

This is a repository copy of *A Hierarchical Transitive-Aligned Graph Kernel for Un-attributed Graphs*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/200185/>

---

**Preprint:**

Bai, Lu, Cui, Lixin and Hancock, Edwin R. orcid.org/0000-0003-4496-2028 (2020) A Hierarchical Transitive-Aligned Graph Kernel for Un-attributed Graphs. [Preprint]

---

**Reuse**

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

**Takedown**

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

# A Hierarchical Transitive-Aligned Graph Kernel for Un-attributed Graphs

Lu Bai<sup>1</sup>, Lixin Cui<sup>1\*</sup>, and Edwin R. Hancock<sup>3</sup>, *IEEE Fellow*

**Abstract**—In this paper, we develop a new graph kernel, namely the Hierarchical Transitive-Aligned kernel, by transitively aligning the vertices between graphs through a family of hierarchical prototype graphs. Comparing to most existing state-of-the-art graph kernels, the proposed kernel has three theoretical advantages. First, it incorporates the locational correspondence information between graphs into the kernel computation, and thus overcomes the shortcoming of ignoring structural correspondences arising in most R-convolution kernels. Second, it guarantees the transitivity between the correspondence information that is not available for most existing matching kernels. Third, it incorporates the information of all graphs under comparisons into the kernel computation process, and thus encapsulates richer characteristics. By transductively training the C-SVM classifier, experimental evaluations demonstrate the effectiveness of the new transitive-aligned kernel. The proposed kernel can outperform state-of-the-art graph kernels on standard graph-based datasets in terms of the classification accuracy.

**Index Terms**—Graph Kernels, Transitive Vertex Alignment

## I. INTRODUCTION

Graph-based representations are powerful tools to represent structure data that is described with pairwise relationships between components. The main challenge arising in analyzing the graph-based data is how to learn effective numeric features of the discrete graph structures. One way to achieve this is to employ graph kernels, that can characterize graph structures in a high dimensional space and thus better preserve the structure information [1].

### A. Related Works

In machine learning, a graph kernel is defined in terms of a similarity measure between graph structures. One of the most successful and widely used approach to defining kernels between a pair of graphs is to decompose the graphs into substructures and to compare/count pairs of specific isomorphic substructures [1]. Specifically, any graph decomposition can be used to define a kernel, e.g., the graph kernel based on comparing all pairs of decomposed a) walks, b) paths and c) restricted subgraph or subtree structures. With this scenario, Kashima et al. [2] have proposed a Random Walk Kernel by comparing pairs of isomorphic random walks in a pair of graphs. Borgwardt et al. [3] have proposed a Shortest Path Kernel by counting the numbers of pairwise shortest paths having the same length in a pair of graphs. Costa and Grave [4] have defined a Neighborhood Subgraph Pairwise Distance Kernel

by counting the number of pairwise isomorphic neighborhood subgraphs. Gaidon et al. [5] have developed a Subtree Kernel for comparing videos, by considering complex actions as decomposed spatio-temporal parts and building corresponding binary trees. The resulting kernel is computed by counting the number of isomorphic subtree patterns. Other alternative graph kernels that are specifically based on the R-convolution framework also include a) the Segmentation Graph Kernel [6], b) the Pyramid Quantized Weisfeiler-Lehman Kernel [7], c) the Subgraph Matching Kernel [8], d) the Quantum-inspired Jensen-Shannon Kernel [9], etc.

One major drawback arising in most existing R-convolution kernels is that they neglect the relative locational information between substructures. Specifically, the R-convolution kernels usually tend to add an unit value when a pair of similar substructures are identified. However, these kernels cannot identify whether these similar substructures are correctly aligned with the overall graph structures, i.e., they do not check if the topological arrangement of the substructures is globally correct. For an instance of a protein matching problem, we may have similar substructures from different parts of the overall structure. R-convolution kernels will count these as being matching substructures, despite the fact that they are not correctly aligned. To overcome this drawback, Bai et al. [10], [11] have developed a family of novel vertex-based matching kernels by aligning depth-based representations of vertices [12]. All these matching kernels can be seen as aligned subgraph or subtree kernels that incorporate explicit structural correspondences, and thus address the drawback of neglecting relative locations between substructures arising in the R-convolution kernels. Unfortunately, these matching kernels are not positive definite in general. This is because the alignment steps for these kernels are not transitive. In other words, if  $\sigma$  is the vertex-alignment between graph  $A$  and graph  $B$ , and  $\pi$  is the alignment between graph  $B$  and graph  $C$ , in general we cannot guarantee that the alignment between graph  $A$  and graph  $C$  is  $\pi \circ \sigma$ . On the other hand, Fröhlich et al. [13] have demonstrated that the transitive alignment step is necessary to guarantee the positive definiteness of the vertex/edge based matching kernels. Furthermore, either the R-convolution kernels or the matching kernels only capture graph characteristics for each pair of graphs, and thus ignore the information over other graphs. As a summary, developing effective graph kernels still remains challenges.

### B. Contributions

The aim of this work is to address the aforementioned shortcomings of existing graph kernels, by developing a new Hier-

Lu Bai, Lixin Cui\* (Corresponding Author: cuilixin@cufe.edu.cn) are with <sup>1</sup>Central University of Finance and Economics, Beijing, China. Edwin R. Hancock is with <sup>3</sup>University of York, York, UK.

