

---

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

---

Lu Bai<sup>1,2</sup> Lixin Cui<sup>2</sup> Edwin R. Hancock<sup>3</sup>

## 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. Experimental evaluations demonstrate the effectiveness of the new transitive-aligned kernel.

## 1. 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 (Borgwardt et al., 2020).

Generally speaking, a graph kernel is defined in terms of a similarity measure between graph structures (Long et al., 2021). 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

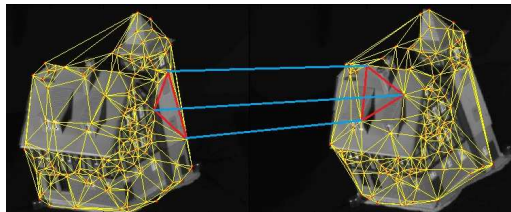


Figure 1. Graphs abstracted from digital images of different angles, specific isomorphic substructures (Haussler, 1999). Specifically, these graph kernels are the so-called R-convolution kernels, and any graph decomposition can be used to define such a kernel, e.g., the graph kernel based on comparing all pairs of decomposed a) walks (Sugiyama & Borgwardt, 2015), b) paths (Aziz et al., 2013), and c) restricted subgraphs or subtrees (Azaïs & Ingels, 2020). With this scenario, Kashima et al. (Kashima et al., 2003) have proposed a Random Walk Kernel by comparing pairs of isomorphic random walks. Borgwardt et al. (Borgwardt & Kriegel, 2005) have proposed a Shortest Path Kernel by counting the numbers of pairwise shortest paths having the same length. Costa and Grave (Costa & Grave, 2010) have defined a Neighborhood Subgraph Pairwise Distance Kernel by counting the number of pairwise isomorphic neighborhood subgraphs. Gaidon et al. (Gaidon et al., 2011) 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 R-convolution graph kernels also include a) the Segmentation Graph Kernel (Harchaoui & Bach, 2007), b) the Pyramid Quantized Weisfeiler-Lehman Kernel (Gkirtzou & Blaschko, 2016), c) the Subgraph Matching Kernel (Kriege & Mutzel, 2012), d) the Wasserstein Weisfeiler-Lehman Kernel (Togninalli et al., 2019), e) the Isolation Graph Kernel (Xu et al., 2021a), f) the Quantum-inspired Jensen-Shannon Kernel (Bai et al., 2020b), 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

<sup>1</sup>School of Artificial Intelligence, Beijing Normal University, Beijing, China. <sup>2</sup>Central University of Finance and Economics, Beijing, China. <sup>3</sup>Department of Computer Science, University of York, York, UK. Correspondence to: Lixin Cui <cuilixin@cufe.edu.cn>.

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. Moreover, for an instance of graph-based image analysis problem, there are two graphs abstracted from two digital images in Fig. 1 and both contain the same house object from different viewpoints. R-convolution kernels will identify the two isomorphic subgraphs consisting of red lines and thus contribute an unit value to the kernel, although they are not locationally or structurally aligned based on the image background, i.e., the vertices connected by blue lines are not aligned.

To overcome the above drawback, Bai et al. (Bai et al., 2015a;b; Xu et al., 2021b) have developed a family of vertex-based matching kernels by aligning depth-based representations of vertices. 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. (Fröhlich et al., 2005) 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.

The aim of this work is to address the aforementioned shortcomings of existing graph kernels, by developing a new Hierarchical 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 preassigned 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 generates 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 (Bai et al., 2015b;a). 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.

This paper is organized as follows. Section 2 introduces the hierarchical prototype representations. Section 3 defines the new kernel, Section 4 provides experimental evaluations and Section 5 concludes this work.

## 2. 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.2. 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  $\mathbf{R}_i^k$ .

