} &= \exp\left[iB(e,\omega)\right]\times\int_0^1 dx\exp\left[-a(x-\mu)^2\right]\exp\left[i\omega x\right]\exp\left[i2\pi nx\right], \\ &= \exp\left[iB(e,\omega)\right]\exp\left[i\mu\omega\right]\exp\left[-\frac{\omega^2}{4a}\right]\times\int_0^1 dx\exp\left[-a\left(x-\mu-\frac{i\omega}{2a}\right)^2\right]\exp\left[i2\pi nx\right]. \end{aligned}$$

When the Gaussian wave packet is fully contained by one edge, i.e., $p(x,e)$ is only supported on one edge, then

$$F_{e,\omega,n} = \exp\left[iB(e,\omega)\right]\exp\left[i\mu\omega\right]\exp\left[-\frac{\omega^2}{4a}\right]\times\int_{-\infty}^{\infty} dx\exp\left[-a\left(x-\mu-\frac{i\omega}{2a}\right)^2\right]\exp\left[i2\pi nx\right].$$

Since $\sin(2\pi nx)$, is an odd function, the integral in the above equation evaluates to

$$\int_{-\infty}^{\infty} dx\exp\left[-a\left(x-\mu-\frac{i\omega}{2a}\right)^2\right]\exp\left[i2\pi nx\right] = \int_{-\infty}^{\infty} dx\exp\left[-a\left(x-\mu-\frac{i\omega}{2a}\right)^2\right]\cos\left[2\pi nx\right].$$

Using integration by parts and substituting the result into equation (A.8), we obtain

$$F_{e,\omega,n} \quad = \quad \sqrt{\frac{\pi}{a}} \exp\left[i\left(B(e,\omega)+\mu(\omega+2\pi n)\right)\right] \times \exp\left[-\frac{1}{4a}(\omega+2n\pi)^2\right].$$

In a similar manner, solving for $F^*_{e,\omega,n}$, we obtain

$$F^*_{e,\omega,n} \quad = \quad \sqrt{\frac{\pi}{a}} \exp\left[-i\left(B(e,\omega)+\mu(\omega+2\pi n)\right)\right] \times \exp\left[-\frac{1}{4a}(\omega+2n\pi)^2\right].$$

Now the coefficient $\alpha_{\omega,n}$ can be found by substituting the values of $F_{e,\omega,n}$ and $F^*_{e,\omega,n}$ into Equation (A.7). Hence,

$$\alpha_{\omega,n} \quad = \quad \sqrt{\frac{\pi}{a}} \exp\left[-\frac{1}{4a}(\omega+2n\pi)^2\right] \times C(e,\omega)\cos[B(e,\omega)+\mu\,(\omega+2\pi n)]. \qquad \text{(A.8)}$$

Since $p(x,e)$ is zero at both ends of the edge, the coefficients β can be found straightforwardly, as

$$\beta_{\omega,n} \quad = \quad \sqrt{\frac{\pi}{a}} \exp\left[-\frac{1}{4a}(\omega+2n\pi)^2\right] C(e,\omega)\sin[B(e,\omega)+\mu\,(\omega+2\pi n)]. \qquad \text{(A.9)}$$

Substituting the values of $\alpha_{\omega,n}$ and $\beta_{\omega,n}$ into Equation (A.2) gives us the general solution to the wave equation on a graph.