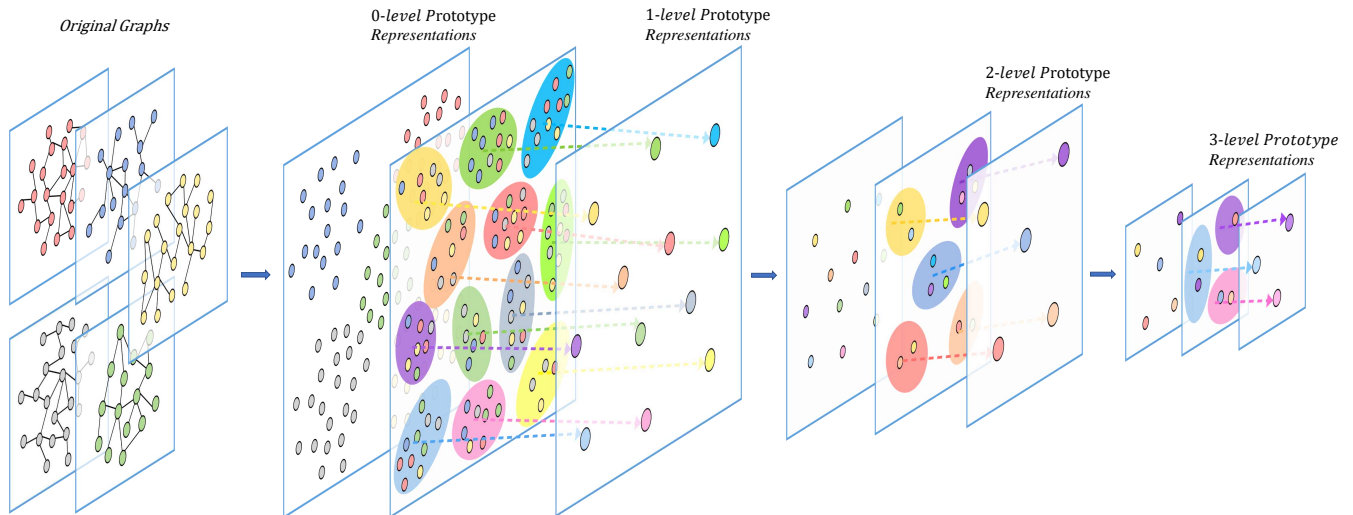


Fig. 1. The framework of constructing the hierarchical prototype representations. For a set of five original graphs, we commence by employing their  $k$ -dimensional vectorial representations as the 0-level prototype representations (in this instance,  $k = 2$ ). By employing the  $\kappa$ -means method, we hierarchically identify a set of centroid points (i.e., means) as the  $h$ -level prototype representations over the set of the last  $h$ -level prototype representations, where  $h$  varies from 1 to 3.

archical Transitive-Aligned Kernel (HTAK) for un-attributed graphs. The key innovation of the proposed kernel is that of transitively aligning vertices between pairs of graphs, through a family of hierarchical prototype representations. That is, given three vertices  $v$ ,  $w$  and  $x$  from three different sample graphs, if  $v$  and  $x$  are aligned, and  $w$  and  $x$  are aligned, the proposed kernel can guarantee that  $v$  and  $w$  are also aligned. As a result, the proposed kernel can theoretically guarantee the positive definiteness. Specifically, the main contributions of this work are threefold.

**First**, we propose a framework to compute a family of  $H$ -hierarchical prototype representations that encapsulate the dominant characteristics of the vectorial vertex representations over a set of graphs  $\mathbf{G}$ . This is achieved by hierarchically performing the  $\kappa$ -means clustering method to identify a pre-assigned number of cluster centroid as the  $h$ -level prototype representations through the last  $h - 1$ -level prototype representations, where the 0-level representations correspond to the original vectorial vertex representations of all graphs. This in turn generate a family of  $H$ -hierarchical prototype representations, when we vary  $h$  from 1 to  $H$  (i.e.,  $1 \leq h \leq H$ ). We show that the new hierarchical prototype representations not only reflect the general structural information over all graphs, but also represent a reliable pyramid of vertices over all graphs at different levels.

**Second**, with the family of  $H$ -hierarchical prototype representations to hand, we develop a graph matching method by hierarchically aligning the vertices of each graph to its different  $h$ -level prototype representations. The resulting HTAK kernel is defined by counting the number of aligned vertex pairs. We show that the proposed kernel not only overcomes the shortcoming of ignoring correspondence information between isomorphic substructures that arises in most existing R-convolution kernels, but also guarantees the transitivity between the correspondence information. As a result, the proposed kernel guarantees positive definite that is not available

in existing alignment kernels [10], [11]. Furthermore, unlike most existing graph kernels, the proposed kernel incorporates the information of all graphs under comparisons into the kernel computation process, and thus encapsulates richer characteristics.

**Third**, by transductively training the C-SVM classifier associated with the proposed HTAK kernel, we empirically demonstrate the effectiveness of the new kernel approach. The proposed kernel can outperform state-of-the-art graph kernels as well as graph neural network models on standard graph datasets in terms of the classification accuracy.

The remainder of this paper is organized as follows. Section II introduces the framework of computing the hierarchical prototype representations. Section III gives the definition of the new kernel, Section IV provides experimental evaluations and Section V concludes the work.

## II. HIERARCHICAL PROTOTYPE REPRESENTATIONS

In this section, we propose a framework to compute a family of  $H$ -hierarchical prototype representations that encapsulate the dominant characteristics over all vectorial vertex representations in a set of graphs  $\mathbf{G}$ . An instance of the proposed framework to compute the hierarchical prototype representations is shown in Fig.1. Specifically, let

$$\mathbf{R}^k = \{\mathbf{R}_1^k, \mathbf{R}_2^k, \dots, \mathbf{R}_i^k, \dots, \mathbf{R}_N^k\}$$

denote the  $k$ -dimensional vectorial representations of  $N$  vertices over all graphs in  $\mathbf{G}$ . We first adopt  $\mathbf{R}^k$  as the set of 0-level prototype representations  $\mathbf{PR}^{0,k}$ , i.e.,

$$\mathbf{PR}^{0,k} = \{\mathbf{PR}_1^{0,k}, \mathbf{PR}_2^{0,k}, \dots, \mathbf{PR}_i^{0,k}, \dots, \mathbf{PR}_{N_0}^{0,k}\}, \quad (1)$$

where all the 0s indicate the current value of the parameter  $h$ ,  $N = N_0$ , and each  $i$ -th element  $\mathbf{PR}_i^{0,k}$  corresponds to  $\mathbf{R}_i^k$ . To compute the set of the higher  $h$ -level (i.e.,  $1 \leq h \leq H$ ) prototype representations  $\mathbf{PR}^{h,k}$ , we employ  $\kappa$ -means [14] to

localize  $N_h$  centroid points over the set of the last lower  $h-1$ -level prototype representations  $\mathbf{PR}^{h-1,k}$ , by minimizing the objective function

$$\arg \min_{\Omega} \sum_{j=1}^{N_h} \sum_{\mathbf{PR}_i^{h-1,k} \in c_j} \|\mathbf{PR}_i^{h-1,k} - \mu_j^k\|_2^2, \quad (2)$$

where  $\Omega = (c_1, c_2, \dots, c_j, \dots, c_{N_h})$  represents  $N_h$  clusters over the set of  $h-1$ -level prototype representations  $\mathbf{PR}^{h-1,k}$ , and  $\mu_j^k$  is the mean of the prototype representations belonging to the  $j$ -th cluster  $c_j$ . We employ the  $N_h$  means  $\{\mu_j^1, \mu_j^2, \dots, \mu_j^k, \dots, \mu_{N_h}^k\}$  as the set of  $h$ -level prototype representations  $\mathbf{PR}^{h,k}$ , i.e.,

$$\mathbf{PR}^{h,k} = \{\mathbf{PR}_1^{h,k}, \mathbf{PR}_2^{h,k}, \dots, \mathbf{PR}_j^{h,k}, \dots, \mathbf{PR}_{N_h}^{h,k}\}, \quad (3)$$

where each  $j$ -th element  $\mathbf{PR}_j^{h,k}$  corresponds to  $\mu_j^k$ , and  $N_h$  corresponds to the number of the  $h$ -layer prototype representations in  $\mathbf{PR}^{h,k}$ .