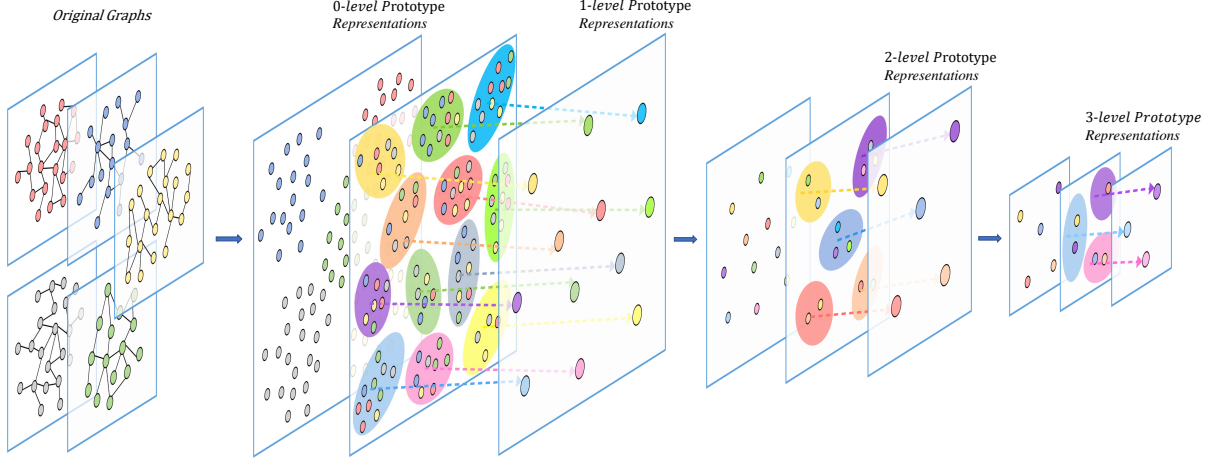


Figure 2. The framework of constructing the hierarchical prototype representations. For a set of five original graphs, we commence by employing their  $k$ -dimensional vectorial representations as the 0-level prototype representations (in this instance,  $k = 2$ ). By employing the  $\kappa$ -means method,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

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 (Witten et al., 2011) 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 (Wilson et al., 2005; Bai et al., 2009). 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 (Bai et al., 2015a), where  $\tilde{k}$  varies from 1 to  $k$ . Since each  $\tilde{k}$ -layer expansion subgraph completely contains the whole topological structure of the  $\tilde{k}-1$  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.3 exhibits the detailed process of computing the DB representation.

### 3. The Definition of the Proposed Kernel

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.