A.3 *Complete reconstruction*

Let f be the single edge of the graph on which the initial function is non-zero. Let the Gaussian wave packet be fully contained on this edge. Then the general solution is given by:

$$\begin{aligned} u(\mathscr{X},t) \quad = \quad & \sum_{\omega} \sqrt{\frac{\pi}{a}} C(e,\omega)C(f,\omega) \times \sum_{n} \exp\left[-\frac{1}{4a}(\omega+2\pi n)^2\right] \\ & \times \cos\left[B(e,\omega)+\omega x+2\pi nx\right] \times \cos\left[B(f,\omega)+(\omega+2\pi n)(t+\mu)\right]. \end{aligned}$$

For a particular principal frequency ω, we need to calculate

$$u_{\omega} \quad = \quad \sum_{n} \sqrt{\frac{\pi}{a}} \exp\left[-\frac{1}{4a}(\omega+2\pi n)^2\right] \times \cos\left[B(\omega,e)+\omega x+2\pi nx\right] \times \cos\left[B(\omega,f)+(\omega+2\pi n)(t+\mu)\right].$$

Writing the cosines appearing above in complex exponential form, we obtain

$$\begin{aligned} u_w \quad = \quad & \frac{1}{4}\sum_{n} \sqrt{\frac{\pi}{a}} \exp\left[-\frac{1}{4a}(\omega+2\pi n)^2\right] \times \left(\exp\left[i[B(e,\omega)+B(f,\omega)]\right] \times \exp\left[i(\omega+2\pi n)(x+t+\mu)\right]\right. \\ & + \exp\left[-i[B(e,\omega)+B(f,\omega)]\right] \times \exp\left[-i(\omega+2\pi n)(x+t+\mu)\right] \\ & + \exp\left[i[B(e,\omega)-B(f,\omega)]\right] \times \exp\left[i(\omega+2\pi n)(x-t-\mu)\right] \\ & \left. + \exp\left[-i[B(e,\omega)-B(f,\omega)]\right] \times \exp\left[-i(\omega+2\pi n)(x-t-\mu)\right]\right). \end{aligned}$$

To solve the above equation, we therefore need to evaluate terms of the form

$$\sum_{n} \frac{\pi}{a} \exp\left[-\frac{1}{4a}\right] \exp\left[i[B(e,\omega)+B(f,\omega)]\right] \exp\left[i(\omega+2\pi n)(x+t+\mu)\right],$$

where the values of ω and n depend on the particular eigenfunction under evaluation. We solve the above equation for each case separately. Table 1 lists all solutions for each possible set of principal frequencies.

References

[1] AUBRY, M., SCHLICKEWEI, U. & CREMERS, D. (2011) The Wave Kernel Signature: A Quantum Mechanical Approach to Shape Analysis. *Tech. rep., TU München, Germany*.

[2] AZIZ, F., WILSON, R. C. & HANCOCK, E. R. (2013a) Analysis of Wave Packet Signature of a Graph. *Computer Analysis of Images and Patterns*, pp. 128–136.

[3] AZIZ, F., WILSON, R. C. & HANCOCK, E. R. (2013b) Backtrackless Walks on a Graph. *IEEE Transactions on Neural Networks and Learning Systems*, **24**(6), 977–989.

[4] AZIZ, F., WILSON, R. C. & HANCOCK, E. R. (2013c) Graph Characterization using Gaussian Wave Packet Signature. *Similarity Based Pattern Recognition*.

[5] BAND, R. & GNUTZMANN, S. (2017) Quantum Graphs via Exercises. *arXiv:1711.07435 [math.SP]*.

[6] BAND, R., SHAPIRA, T. & SMILANSKY, U. (2006) Nodal domains on isospectral quantum graphs: the resolution of isospectrality?. *Journal of Physics A: Mathematical and General*, **39**(45), 13999.

[7] BARABÁSI, A. & ALBERT, R. (1999) Emergence of Scaling in Random Networks. *Science*, pp. 509–512.

[8] BENZI, M. & KLYMKO, C. (2013) Total communicability as a centrality measure. *Journal of Complex Networks*, **1**(2), 124–149.

[9] BIAMONTE, J., FACCIN, M. & DOMENICO, M. D. (2017) Complex Networks: from Classical to Quantum. *arXiv:1702.08459v1*.

[10] BORGWARDT, K. M. & KRIEGEL, H. P. (2005) Shortest-path kernels on graphs. in *Fifth IEEE International Conference on Data Mining*, pp. 8–pp. IEEE.