Since the value of  $N_h$  (i.e.,  $|\mathbf{PR}^{h,k}|$ ) is usually much lower than that of  $N_{h-1}$  (i.e.,  $|\mathbf{PR}^{h-1,k}|$ ), the initialized set of 0-level prototype representations  $\mathbf{PR}^{0,k}$  correspond to the original vectorial representations of the vertices over all graphs in  $\mathbf{G}$ , and the set of  $h$ -level prototype representations  $\mathbf{PR}^{h,k}$  are computed through the objective function of  $\kappa$ -means (i.e., Eq.(2)) that can gradually minimize the inner-vertex-cluster sum of squares over the set of the last  $h-1$ -level prototype representations  $\mathbf{PR}^{h-1,k}$ . When we vary the parameter  $h$  from 1 to  $H$ , this procedure naturally forms a family of  $H$ -hierarchical prototype representations as

$$\mathbb{PR}^{H,k} = \{\mathbf{PR}^{1,k}, \mathbf{PR}^{2,k}, \dots, \mathbf{PR}^{h,k}, \dots, \mathbf{PR}^{H,k}\}, \quad (4)$$

where each  $\mathbf{PR}^{h,k}$  is the set of  $h$ -level prototype representations, and  $\mathbb{PR}^{H,k}$  represents a reliable pyramid of the original vertex representations over all graphs at different levels (i.e., the prototype representations of different  $h$ -levels).

Note that, to compute the family of  $H$ -hierarchical prototype representations, in this work we employ the  $k$ -dimensional depth-based (DB) representations as the original  $k$ -dimensional vectorial vertex representations  $\mathbf{R}^k$  (i.e.,  $\mathbf{PR}_i^{0,k}$ ) to compute the different sets of  $h$ -level prototype representations  $\mathbf{PR}^{h,k}$ . Certainly, computing the vertex representations is an open problem, one can also utilize any other approach to compute the initialized vectorial vertex representations [15], [16]. Specifically, in this work, the specified DB representation of each vertex is defined by measuring the entropies on a family of  $\tilde{k}$ -layer expansion subgraphs rooted at the vertex [12], where  $\tilde{k}$  varies from 1 to  $k$ . Since each  $\tilde{k}$ -layer expansion subgraph completely contains the whole topological structure of the  $k-1$ -layer expansion subgraph, it is shown that such a  $k$ -dimensional DB representation encapsulates rich entropic content flow from each local vertex to the global graph structure, as a function of depth. Fig.2 exhibits the detailed process of computing the DB representation. Specifically, for each sample graph  $G_p(V_p, E_p) \in \mathbf{G}$  indicated by the black color and its  $i$ -th vertex  $v_i$  indicated by the red color in Fig.2, we commence by computing the 1-layer neighborhood set  $\mathcal{N}_i^1$  as

$$\mathcal{N}_i^1 = \{v_j \in V_p \mid s(v_i, v_j) \leq 1\},$$

where  $s(v_i, v_j)$  is the shortest path between the  $i$ -th vertex  $v_i$  and the  $j$ -th vertex  $v_j$ . The resulting 1-layer expansion subgraph  $\mathcal{G}_{p;i}^1$  is defined as the substructure preserving the vertices in  $\mathcal{N}_i^1$  as well as the edges between them from the original global graph  $G_p$ , i.e., the substructures surrounded by the red broken line in Fig.2. Similarly, we also construct the 2-layer and 3-layer expansion subgraphs surrounded by the green and blue broken lines respectively in Fig.2. By parity of reasoning, we generate a family of  $\tilde{k}$ -layer expansion subgraphs rooted at  $v_i$  ( $1 \leq \tilde{k} \leq k$ ). Note that, if  $k$  is greater than the longest shortest path rooted from  $v_i$  to the remaining vertices of  $G_p$ , the  $k$ -layer expansion subgraph  $\mathcal{G}_{p;i}^k$  is the global structure of  $G_p$ . The resulting  $k$ -dimensional DB representation rooted at  $v_i$  is

$$\text{DB}_{p;i}^k = \{H_S(\mathcal{G}_{p;i}^1), \dots, H_S(\mathcal{G}_{p;i}^2), \dots, H_S(\mathcal{G}_{p;i}^k)\}^T,$$

where  $H_S(\cdot)$  is the Shannon entropy of a (sub)graph associated with the steady state random walk [11].

### III. HIERARCHICAL TRANSITIVE-ALIGNED KERNELS

In this section, we propose a novel Hierarchical Transitive-Aligned Kernel (HTAK) for un-attributed graphs. We commence by introducing a new hierarchical transitive vertex matching method, through the family of  $H$ -hierarchical prototype representations. Moreover, we develop the HTAK kernel based on the new vertex matching method.

#### A. Hierarchical Transitive Vertex Matching Methods

In this subsection, we develop a new hierarchical transitive vertex matching method, by hierarchically aligning the vertices of each graph to each set of  $h$ -level prototype representations from the family of  $H$ -hierarchical prototype representations defined in Section II. For a set of  $T$  graphs  $\mathbf{G} = \{G_1, \dots, G_T\}$ , we commence by computing the family of  $H$ -hierarchical prototype representations over the  $k$ -dimensional vectorial vertex representations of all  $T$  graphs as  $\mathbb{PR}^{H,k} = \{\mathbf{PR}^{1,k}, \dots, \mathbf{PR}^{h,k}, \dots, \mathbf{PR}^{H,k}\}$ . To establish the correspondence information between the graph vertices, we align the vectorial vertex representations of a sample graph  $G_p(V_p, E_p) \in \mathbf{G}$  to each set of  $h$ -level prototype representations  $\mathbf{PR}^{h,k} = \{\mathbf{PR}_1^{h,k}, \dots, \mathbf{PR}_n^{h,k}, \dots, \mathbf{PR}_{N_h}^{h,k}\}$ . The alignment process is similar to that introduced in [11] for point matching in a pattern space. Specifically, we compute a  $h$ -level affinity matrix in terms of the Euclidean distances between the two sets of points as

$$R_p^{h,k}(i, n) = \|\mathbf{R}_{p;i}^k - \mathbf{PR}_n^{h,k}\|_2, \quad (5)$$

where  $R_p^{h,k}$  is a  $|V_p| \times N_h$  matrix, and each element  $R_p^{h,k}(i, n)$  represents the distance between the  $k$ -dimensional vectorial representation  $\mathbf{R}_{p;i}^k$  of  $v_i \in V_p$  and the  $n$ -th  $h$ -level prototype representation  $\mathbf{PR}_n^{h,k} \in \mathbf{PR}^{h,k}$ . For the affinity matrix  $R_p^{h,k}$ , the rows index the vertices of  $G_p$ , and the columns index the  $h$ -level prototype representations in  $\mathbf{PR}^{h,k}$ . If  $R_p^{h,k}(i, n)$  is the smallest element in column  $n$ , we say that the  $k$ -dimensional vectorial representation of  $v_i$  is aligned to the  $n$ -th  $h$ -level prototype representation  $\mathbf{PR}_n^{h,k} \in \mathbf{PR}^{h,k}$ .