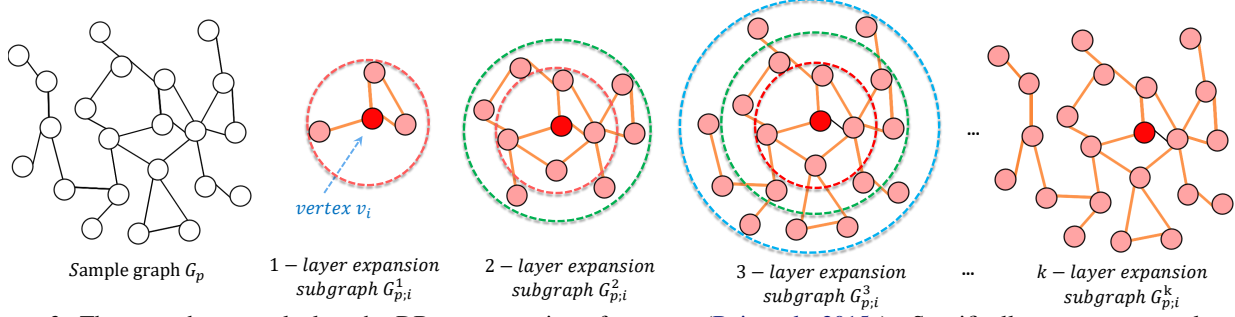


Figure 3. The procedure to calculate the DB representation of a vertex (Bai et al., 2015a). Specifically, assume a sample graph  $G_p(V_p, E_p) \in \mathbf{G}$  marked by black color,  $v_i$  is its  $i$ -th vertex (marked by red color). For  $v_i$ , we first compute its 1-th order neighborhood set  $\mathcal{N}_i^1$  as  $\mathcal{N}_i^1 = \{v_j \in V_p \mid d_s(v_i, v_j) \leq 1\}$ , where  $d_s(v_i, v_j)$  represents the shortest path length between  $j$ -th vertex  $v_j$  and  $i$ -th vertex  $v_i$ . The resulting 1-layer expansion subgraph  $\mathcal{G}_{p,i}^1$  rooted at  $v_i$  is defined as the substructures (surrounded by the red broken line) associated with the vertices in  $\mathcal{N}_i^1$  and the edges between the vertices from the original graph  $G_p$ . Analogously, we also abstract the 2-layer subgraph  $\mathcal{G}_{p,i}^2$  (surrounded by the green line) and the 3-layer expansion subgraph  $\mathcal{G}_{p,i}^3$  (surrounded by the blue broken line), respectively. Consequently, we establish a family of  $\tilde{k}$ -layer expansion subgraphs rooted at  $v_i$  ( $\tilde{k} \in [1, k]$ ). Clearly, if  $k$  is greater than the length of the longest shortest path rooted from  $v_i$  to the remaining vertices, the  $k$ -layer expansion subgraph  $\mathcal{G}_{p,i}^k$  is the global graph  $G_p$  itself. As a result, the  $k$ -dimensional DB representation of  $v_i$  is defined as  $\text{DB}_{p,i}^k = \{H_S(\mathcal{G}_{p,i}^1), \dots, H_S(\mathcal{G}_{p,i}^{\tilde{k}}), \dots, H_S(\mathcal{G}_{p,i}^k)\}^T$ , where  $H_S(\cdot)$  is the classical Shannon entropy of a subgraph based on classical random walks (Bai et al., 2015a).

### 3.1. 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 2. 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  $\mathbf{PR}^{H,k} = \{\mathbf{PR}_1^{1,k}, \dots, \mathbf{PR}_n^{h,k}, \dots, \mathbf{PR}_{N_h}^{H,k}\}$ . 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 (Bai et al., 2015a) 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

$$R_p^{h,k}(i, n) = \|R_{p,i}^k - \text{PR}_n^{h,k}\|_2, \quad (5)$$

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  $R_{p,i}^k$  of  $v_i \in V_p$  and the  $n$ -th  $h$ -level prototype representation  $\text{PR}_n^{h,k} \in \mathbf{PR}^{h,k}$ . 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  $\text{PR}_n^{h,k} \in \mathbf{PR}^{h,k}$ .

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  $R_q^{h,k}(j, n)$  for the corresponding affinity matrix  $R_q^{h,k}$  as

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

#### 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 a 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 a 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  $M_{p;q}^{(h,k)}$  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.

### 3.2. 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):** Based on the definition in Section 2, for a set of graphs  $\mathbf{G}$  we commence by computing the 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 2, 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 (Bai et al., 2015a). 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  $F_{G_p}^{(h,k)}$  be a  $N_h$ -dimensional feature vector that counts the number of vertices aligned to the corresponding  $h$ -level prototype representations  $\mathbf{PR}^{h,k} \in \mathbb{PR}^{H,k}$ , and

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

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

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

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

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

where  $\langle F_{G_p}^{(h,k)}, F_{G_q}^{(h,k)} \rangle$  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  $\langle F_{G_p}^{(h,k)}, F_{G_q}^{(h,k)} \rangle$ , and is thus **pd**.  $\blacksquare$

**Remarks:** The proposed kernel  $k_{\text{HTAK}}^{(H)}$  relies on the prototype representations that are identified by the classical  $\kappa$ -means method. As we know,  $\kappa$ -means tends to randomly pick predefined number of points as the initial centroids. However, this will not influence the robustness of the proposed kernel. This is because the proposed kernel is defined based on the  $k$ -dimensional ( $k = 1, 2, \dots, K$ ) D-B representations and thus needs to perform  $\kappa$ -means for  $K$  times, when vary  $k$  from 1 to  $K$  under each  $h$ -level ( $h = 1, 2, \dots, H$ ) alignment. As a result, this kernel will perform  $\kappa$ -means for  $KH$  time at least, avoiding the influence of the randomness from  $\kappa$ -means.