[11] BORGWARDT, K. M., ONG, C. S., SCHÖNAUER, S., VISHWANATHAN, S. V. N., SMOLA, A. J. & KRIEGEL, H.-P. (2005) Protein function prediction via graph kernels. *Bioinformatics*, **21**, 4756.

[12] BUNKE, H. (1997) On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, **18**(8), 689–694.

[13] CATTANEO, C. (1997) The spectrum of the continuous Laplacian on a graph. *Monatshefte für Mathematik*, **124**(3), 215–235.

[14] CHEN, I., BENZI, M., CHANG, H. H. & HERTZBERG, V. S. (2017) Dynamic communicability and epidemic spread: a case study on an empirical dynamic contact network. *Journal of Complex Networks*, **5**(2), 274–302.

[15] COIFMAN, R. R. & LAFON, S. (2006) Diffusion maps. *Applied and Computational Harmonic Analysis*, **21**(1), 5 – 30.

[16] COJOCARU, R. I. (2006) Complex Network Problems in Physics, Computer Science and Biology. Ph.D. thesis.

[17] CUQUET, M. & CALSAMIGLIA, J. (2009) Entanglement Percolation in Quantum Complex Networks. *Phys. Rev. Lett.*, **103**, 240503.

[18] DELAUNAY, B. (1934) Sur la sphére vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, pp. 793–800.

[19] DOBSON, P. D. & DOIG, A. J. (2003) Distinguishing Enzyme Structures from Non-enzymes Without Alignments. *Journal of Molecular Biology*, **330**, 771783.

[20] ERDÕS, P., R. (1960) A.: On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, p. 1761.

[21] ESCOLANO, F., HANCOCK, E. & LOZANO, M. (2012) Heat diffusion: Thermodynamic depth complexity of networks. *Physics Review E.*

[22] ESTRADA, E. & RODRÍGUEZ-VELÁZQUEZ, J. A. (2005) Subgraph centrality in complex networks. *Phys. Rev. E*, **71**, 056103.

[23] EXNER, P. & BARSEGHYAN, D. (2013) Spectral Analysis of Schrödinger Operators with Unusual Semiclassical Behavior. *Acta Polytechnica*, **53**(3).

[24] EXNER, P. & POST, O. (2008) Quantum networks modelled by graphs. *AIP Conference Proceedings*, **998**(1), 1–17.

[25] FACCIN, M., JOHNSON, T., BIAMONTE, J., KAIS, S. & MIGDAŁ, P. (2013) Degree Distribution in Quantum Walks on Complex Networks. *Phys. Rev. X*, **3**, 041007.

[26] FACCIN, M., MIGDAŁ, P., JOHNSON, T. H., BERGHOLM, V. & BIAMONTE, J. D. (2014) Community Detection in Quantum Complex Networks. *Phys. Rev. X*, **4**, 041012.

[27] FERRAZ DE ARRUDA, H., NASCIMENTO SILVA, F., QUEIROZ MARINHO, V., RAPHAEL AMANCIO, D. & DA FONTOURA COSTA, L. (2018) Representation of texts as complex networks: a mesoscopic approach. *Journal of Complex Networks*, **6**(1), 125–144.

[28] FREEMAN, L. (1997) A set of measures of centrality based on betweenness. *Sociometry*, **40**, 35–41.

[29] FRIEDMAN, J. & TILLICH, J. P. (2004a) Calculus on Graphs. *CoRR*.

[30] FRIEDMAN, J. & TILLICH, J. P. (2004b) Wave equations for graphs and the edge based Laplacian. *Pacific Journal of Mathematics*, pp. 229–266.

[31] GABRIEL, K. R. & SOKAL, R. R. (1969) A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology*, pp. 205–222.