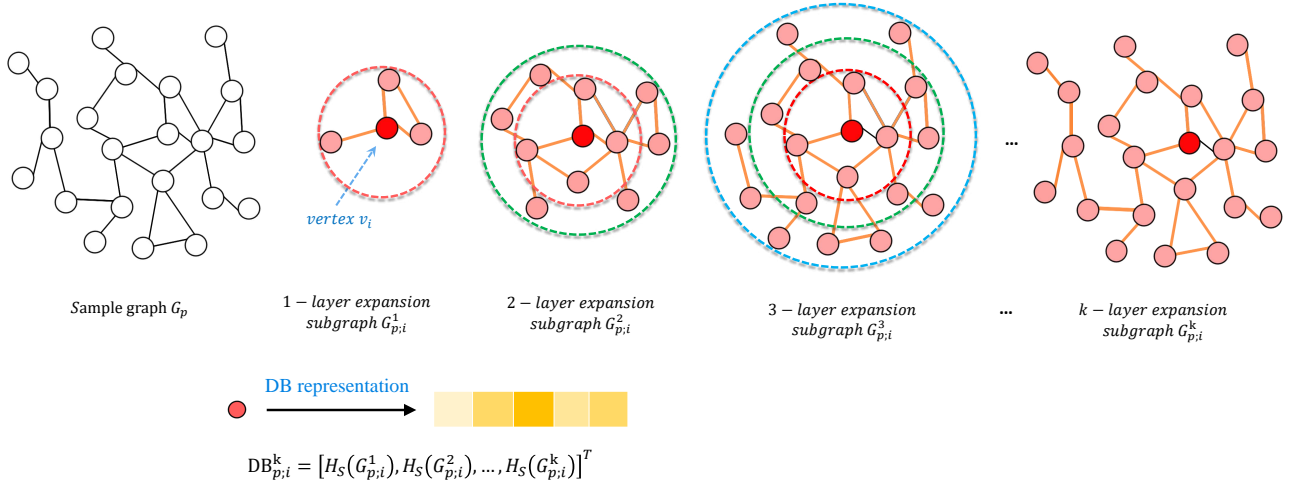


Fig. 2. The process of computing the depth-based representation rooted at each vertex.

Similarly, for each other sample graph  $G_q(V_q, E_q) \in \mathbf{G}$ , we also align its  $k$ -dimensional vectorial representation  $R_{q;j}^k$  of each vertex  $v_j \in V_q$  to each set of  $h$ -level prototype representations  $\mathbf{PR}^{h,k}$ . We compute the element  $R_{q;j}^{h,k}(j, n)$  for the corresponding affinity matrix  $R_q^{h,k}$  as

$$R_q^{h,k}(j, n) = \|R_{q;j}^k - \mathbf{PR}_n^{h,k}\|_2. \quad (6)$$

**Definition (Vertex matching between a pair of graphs):** For the pair of graphs  $G_p$  and  $G_q$  of  $\mathbf{G}$ , if  $R_p^{h,k}(i, n)$  and  $R_q^{h,k}(j, n)$  are both the smallest elements in columns  $n$  of  $R_p^{h,k}$  and  $R_q^{h,k}$  respectively, we say that the vertex  $v_i$  of  $G_p$  and the vertex  $v_j$  of  $G_q$  are aligned, i.e., there is an one-to-one correspondence between  $v_i$  and  $v_j$ . More formally, let the  $h$ -level correspondence matrix  $M_p^{(h,k)} \in \{0, 1\}^{|V_p| \times |N_h|}$  record the state of alignments for  $R_p^{h,k}$ , and

$$M_p^{h,k}(i, n) = \begin{cases} 1 & \text{if } R_p^{h,k}(i, n) \text{ is the smallest} \\ & \text{element in row } n, \text{ and } |\mathcal{S}_i^k| \neq 0; \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Note that,  $\mathcal{S}_i^k$  indicates the set of vertices having the shortest path of length  $k$  to  $v_i$ , and the condition  $|\mathcal{S}_i^k| \neq 0$  guarantees that the  $k$ -layer expansion subgraph rooted at  $v_i$  does not surpass the global structure of  $G_p$  (i.e., the  $k$ -dimensional DB representation of  $v_i$  exists). Similarly, the  $h$ -level correspondence matrix  $M_q^{(h,k)} \in \{0, 1\}^{|V_q| \times |N_h|}$  records the state of alignments for  $R_q^{h,k}$ , and satisfies

$$M_q^{h,k}(j, n) = \begin{cases} 1 & \text{if } R_q^{h,k}(j, n) \text{ is the smallest} \\ & \text{element in row } n; \text{ and } |\mathcal{S}_j^k| \neq 0; \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Based on Eq.(7) and Eq.(8), the  $h$ -level correspondence matrix  $M_{p;q}^{(h,k)} \in \{0, 1\}^{|V_p| \times |V_q|}$ , that records the state of correspondence information between pairwise vertices of  $G_p$  and  $G_q$ , is defined as

$$\mathcal{M}_{p;q}^{(h,k)} = (M_p^{h,k})(M_q^{h,k})^T. \quad (9)$$

For the  $h$ -level correspondence matrix  $\mathcal{M}_{p;q}^{(h,k)}$ , the rows index the vertices of  $G_p$ , and the columns index the the vertices of

$G_q$ . If  $\mathcal{M}_{p;q}^{(h,k)}(i, j) = 1$ , there is an one-to-one correspondence between the vertices  $v_i \in V_p$  and  $v_j \in V_q$ , i.e., we say that they are aligned or matched.  $\square$

Note that, the vertex alignment information identified by  $\mathcal{M}_{p;q}^{(h,k)}$  is transitive, i.e., for three vertices  $u, v$  and  $w$ , if  $u$  and  $v$  are aligned, and  $v$  and  $w$  are aligned, then  $u$  and  $w$  are also aligned. This is because  $\mathcal{M}_{p;q}^{(h,k)}$  identifies the vertex correspondences by evaluating whether the vertices are aligned to the same set of  $h$ -level prototype representations  $\mathbf{PR}^{h,k} \in \mathbb{PR}^{H,k}$ . Finally, by hierarchically aligning each graph to the set of different  $h$ -level prototype representations  $\mathbf{PR}^{h,k}$  from the  $H$ -hierarchical prototype representations  $\mathbb{PR}^{H,k}$ , we obtain a family of  $H$ -hierarchical transitive vertex correspondence matrices between  $G_p$  and  $G_q$  as

$$\mathbf{M}_{p;q}^{(H,k)} = \{\mathcal{M}_{p;q}^{(1,k)}, \dots, \mathcal{M}_{p;q}^{(h,k)}, \dots, \mathcal{M}_{p;q}^{(H,k)}\}. \quad (10)$$

**Remarks:** The procedure of computing the family of  $H$ -hierarchical correspondence matrices  $\mathbf{M}_{p;q}^{(H,k)}$  is **completely unsupervised**, since we do not utilize any class labels of the graphs in  $\mathbf{G}$  during the computational process.