### 3.3. Discussions of the Proposed Kernel

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

**First**, unlike the existing alignment kernels (Fröhlich et al., 2005; Bai et al., 2015a;b; Neuhaus & Bunke, 2006) that can also identify correspondence information between vertices or edges, the aligned vertices identified by the proposed HTAK kernel  $k_{\text{HTAK}}^{(H)}$  are transitive. This is because, as we have stated in Section 3.1, the vertex alignment method employed in the proposed kernel can transitively align vertices between graphs. As a result, the proposed HTAK kernel  $k_{\text{HTAK}}^{(H)}$  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.

**Second**, as Fröhlich et al. (Fröhlich et al., 2005) have stated, the transitive alignment step is necessary to guarantee the positive definiteness of alignment kernels. Thus, the proposed HTAK kernel  $k_{\text{HTAK}}^{(H)}$  guarantees the positive definiteness that is not available to the aforementioned alignment kernels (Fröhlich et al., 2005; Bai et al., 2015a;b; Neuhaus & Bunke, 2006).

**Third**, the computation of the proposed HTAK kernel  $k_{\text{HTAK}}^{(H)}$  for a pair of graphs incorporates the information over all graphs under comparisons. This is because  $k_{\text{HTAK}}^{(H)}$  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.,  $k_{\text{HTAK}}^{(H)}$  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 (Borgwardt & Kriegel, 2005; Bach, 2008; Gärtner et al., 2003; Aziz et al., 2013; Shervashidze et al., 2010; Costa & Grave, 2010). As a result, the proposed kernel  $k_{\text{HTAK}}^{(H)}$  may reflect richer graph characteristics.

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

### 3.4. Computational Analysis

For the set of  $T$  graphs  $\mathbf{G}$  each of which has  $n$  vertices and  $m$  edges, computing the proposed kernel  $k_{\text{HTAK}}^{(H)}$  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 D-B 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  $O(HIN_1Tn)$ . 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  $O(HIN_1Tn + HT^2N_1 + HTN_1n + Tn \log n + Tmn)$ . Note that, in this work, we employ the fastest  $K$ -means MATLAB implementation developed by Deng Cai (Cai, 2012), 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. Thus, the whole time complexity is approximately  $O(N_1Tn + T^2N_1 + TN_1n + Tn^2)$ , indicating that our kernel can be computed in a polynomial time.

## 4. 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

Table 1. Information of the graph based computer vision (CV), bioinformatics (Bio), and social network (SN) datasets.

Datasets	BAR31	BSPHERE31	GEOD31	MUTAG	NCI1	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 2. Classification Accuracy (In %  $\pm$  Standard Error) for Comparisons with Graph Kernels.

Datasets	BAR31	BSPHERE31	GEOD31	MUTAG	NCI1	CATH2	COLLAB	IMDB-B	IMDB-M
<b>HTAK</b>	71.00 $\pm$ .45	<b>62.90<math>\pm</math>.65</b>	<b>47.80<math>\pm</math>.49</b>	87.32 $\pm$ .60	79.01 $\pm$ .14	<b>87.89<math>\pm</math>.71</b>	<b>79.87<math>\pm</math>.15</b>	<b>72.89<math>\pm</math>.56</b>	50.23 $\pm$ .18
ASK	<b>73.10<math>\pm</math>.67</b>	60.30 $\pm$ .44	46.21 $\pm$ .69	87.50 $\pm$ .65	78.47 $\pm$ .12	78.52 $\pm$ .67	77.53 $\pm$ .31	70.38 $\pm$ .72	50.12 $\pm$ .51
WLSK	58.53 $\pm$ .53	42.10 $\pm$ .68	38.20 $\pm$ .68	82.88 $\pm$ .57	84.77 $\pm$ .13	67.36 $\pm$ .63	77.39 $\pm$ .35	71.88 $\pm$ .77	49.50 $\pm$ .49
SPGK	55.73 $\pm$ .44	48.20 $\pm$ .76	38.40 $\pm$ .65	83.38 $\pm$ .81	74.21 $\pm$ .30	81.89 $\pm$ .63	58.80 $\pm$ .20	71.26 $\pm$ 1.04	<b>51.33<math>\pm</math>.57</b>
CORE SP	—	—	—	<b>88.29<math>\pm</math>1.55</b>	73.46 $\pm$ .32	—	—	72.62 $\pm$ .59	49.43 $\pm$ .42
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	<b>85.32<math>\pm</math>.14</b>	68.70 $\pm$ .69	76.85 $\pm$ .40	72.45 $\pm$ .81	50.33 $\pm$ .49
PMGK	—	—	—	80.66 $\pm$ .90	72.27 $\pm$ .59	—	—	68.53 $\pm$ .61	45.75 $\pm$ .66
CORE PM	—	—	—	87.19 $\pm$ 1.47	74.90 $\pm$ .45	—	—	71.04 $\pm$ .64	48.30 $\pm$ 1.01
RetGK(MC)	—	—	—	—	—	—	73.60 $\pm$ .30	71.00 $\pm$ .60	46.70 $\pm$ .60