[32] GAMBLE, J., CHINTAKUNTA, H., WILKERSON, A. & KRIM, H. (2016) Node Dominance: Revealing Community and Core-Periphery Structure in Social Networks. *IEEE Transactions on Signal and Information Processing over Networks*, **2**(2), 186–199.

[33] GÄRTNER, T., FLACH, P. & WROBEL, S. (2003) On graph kernels: Hardness results and efficient alternatives. in *Learning Theory and Kernel Machines*, pp. 129–143. Springer.

[34] GIRAUD, O. & THAS, K. (2010) Hearing shapes of drums: Mathematical and physical aspects of isospectrality. *Reviews of Modern Physics*, **82**(3), 2213–2255.

[35] GIRVAN, M. & NEWMAN, M. (2002) Community structure in social and biological networks. in *Proceedings of National Academy of Sciences of USA*, vol. 99, pp. 7821–7826.

[36] GORDON, C., WEBB, D. & WOLPERT, S. (1992) Isospectral plane domains and surfaces via Riemannian orbifolds. *Inventiones mathematicae*, **110**(1), 1–22.

[37] GUTKIN, B. & SMILANSKY, U. (2001) Can one hear the shape of a graph?. *Journal of Physics A-Mathematical and General*, **34**(31), 6061–6068.

[38] HAMON, R., BORGNAT, P., FLANDRIN, P. & ROBARDET, C. (2016) Extraction of Temporal Network Structures From Graph-Based Signals. *IEEE Transactions on Signal and Information Processing over Networks*, **2**(2), 215–226.

[39] HARRIS, C. & STEPHENS, M. (1988) A combined corner and edge detector. *In Fourth Alvey Vision Conference, Manchester, UK*, pp. 147–151.

[40] HAUTPHENNE, S., KRINGS, G., DELVENNE, J.-C. & BLONDEL, V. D. (2015) Sensitivity analysis of a branching process evolving on a network with application in epidemiology. *Journal of Complex Networks*, **3**(4), 606–641.

[41] HIROTA, R. & SUZUKI, K. (1973) Theoretical and experimental studies of lattice solitons in nonlinear lumped networks. *Proceedings of the IEEE*, **61**(10), 1483–1491.

[42] JOLLIFFE, I. T. (1986) Principal Component Analysis. *Springer-Verlag, New York.*

[43] KONDOR, R., SHERVASHIDZE, N. & BORGWARDT, K. (2009) The graphlet spectrum. in *In Proceedings of the 26th International Conference on Machine Learning*.

[44] KOSTRYKIN, V., POTTHOFF, J. & SCHRADER, R. (2012) Finite propagation speed for solutions of the wave equation on metric graphs. *Journal of Functional Analysis*, **263**(5), 1198 – 1223.

[45] KUCHMENT, P. (2004) Quantum graphs I: Some basic structures. *Waves in Random Media*, **14**(1).

[46] KUCHMENT, P. (2005) Quantum graphs II: Some spectral properties of quantum and combinatorial graphs. *Journal of Physics A-Mathematical and General*, **38**(22), 4887–4900.

[47] KUCHMENT, P. (2008) Quantum graphs: an introduction and a brief survey Analysis on Graphs and its applications. *Proc. Symp. Pure Math. (Providence, RI: American Mathematical Society)*, pp. 291–314.

[48] KURASOV, P. (2008) Schrödinger operators on graphs and geometry I: Essentially bounded potentials. *Journal of Functional Analysis*, **254**(4), 934 – 953.

[49] MESSMER, B. T. & BUNKE, H. (2000) Efficient Subgraph Isomorphism Detection: A Decomposition Approach. *IEEE Transactions on Knowledge and Data Engineering*, **12**(2), 307–323.

[50] MORRISON, G., GIOVANIS, E., PAMMOLLI, F. & RICCABONI, M. (2014) Border sensitive centrality in global patent citation networks. *Journal of Complex Networks*, **2**(4), 518–536.