### B. The Hierarchical Transitive-Aligned Kernel

We develop a new Hierarchical Transitive-Aligned Kernel (HTAK) for graphs, based on the  $H$ -hierarchical transitive vertex correspondence matrices between graphs

**Definition (The HTAK kernel):** For the set of graphs  $\mathbf{G}$ , we commence by computing the  $k$ -dimensional DB representations of the vertices over all graphs in  $\mathbf{G}$ , as the 0-level prototype representations  $\mathbf{PR}^{0,k}$ . Based on  $\mathbf{PR}^{0,k}$  and the definition in Section II, we generate a family of  $H$ -hierarchical prototype representations as

$$\mathbb{PR}^{H,k} = \{\mathbf{PR}^{1,k}, \dots, \mathbf{PR}^{h,k}, \dots, \mathbf{PR}^{H,k}\},$$

where  $\mathbf{PR}^{h,k}$  represents the set of  $h$ -level prototype representations, and  $1 \leq h \leq H$ . For a pair of graphs  $G_p$  and  $G_q$  from  $\mathbf{G}$ , by aligning the vertices of  $G_p$  and  $G_q$  to the sets of different  $h$ -level prototype representations  $\mathbf{PR}^{h,k} \in \mathbb{PR}^{H,k}$ ,

we compute the family of  $H$ -hierarchical transitive vertex correspondence matrices as

$$\mathbf{M}_{p;q}^{(H,k)} = \{\mathcal{M}_{p;q}^{(1,k)}, \dots, \mathcal{M}_{p;q}^{(h,k)}, \dots, \mathcal{M}_{p;q}^{(H,k)}\}$$

between  $G_p(V_p, E_p)$  and  $G_q(V_q, E_q)$  based on Eq.(10). With  $\mathbf{M}_{p;q}^{(H,k)}$  to hand, the proposed HTAK kernel  $k_{HTAK}^{(H)}$  between  $G_p$  and  $G_q$  is defined as

$$k_{HTAK}^{(H)}(G_p, G_q) = \sum_{h=1}^H \sum_{k=1}^K \sum_{i=1}^{|V_p|} \sum_{j=1}^{|V_q|} \mathcal{M}_{p;q}^{(h,k)}(i, j), \quad (11)$$

where  $K$  is the greatest value of the parameter  $k$  (i.e.,  $k$  varies from 1 to  $K$ ). As we have stated in Section II, the parameter  $k$  indicates the dimension of the vectorial vertex representations, and we propose to employ the  $k$ -dimensional DB representations of vertices as the vectorial vertex representations [12]. Since the DB representations are computed based on the  $\tilde{k}$ -layer expansion subgraphs ( $1 \leq \tilde{k} \leq k$ ), the greatest value  $K$  of the parameter  $k$  corresponds to that of the longest shortest path between vertices over all graphs in  $\mathbf{G}$ . Eq.(11) indicates that  $k_{HTAK}^{(H)}(G_p, G_q)$  counts the number of aligned vertex pairs between  $G_p$  and  $G_q$  over all the  $h$ -level vertex correspondence matrices  $\mathcal{M}_{p;q}^{(h,k)} \in \mathbf{M}_{p;q}^{(H,k)}$ .  $\square$

**Lemma.** *The kernel  $k_{HTAK}^{(H)}$  is positive definite (pd).*

**Proof.** Intuitively, the proposed HTAK kernel  $k_{HTAK}^{(H)}$  is **pd**, since it counts pairs of aligned vertices over the  $H$  correspondence matrices  $\mathcal{M}_{p;q}^{(h,k)} \in \mathbf{M}_{p;q}^{(H,k)}$  and the correspondence information identified by the proposed kernel is transitive. More formally, for the graph  $G_p \in \mathbf{G}$ , let  $F^{(h,k)}(G_p)$  be a  $N_h$ -dimensional feature vector that counts the number of vertices aligned to the corresponding  $h$ -level prototype representations  $\mathbf{PR}^{h,k} \in \mathbb{P}\mathbb{R}^{H,k}$ , and

$$F^{(h,k)}(G_p) = \left[ \sum_{i=1}^{|V_p|} M_p^{h,k}(i, 1), \dots, \sum_{i=1}^{|V_p|} M_p^{h,k}(i, n), \dots, \sum_{i=1}^{|V_p|} M_p^{h,k}(i, N_h) \right]^T, \quad (12)$$

where the  $n$ -th element  $\sum_{i=1}^{|V_p|} M_p^{h,k}(i, n)$  of  $F^{(h,k)}(G_p)$  counts the number of vertices (from  $G_p$ ) that are all aligned to the  $n$ -th  $h$ -level prototype representation  $\mathbf{PR}_n^{h,k} \in \mathbf{PR}^{h,k}$ , and  $M_p^{h,k}(i, n)$  is defined by Eq.(7). Similarly, for the graph  $G_q$ , we have the feature vector  $F^{(h,k)}(G_q)$  as

$$F^{(h,k)}(G_q) = \left[ \sum_{j=1}^{|V_q|} M_q^{h,k}(j, 1), \dots, \sum_{j=1}^{|V_q|} M_q^{h,k}(j, n), \dots, \sum_{j=1}^{|V_q|} M_q^{h,k}(j, N_h) \right]^T, \quad (13)$$

Based on Eq.(12) and Eq.(13), the HTAK kernel  $k_{HTAK}^{(H)}$  defined by Eq.(11) can be re-written as

$$k_{HTAK}^{(H)}(G_p, G_q) = \sum_{h=1}^H \sum_{k=1}^K \langle F^{(h,k)}(G_p), F^{(h,k)}(G_q) \rangle, \quad (14)$$

where  $\langle F^{(h,k)}(G_p), F^{(h,k)}(G_q) \rangle$  is an inner product, i.e., it is a **pd** linear kernel. As a result, the kernel  $k_{HTAK}^{(H)}$  can be seen as a kernel that sums the linear kernels  $\langle F^{(h,k)}(G_p), F^{(h,k)}(G_q) \rangle$ , and is thus **pd**.  $\blacksquare$

### C. Discussions of the Proposed Kernel

The new vertex alignment kernel  $k_{HTAK}^{(H)}$  has some important properties that are not available for some existing state-of-the-art graph kernels.

**First**, unlike the existing alignment kernels [13], [11], [10], [17], [18] that can also identify correspondence information between vertices or edges, the aligned vertices identified by the proposed HTAK kernel  $k_{HTAK}^{(H)}$  are transitive. This is because, as we have stated in Section III-A, the vertex alignment method employed in the proposed kernel can transitively align vertices between graphs. As a result, the proposed HTAK kernel  $k_{HTAK}^{(H)}$  not only overcomes the shortcoming of ignoring structural correspondences arising in most R-convolution kernels, but also reflects more precise correspondence information than the existing alignment or matching kernels [13], [11], [10], [17], [19], [18].