The symbol - means that some approaches were not evaluated by the original authors.

3D Shape database, that consists of 15 classes and 20 individuals per class (Biasotti et al., 2003). 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.1.

#### 4.1. 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) (Bai et al., 2015a), 2) the Weisfeiler-Lehman subtree kernel (WLSK) (Shervashidze et al., 2011), 3) the shortest path graph kernel (SPGK) (Borgwardt & Kriegel, 2005), 4) the SPGK kernel associated with Core Variants (CORE SP) (Nikolentzos et al., 2018), 5) the graphlet count graph kernel (Shervashidze et al., 2009) with graphlet of size 4 (GCGK), 6) the Jensen-Tsallis q-difference kernel (JTQK) (Bai et al., 2014) with  $q = 2, 7$ ) the Pyramid Match Graph Kernel (PMGK) (Nikolentzos et al., 2017), 8) the PMGK kernel associated with Core Variants (CORE PM) (Nikolentzos et al., 2018), and the 9) the graph kernel based on the Monte Carlo computation of return probability features (RetGK<sub>MC</sub>) (Zhang et al., 2018b). On the other hand, **the deep learning methods include** 1) the deep graph convolutional neural network (DGCNN) (Zhang et al., 2018a), 2) the PATCHY-SAN based convolutional neural network for graphs (PSGCNN) (Niepert et al., 2016), 3) the diffusion convolutional neural network (DCNN) (Atwood & Towsley,

2016), and 4) the deep graphlet kernel (DGK) (Yanardag & Vishwanathan, 2015).

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 (Bai et al., 2014; Shervashidze et al., 2011). For the ASK kernel, we set the highest layer of the required DB representation as 50 based on (Bai et al., 2015a), 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 (Chang & Lin, 2011). 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 2. 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. In other words, we initialize the number of prototype representations as the number of 20% vertices over all graphs, capturing the main characteristics from all graphs.

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

Table 3. Classification Accuracy (In %  $\pm$  Standard Error) for Comparisons with Deep Learning Methods.

Datasets	MUTAG	NCII	COLLAB	IMDB-B	IMDB-M
<b>HTAK</b>	87.32 $\pm$ .60	<b>79.01</b> $\pm$ .14	<b>79.87</b> $\pm$ .15	<b>72.89</b> $\pm$ .56	<b>50.23</b> $\pm$ .18
DGCNN	85.83 $\pm$ 1.66	74.44 $\pm$ .47	73.76 $\pm$ .49	70.03 $\pm$ .86	47.83 $\pm$ .85
PSGCNN	<b>88.95</b> $\pm$ 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 $\pm$ .71	49.06 $\pm$ 1.37	33.49 $\pm$ 1.42
DGK	82.66 $\pm$ 1.45	62.48 $\pm$ .25	73.09 $\pm$ .25	66.96 $\pm$ .56	44.55 $\pm$ .52

are shown in Table.3. Finally, note that, as we have stated in Section 3.2, 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 (Gammerman et al., 1998) (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. In fact, on the other hand, one can also construct the prototype representations based on just the training graphs, and we observe that this will not significantly influence the final performance. This is because the training process will employ 90 percent of the available sample graphs as the training set, reflecting the main characteristics over all graphs. In other words, for dynamic graph classification problems, one does not need to re-construct the prototype representations and re-train the C-SVMs associated with the proposed kernel, avoiding the time-consuming problem. Finally, note that, some methods are not evaluated by the original authors on some datasets, thus we do not exhibit these results.