[51] MURASE, H. & K., N. S. (1995) Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, **14**(1), 5–24.

[52] NICOSIA, V., MACHIDA, T., WILSON, R. C., HANCOCK, E. R., KONNO, N., LATORA, V. & SEVERINI, S. (2013) Co-evolution of networks and quantum dynamics: a generalization of preferential attachment. *Journal of Statistical Mechanics: Theory and Experiment*, **2013**(08), P08016.

[53] PESENSON, I. (2005) Band limited functions on quantum graphs. *Proceedings of the American Mathematical Society*, **133**, 36473655.

[54] PESENSON, I. (2006) Analysis of band-limited functions on quantum graphs. *Applied and Computational Harmonic Analysis*, **21**(2), 230 – 244.

[55] PESENSON, I. (2015) Sampling solutions of Schrödinger equations on combinatorial graphs. in *2015 International Conference on Sampling Theory and Applications (SampTA)*, pp. 82–85.

[56] REN, P., WILSON, R. C. & HANCOCK, E. R. (2011) Graph Characterization via Ihara Coefficients. *IEEE Transactions on Neural Networks*, **22**, 233–245.

[57] REUTER, M., WOLTER, F.-E. & PEINECKE, N. (2006) LaplaceBeltrami spectra as Shape-DNA of surfaces and solids. *Computer-Aided Design*, **38**(4), 342 – 366.

[58] ROSSI, L., TORSELLO, A. & HANCOCK, E. R. (2014) Node Centrality for Continuous-Time Quantum Walks. in *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 103–112.

[59] SRINIVASAN, A., MUGGLETON, S. H., STERNBERG, M. J. E. & KING, R. D. (1996) Theories for mutagenicity: a study in first-order and feature-based induction. *Artificial Intelligence*, pp. 277–299.

[60] SUN, J., OVSJANIKOV, M. & GUIBAS, L. (2009) A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion. in *Proceedings of the Symposium on Geometry Processing*, pp. 1383–1392.

[61] SUNADA, T. (1985) Riemannian Coverings and Isospectral Manifolds. *Annals of Mathematics*, **121**(1), 169–186.

[62] TORSELLO, A., ROBLES-KELLY, A. & HANCOCK, E. R. (2007) Discovering Shape Classes using Tree Edit-Distance and Pairwise Clustering. *International Journal of Computer Vision*, **72**(3), 259–285.

[63] VISHWANATHAN, S. V. N., SCHRAUDOLPH, N. N., KONDOR, R. & BORGWARDT, K. M. (1201-1242) Graph Kernels. *Journal of Machine Learning Research*, p. 2010.

[64] WANG, J., WILSON, R. C. & HANCOCK, E. R. (2017) Spin statistics, partition functions and network entropy. *Journal of Complex Networks*, **5**(6), 858–883.

[65] WATTS, D. J. & STROGATZ, S. (1998) Collective dynamics of 'small-world' networks. *Nature*, p. 440442.

[66] WILSON, R. C., AZIZ, F. & HANCOCK, E. R. (2013) Eigenfunctions of the Edge-Based Laplacian on a Graph. *Linear Algebra and its Applications*, **438**(11), 4183–4189.

[67] XIAO, B., HANCOCK, E. & WILSON, R. (2009) Graph characteristics from the heat kernel trace. *Pattern Recognition*, **42**, 2589–2606.

[68] XIAO, B., HANCOCK, E. R. & WILSON, R. C. (2010) Geometric characterization and clustering of graphs using heat kernel embeddings. *Image Vision Comput.*, **28**, 1003–1021.

[69] YUCEL, M., MUCHNIK, L. & HERSHBERG, U. (2017) Detection of network communities with memory-biased random walk algorithms. *Journal of Complex Networks*, **5**(1), 48–69.

[70] ZHAN, C. & TSE, C. K. (2017) A network model for growth of publications and citations. *Journal of Complex Networks*, **5**(2), 303–314.