**Second**, as Fröhlich et al. [13] have stated, the transitive alignment step is necessary to guarantee the positive definiteness of alignment kernels. Thus, the proposed HTAK kernel  $k_{HTAK}^{(H)}$  guarantees the positive definiteness that is not available to the aforementioned alignment kernels [13], [11], [10], [17], [19], [18].

**Third**, the computation of the proposed HTAK kernel  $k_{HTAK}^{(H)}$  for a pair of graphs incorporates the information over all graphs under comparisons. This is because  $k_{HTAK}^{(H)}$  is computed by hierarchically aligning the vertices of each graph to the different  $h$ -level prototype representations of the family of  $H$ -hierarchical prototype representations, that is hierarchically identified by  $\kappa$ -means method over the  $k$ -dimensional vectorial vertex representations over all graphs in  $\mathbf{G}$ , i.e.,  $k_{HTAK}^{(H)}$  is not only computed though each individual pair of graphs. By contrast, most existing graph kernels only capture graph characteristics for each pair of graphs [3], [20], [21], [22], [23], [4]. As a result, the proposed kernel  $k_{HTAK}^{(H)}$  may reflect richer graph characteristics.

Finally, note that, since the basics of the proposed kernel  $k_{HTAK}^{(H)}$  is based on  $k$ -dimensional DB representations of vertices than do not encapsulate any vertex or edge label information. The proposed kernel  $k_{HTAK}^{(H)}$  cannot accommodate the vertex or edge label information. However, we can still perform  $k_{HTAK}^{(H)}$  on attributed graphs by focusing on topological information without vertex/edge labels.

### D. Computational Analysis

For the set of  $T$  graphs  $\mathbf{G}$  each of which has  $n$  vertices and  $m$  edges, computing the proposed kernel  $k_{HTAK}^{(H)}$  requires time complexity  $O(HIN_1Tn + HT^2N_1 + HTN_1n + Tn \log n + Tmn)$ , where  $H$  corresponds to the set number of different  $h$ -level prototype representations from the family of  $H$ -hierarchical prototype representations,  $I$  is the iteration number for the  $\kappa$ -means method, and  $N_1$  is the number of

1-level prototype representations in  $\mathbf{PR}^{1,k}$ . This is because computing the required  $k$ -dimensional DB representations of vertices (i.e., the 0-level prototype representations  $\mathbf{PR}^{0,k}$ ) relies on the shortest path computation on each graph, and thus requires time complexity  $O(Tn \log n + Tmn)$ . Computing the  $N_h$   $h$ -level prototype representations of  $\mathbf{PR}^{h,k}$  relies on  $\kappa$ -means method on the last  $N_{h-1}$   $h-1$ -layer prototype representations of  $\mathbf{PR}^{h-1,k}$ . Since  $N_1 \gg N_2 \gg \dots \gg N_H$ , the whole process requires time complexity  $O(HN_1Tn)$ . Calculating the kernel value between graphs relies on computing the  $k$ -level correspondence matrix  $M_p^{(h,k)} \in \{0, 1\}^{|V_p| \times N_h}$  in terms of each set of  $h$ -level prototype representations  $\mathbf{PR}^{h,k}$ , and counting the number of vertices of each graph aligning to the  $N_h$  prototype representations in  $\mathbf{PR}^{h,k}$ . Since  $N_1 \gg N_2 \gg \dots \gg N_H$ , the whole process requires time complexity  $O(HT^2N_1 + HTN_1n)$ . As a result, the whole time complexity of computing the proposed kernel  $k_{\text{HTAK}}^{(H)}$  over all  $T$  graphs of  $\mathbf{G}$  requires time complexity  $O(HN_1Tn + HT^2N_1 + HTN_1n + Tn \log n + Tmn)$ . Note that, in this work, we employ the fastest  $K$ -means MATLAB implementation developed by Deng Cai [24], and the default number of  $I$  is 100. Moreover, in this work, most graphs are sparse graphs (i.e.,  $n < m \ll n^2$ ) and  $H$  is set as 5. As a result, the whole time complexity is approximately  $O(N_1Tn + T^2N_1 + TN_1n + Tn^2)$ , indicating that our kernel can usually be computed in a polynomial time.

#### IV. EXPERIMENTS

We evaluate the proposed HTAK kernels on nine benchmark graph datasets from computer vision, bioinformatics, and social networks. These datasets include: BAR31, BSPHERE31, GEOD31, MUTAG, NCI1, CATH2, COLLAB, IMDB-B, and IMDB-M. Here the BAR31, BSPHERE31 and GEOD31 datasets are all abstracted from the SHREC 3D Shape database, that consists of 15 classes and 20 individuals per class [25]. Specifically, we establish the BAR31, BSPHERE31 and GEOD31 datasets through three mapping functions, i.e., a) ERG barycenter: distance from the center of mass/barycenter, b) ERG bsphere: distance from the center of the sphere that circumscribes the object, and c) ERG integral geodesic: the average of the geodesic distances to the all other points. On the other hand, other datasets are all available on the website <http://graphkernels.cs.tu-dortmund>. More details of these datasets are shown in Table.I.

##### A. Experiments on Graph Classification

**Experimental Setup:** We evaluate the performance of the proposed HTAK kernel in terms of graph classification problems on the aforementioned nine benchmark graph datasets. We also compare our kernel with a) five alternative state-of-the-art graph kernels and b) four alternative state-of-the-art deep learning methods for graphs. Specifically, **the graph kernels include** 1) the aligned subtree kernel (ASK) [11], 2) the Weisfeiler-Lehman subtree kernel (WLSK) [26], 3) the shortest path graph kernel (SPGK) [3], 4) the graphlet count graph kernel [27] with graphlet of size 4 (GCGK), and 5) the Jensen-Tsallis  $q$ -difference kernel (JTQK) [28] with  $q = 2$ .

On the other hand, **the deep learning methods include** 1) the deep graph convolutional neural network (DGCNN) [29], 2) the PATCHY-SAN based convolutional neural network for graphs (PSGCNN) [30], 3) the diffusion convolutional neural network (DCNN) [31], and 4) the deep graphlet kernel (DGK) [32].

For the WLSK kernel and the JTQK kernel, we set the highest dimension (i.e., the highest height of subtrees) of the Weisfeiler-Lehman isomorphism (for the WLSK kernel) and the tree-index method (for the JTQK kernel) as 10, based on the statements of the authors in [28], [26]. For the ASK kernel, we set the highest layer of the required DB representation as 50 based on [11], to guarantee the best performance. For each kernel, we compute the kernel matrix on each graph dataset. We perform a 10-fold cross-validation where the classification accuracy is computed using a C-Support Vector Machine (C-SVM). In particular, we make use of the LIBSVM library [33]. For each dataset and each kernel, we compute the optimal C-SVMs parameters. We repeat the whole experiment 10 times and report the average classification accuracy ( $\pm$  standard error) in Table II. Note that, for the proposed HTAK kernel we vary the parameter  $H$  from 1 to 5. Thus, for each dataset we compute 5 kernel matrices for the HTAK kernel. The classification accuracy for each time is thus the average accuracy over the 5 kernel matrices. Moreover, for the proposed HTAK kernel on each dataset, we set the parameter  $N_h$  as  $N_h = 0.2N_{h-1}$ , where  $h$  varies from 1 to 5 and  $N_0$  corresponds to the vertex number over all graphs in the dataset.