**Experimental Analysis:** 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 NCII dataset are higher than the proposed HTAK kernel. On the 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

indicate 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 sub-structures, 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.

## 5. Conclusions

In this paper, we develop a new Hierarchical Transitive-Aligned Kernel for un-attributed graphs. Unlike most existing kernels, this kernel not only overcomes the shortcoming of ignoring correspondence information between graphs, but also guarantees the transitivity between the correspondence information. Experiments demonstrate that the proposed kernel can outperform state-of-the-art graph kernels and deep learning methods in terms of graph classifications.

Our future work is to extend the proposed kernel for vertex and edge attributed graphs. Moreover, we will also employ the kernel on more real-world graph structure data and investigate the performance, e.g., financial networks (Bai et al., 2020a), brain networks (Wang et al., 2020), etc.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants T2122020, 61976235, and 61602535. Corresponding Author: Lixin Cui.



## References

- Atwood, J. and Towsley, D. Diffusion-convolutional neural networks. In *Proceedings of NIPS*, pp. 1993–2001, 2016.
- Azaïs, R. and Ingels, F. The weight function in the subtree kernel is decisive. *J. Mach. Learn. Res.*, 21:67:1–67:36, 2020.
- Aziz, F., Wilson, R. C., and Hancock, E. R. Backtrackless walks on a graph. *IEEE Transactions on Neural Networks and Learning Systems*, 24(6):977–989, 2013.
- Bach, F. R. Graph kernels between point clouds. In *Proceedings of ICML*, pp. 25–32, 2008.
- Bai, L., Rossi, L., Bunke, H., and Hancock, E. R. Attributed graph kernels using the jensen-tsallis q-differences. In *Proceedings of ECML-PKDD*, pp. I:99–114, 2014.
- Bai, L., Rossi, L., Zhang, Z., and Hancock, E. R. An aligned subtree kernel for weighted graphs. In *Proceedings of ICML*, pp. 30–39, 2015a.
- Bai, L., Zhang, Z., Wang, C., Bai, X., and Hancock, E. R. A graph kernel based on the jensen-shannon representation alignment. In *Proceedings of IJCAI*, pp. 3322–3328, 2015b.
- Bai, L., Cui, L., Zhang, Z., Xu, L., Wang, Y., and Hancock, E. R. Entropic dynamic time warping kernels for co-evolving financial time series analysis. *IEEE Transactions on Neural Networks Learn. Syst.*, In Press, 2020a. doi: 10.1109/TNNLS.2020.3006738.
- Bai, L., Rossi, L., Cui, L., Cheng, J., Wang, Y., and Hancock, E. R. A quantum-inspired similarity measure for the analysis of complete weighted graphs. *IEEE Transactions on Cybernetics*, 50(3):1264 – 1277, 2020b.
- Bai, X., Hancock, E. R., and Wilson, R. C. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589–2606, 2009.
- Biasotti, S., Marini, S., Mortara, M., Patanè, G., Spagnuolo, M., and Falcidieno, B. 3d shape matching through topological structures. In *Proceedings of DGCI*, pp. 194–203, 2003.
- Borgwardt, K. M. and Kriegel, H. Shortest-path kernels on graphs. In *Proceedings of ICDM*, pp. 74–81, 2005.
- Borgwardt, K. M., Ghisu, M. E., Llinares-López, F., O’Bray, L., and Rieck, B. Graph kernels: State-of-the-art and future challenges. *Found. Trends Mach. Learn.*, 13(5-6), 2020.
- Cai, D. The fastest matlab implementation of the  $k$ -means. In *Software available at <http://www.zjucadcg.cn/dengcai/Data/Clustering.html>*, 2012.
- Chang, C.-C. and Lin, C.-J. Libsvm: A library for support vector machines. *Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>*, 2011.
- Costa, F. and Grave, K. D. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of ICML*, pp. 255–262, 2010.
- Fröhlich, H., Wegner, J. K., Sieker, F., and Zell, A. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of ICML*, pp. 225–232, 2005.
- Gaidon, A., Harchaoui, Z., and Schmid, C. A time series kernel for action recognition. In *Proceedings of BMVC*, pp. 1–11, 2011.
- Gammerman, A., Vovk, V., and Vapnik, V. Learning by transduction. In *Proceedings of UAI*, pp. 148–155, 1998.
- Gärtner, T., Flach, P., and Wrobel, S. On graph kernels: hardness results and efficient alternatives. In *Proceedings of COLT*, pp. 129–143, 2003.
- Gkirtzou, K. and Blaschko, M. B. The pyramid quantized weisfeiler-lehman graph representation. *Neurocomputing*, 173:1495–1507, 2016.
- Harchaoui, Z. and Bach, F. Image classification with segmentation graph kernels. In *Proceedings of CVPR*, 2007.
- Haussler, D. Convolution kernels on discrete structures. In *Technical Report UCS-CRL-99-10*, Santa Cruz, CA, USA, 1999.
- Kashima, H., Tsuda, K., and Inokuchi, A. Marginalized kernels between labeled graphs. In *Proceedings of ICML*, pp. 321–328, 2003.
- Kriege, N. and Mutzel, P. Subgraph matching kernels for attributed graphs. In *Proceedings of ICML*, 2012.
- Long, Q., Jin, Y., Wu, Y., and Song, G. Theoretically improving graph neural networks via anonymous walk graph kernels. In *Proceedings of WWW*, pp. 1204–1214, 2021.
- Neuhaus, M. and Bunke, H. Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006.
- Niepert, M., Ahmed, M., and Kutzkov, K. Learning convolutional neural networks for graphs. In *Proceedings of ICML*, pp. 2014–2023, 2016.

- Nikolentzos, G., Meladianos, P., and Vazirgiannis, M. Matching node embeddings for graph similarity. In *Proceedings of AAAI*, pp. 2429–2435, 2017.
- Nikolentzos, G., Meladianos, P., Limnios, S., and Vazirgiannis, M. A degeneracy framework for graph similarity. In *Proceedings of IJCAI*, pp. 2595–2601, 2018.
- Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., and Borgwardt, K. Efficient graphlet kernels for large graph comparison. *Journal of Machine Learning Research*, 5:488–495, 2009.
- Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 1: 1–48, 2010.
- Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.
- Sugiyama, M. and Borgwardt, K. M. Halting in random walk kernels. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Proceedings of NIPS*, pp. 1639–1647, 2015.
- Togninalli, M., Ghisu, M. E., Llinares-López, F., Rieck, B., and Borgwardt, K. M. Wasserstein weisfeiler-lehman graph kernels. In *Proceedings of NeurIPS*, pp. 6436–6446, 2019.
- Wang, J., Zhang, L., Wang, Q., Chen, L., Shi, J., Chen, X., Li, Z., and Shen, D. Multi-class ASD classification based on functional connectivity and functional correlation tensor via multi-source domain adaptation and multi-view sparse representation. *IEEE Trans. Medical Imaging*, 39 (10):3137–3147, 2020.
- Wilson, R. C., Hancock, E. R., and Luo, B. Pattern vectors from algebraic graph theory. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1112–1124, 2005.
- Witten, I. H., Frank, E., and Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.
- Xu, B., Ting, K. M., and Jiang, Y. Isolation graph kernel. In *Proceedings of AAAI*, pp. 10487–10495, 2021a.
- Xu, L., Bai, L., Jiang, X., Tan, M., Zhang, D., and Luo, B. Deep rényi entropy graph kernel. *Pattern Recognit.*, 111: 107668, 2021b.
- Yanardag, P. and Vishwanathan, S. V. N. Deep graph kernels. In *Proceedings of KDD*, pp. 1365–1374, 2015.
- Zhang, M., Cui, Z., Neumann, M., and Chen, Y. An end-to-end deep learning architecture for graph classification. In *Proceedings of AAAI*, 2018a.
- Zhang, Z., Wang, M., Xiang, Y., Huang, Y., and Nehorai, A. Retgk: Graph kernels based on return probabilities of random walks. In *Proceedings of NeurIPS*, pp. 3968–3978, 2018b.