For the alternative deep learning methods, we report the best results for the DGCNN, PSGCNN, DCNN, DGK models from their original papers. Moreover, note that the PSGCNN model can leverage additional edge features, most of the graph datasets and the alternative methods do not leverage edge features. Thus, we do not report the results associated with edge features in the evaluation. The classification accuracies and standard errors for each deep learning method are shown in Table.III. Finally, note that, as we have stated in Section III-B, the computation of the HTAK kernel for a pair of graphs incorporates the information over all graphs under comparisons. Thus the proposed HTAK kernel can also incorporate the test graphs into the training process of C-SVMs. In this sense, the proposed HTAK kernel can be seen as an instance of **transductive learning** [34] (i.e., we transductively train the C-SVM), where all the graphs available (both from the training and test sets) are used to compute the graph centroid representations. However, note that we **do not** observe the class labels of the test graphs during the training. Finally, note that, some methods are not evaluated by the original authors on some datasets, thus we do not exhibit these results.

**Results and Discussions:** In terms of the classification accuracy, we observe that our HTAK kernel can outperform the alternative graph kernels and deep learning methods on most datasets. For the alternative graph kernel methods, only the accuracies of the ASK kernel on the BAR31 and MUTAG datasets, and the accuracy of the SPGK kernel on the IMDB-M dataset as well as that of the JTQK kernel on the NCI1 dataset are higher than the proposed HTAK kernel. On the

TABLE I  
INFORMATION OF THE GRAPH BASED COMPUTER VISION (CV), BIOINFORMATICS (BIO), AND SOCIAL NETWORK (SN) DATASETS

| Datasets        | BAR31 | BSPHERE31 | GEOD31 | MUTAG | NCII  | CATH2  | COLLAB | IMDB-B | IMDB-M |
|-----------------|-------|-----------|--------|-------|-------|--------|--------|--------|--------|
| Max # vertices  | 220   | 227       | 380    | 28    | 111   | 568    | 492    | 136    | 89     |
| Mean # vertices | 95.42 | 99.83     | 57.42  | 17.93 | 29.87 | 308.03 | 74.49  | 19.77  | 13.00  |
| # graphs        | 300   | 300       | 300    | 188   | 4110  | 190    | 5000   | 1000   | 1500   |
| # classes       | 15    | 15        | 15     | 2     | 2     | 2      | 3      | 2      | 3      |
| Description     | CV    | CV        | CV     | Bio   | Bio   | Bio    | SN     | SN     | SN     |

TABLE II  
CLASSIFICATION ACCURACY (IN %  $\pm$  STANDARD ERROR) FOR COMPARISONS WITH GRAPH KERNELS.

| Datasets | BAR31           | BSPHERE31       | GEOD31          | MUTAG            | NCII            | CATH2            | COLLAB           | IMDB-B           | IMDB-M           |
|----------|-----------------|-----------------|-----------------|------------------|-----------------|------------------|------------------|------------------|------------------|
| HTAK     | 71.00 $\pm$ .45 | 62.90 $\pm$ .65 | 47.80 $\pm$ .49 | 87.32 $\pm$ .60  | 79.01 $\pm$ .14 | 87.89 $\pm$ .71  | 79.87 $\pm$ 0.15 | 72.89 $\pm$ 0.56 | 50.23 $\pm$ 0.18 |
| ASK      | 73.10 $\pm$ .67 | 60.30 $\pm$ .44 | 46.21 $\pm$ .69 | 87.50 $\pm$ .65  | 78.47 $\pm$ .12 | 78.52 $\pm$ .67  | 77.53 $\pm$ 0.31 | 70.38 $\pm$ 0.72 | 50.12 $\pm$ 0.51 |
| WLSK     | 58.53 $\pm$ .53 | 42.10 $\pm$ .68 | 38.20 $\pm$ .68 | 82.88 $\pm$ 0.57 | 84.77 $\pm$ .13 | 67.36 $\pm$ .63  | 77.39 $\pm$ 0.35 | 71.88 $\pm$ 0.77 | 49.50 $\pm$ 0.49 |
| SPGK     | 55.73 $\pm$ .44 | 48.20 $\pm$ .76 | 38.40 $\pm$ .65 | 83.38 $\pm$ 0.81 | 74.21 $\pm$ .30 | 81.89 $\pm$ .63  | 58.80 $\pm$ 0.2  | 71.26 $\pm$ 1.04 | 51.33 $\pm$ 0.57 |
| GCGK     | 23.40 $\pm$ .60 | 18.80 $\pm$ .50 | 22.36 $\pm$ .55 | 82.04 $\pm$ .39  | 63.72 $\pm$ .12 | 73.68 $\pm$ 1.09 | —                | —                | —                |
| JTQK     | 60.56 $\pm$ .35 | 46.93 $\pm$ .61 | 40.10 $\pm$ .46 | 85.50 $\pm$ .55  | 85.32 $\pm$ .14 | 68.70 $\pm$ .69  | 76.85 $\pm$ 0.40 | 72.45 $\pm$ 0.81 | 50.33 $\pm$ 0.49 |

TABLE III  
CLASSIFICATION ACCURACY (IN %  $\pm$  STANDARD ERROR) FOR COMPARISONS WITH DEEP LEARNING METHODS.

| Datasets | MUTAG            | NCII             | COLLAB           | IMDB-B           | IMDB-M           |
|----------|------------------|------------------|------------------|------------------|------------------|
| HTAK     | 87.32 $\pm$ .60  | 79.01 $\pm$ .14  | 79.87 $\pm$ 0.15 | 72.89 $\pm$ 0.56 | 50.23 $\pm$ 0.18 |
| DGCNN    | 85.83 $\pm$ 1.66 | 74.44 $\pm$ .47  | 73.76 $\pm$ 0.49 | 70.03 $\pm$ 0.86 | 47.83 $\pm$ 0.85 |
| PSGCNN   | 88.95 $\pm$ 4.37 | 76.34 $\pm$ 1.68 | 72.60 $\pm$ 2.15 | 71.00 $\pm$ 2.29 | 45.23 $\pm$ 2.84 |
| DCNN     | 66.98            | 56.61 $\pm$ 1.04 | 52.11 $\pm$ 0.71 | 49.06 $\pm$ 1.37 | 33.49 $\pm$ 1.42 |
| DGK      | 82.66 $\pm$ 1.45 | 62.48 $\pm$ .25  | 73.09 $\pm$ 0.25 | 66.96 $\pm$ 0.56 | 44.55 $\pm$ 0.52 |

other hand, for the alternative deep learning methods, only the PSGCNN model on the MUTAG dataset is higher than the proposed HTAK kernel.

In fact, the WLSK, ASK and JTQK kernels, as well as the alternative deep learning approaches can all accommodate the vertex label information, i.e., they can accommodate attributed graphs. By contrast, the proposed HTAK kernel is designed for un-attributed graphs and can not associate with any vertex label information. On the other hand, only these deep learning methods can provide an end-to-end learning framework for graph classification. By contrast, the proposed HTAK kernel associated with the C-SVM can only provide a shallow learning framework. However, even under such disadvantageous situations, the proposed HTAK kernel can still outperform these methods on most datasets. This indicates that the proposed kernel can learn better topological characteristics of graphs than the remaining alternative methods, through the family of  $H$ -hierarchical prototype representations that represent a reliable pyramid of the original vertex representations over all graphs at different levels (i.e., the prototype representations of different  $h$ -levels).

The reasons for the effectiveness are fourfold. **First**, unlike the alternative WLSK, SPGK, GCGK and JTQK kernels that ignore the correspondences information between substructures, the proposed HTAK kernel can hierarchically identify the vertex correspondence information through the  $H$ -hierarchical prototype representations. **Second**, compared to the ASK kernel, the correspondence information identified by the HTAK kernel are transitive. By contrast, the ASK kernel cannot guarantee the transitivity. As a result, the HTAK kernel can capture more precise information for graphs than the ASK kernel. **Third**, unlike alternative kernels, only the proposed kernel incorporates the information of all graphs under comparisons into the kernel computation. The HTAK kernel thus reflects richer graph characteristics. **Fourth**, similar to the WLSK, SPGK, GCGK and JTQK kernels, all the alternative deep

learning methods also do not associate with the structural correspondence information into the learning framework. Overall, the above observations demonstrate the effectiveness of the proposed HTAK kernel.

## V. CONCLUSIONS

In this paper, we develop a new Hierarchical Transitive-Aligned kernel for graphs, that can transitively align the vertices between graphs through a family of  $H$ -hierarchical prototype graphs. Unlike most state-of-the-art graph kernels, this kernel not only overcomes the shortcoming of ignoring correspondence information between graphs, but also guarantees the transitivity between the correspondence information. Experimental evaluations have demonstrated the effectiveness of the proposed new transitive aligned kernel. The proposed kernel can outperform state-of-the-art graph kernels as well as the deep learning methods in terms of graph classifications.

Our future work is to further extend the proposed kernel for attributed graphs, so that the proposed kernel can accommodate the vertex label information into the computation, improving the performance of the proposed kernel.

## ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant no. 61976235 and 61602535), the Open Projects Program of the National Laboratory of Pattern Recognition (NLPR), and the program for innovation research in Central University of Finance and Economics and the Youth Talent Development Support Program by Central University of Finance and Economics, No. QYP1908.

## REFERENCES

- [1] D. Haussler, "Convolution kernels on discrete structures," in *Technical Report UCS-CRL-99-10*, Santa Cruz, CA, USA, 1999.
- [2] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," in *Proceedings of ICML*, 2003, pp. 321–328.



- [3] 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.
- [4] F. Costa and K. D. Grave, "Fast neighborhood subgraph pairwise distance kernel," in *Proceedings of ICML*, 2010, pp. 255–262.
- [5] A. Gaidon, Z. Harchaoui, and C. Schmid, "A time series kernel for action recognition," in *Proceedings of BMVC*, 2011, pp. 1–11.
- [6] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," in *Proceedings of CVPR*, 2007.
- [7] K. Gkirtzou and M. B. Blaschko, "The pyramid quantized weisfeiler-lehman graph representation," *Neurocomputing*, vol. 173, pp. 1495–1507, 2016.
- [8] N. Kriege and P. Mutzel, "Subgraph matching kernels for attributed graphs," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.
- [9] L. Bai, L. Rossi, L. Cui, J. Cheng, Y. Wang, and E. R. Hancock, "A quantum-inspired similarity measure for the analysis of complete weighted graphs," *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 1264 – 1277, 2020.
- [10] L. Bai, Z. Zhang, C. Wang, X. Bai, and E. R. Hancock, "A graph kernel based on the jensen-shannon representation alignment," in *Proceedings of IJCAI*, 2015, pp. 3322–3328.
- [11] 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.
- [12] L. Bai and E. R. Hancock, "Depth-based complexity traces of graphs," *Pattern Recognition*, vol. 47, no. 3, pp. 1172–1186, 2014.
- [13] 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.
- [14] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.
- [15] R. C. Wilson, E. R. Hancock, and B. Luo, "Pattern vectors from algebraic graph theory," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1112–1124, 2005.
- [16] X. Bai, E. R. Hancock, and R. C. Wilson, "Graph characteristics from the heat kernel trace," *Pattern Recognition*, vol. 42, no. 11, pp. 2589–2606, 2009.
- [17] L. Bai, Z. Zhang, P. Ren, L. Rossi, and E. R. Hancock, "An edge-based matching kernel through discrete-time quantum walks," in *Proceedings of ICIAP*, 2015, pp. 27–38.
- [18] M. Neuhaus and H. Bunke, "Edit distance-based kernel functions for structural pattern classification," *Pattern Recognition*, vol. 39, no. 10, pp. 1852–1863, 2006.
- [19] L. Bai, P. Ren, X. Bai, and E. R. Hancock, "A graph kernel from the depth-based representation," in *Proceedings of S+SSPR*, 2014, pp. 1–11.
- [20] F. R. Bach, "Graph kernels between point clouds," in *Proceedings of ICML*, 2008, pp. 25–32.
- [21] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: hardness results and efficient alternatives," in *Proceedings of COLT*, 2003, pp. 129–143.
- [22] 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.
- [23] 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.
- [24] D. Cai, "The fastest matlab implementation of the  $k$ -means," in *Software available at <http://www.zjucadcg.cn/dengcai/Data/Clustering.html>*, 2012.
- [25] S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, and B. Falcidieno, "3d shape matching through topological structures," in *Proceedings of DGCI*, 2003, pp. 194–203.
- [26] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *Journal of Machine Learning Research*, vol. 12, pp. 2539–2561, 2011.
- [27] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," *Journal of Machine Learning Research*, vol. 5, pp. 488–495, 2009.
- [28] L. Bai, L. Rossi, H. Bunke, and E. R. Hancock, "Attributed graph kernels using the jensen-tsallis  $q$ -differences," in *Proceedings of ECML-PKDD*, 2014, pp. 1:99–114.
- [29] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proceedings of AAAI*, 2018.
- [30] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of ICML*, 2016, pp. 2014–2023.
- [31] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proceedings of NIPS*, 2016, pp. 1993–2001.
- [32] P. Yanardag and S. V. N. Vishwanathan, "Deep graph kernels," in *Proceedings of KDD*, 2015, pp. 1365–1374.
- [33] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>*, 2011.
- [34] A. Gammerman, V. Vovk, and V. Vapnik, "Learning by transduction," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 148–155.