

# Hypergraph Learning for Unsupervised Graph Alignment via Optimal Transport

Yuguang Yan<sup>1</sup>, Canlin Yang<sup>1</sup>, Yuanlin Chen<sup>1</sup>, Ruichu Cai<sup>1,2\*</sup>, Michael Ng<sup>3</sup>

<sup>1</sup>School of Computer Science, Guangdong University of Technology, Guangzhou, China

<sup>2</sup>Peng Cheng Laboratory, Shenzhen, China

<sup>3</sup>Department of Mathematics, Hong Kong Baptist University, Hong Kong, China

ygyan@gdut.edu.cn, {yangcl0608,chenyuanlin27,cairuichu}@gmail.com, michael-ng@hkbu.edu.hk

## Abstract

Unsupervised graph alignment aims to find corresponding nodes across different graphs without supervision. Existing methods usually leverage the graph structure to aggregate features of nodes to find relations between nodes. However, the graph structure is inherently limited in pairwise relations between nodes without considering higher-order dependencies among multiple nodes. In this paper, we take advantage of the hypergraph structure to characterize higher-order structural information among nodes for better graph alignment. Specifically, we propose an optimal transport model to learn a hypergraph to capture complex relations among nodes, so that the nodes involved in one hyperedge can be adaptively based on local geometric information. In addition, inspired by the Dirichlet energy function of a hypergraph, we further refine our model to enhance the consistency between structural and feature information in each hyperedge. After that, we jointly leverage graphs and hypergraphs to extract structural and feature information to better model the relations between nodes, which is used to find node correspondences across graphs. We conduct experiments on several benchmark datasets with different settings, and the results demonstrate the effectiveness of our proposed method.

## Introduction

Graph alignment aims to recognize node correspondences across different graphs (Wang et al. 2018; Heimann et al. 2018). As structured data become ubiquitous, graph alignment has received much attention and been widely used in real-world applications, such as finding the same users on different social network platforms (Li et al. 2019), recognizing the same entities in different knowledge graphs (Liu et al. 2022). In the problem of graph alignment with supervision, some node correspondences are given as ground-truth information (Liu et al. 2016). However, it is difficult to obtain supervised information since annotation is usually expensive and time-consuming. Therefore, unsupervised graph alignment without requiring supervision has drawn more attention in recent years (Chen et al. 2019).

Existing methods of unsupervised graph alignment usually leverage features of nodes and the graph structure to

find the node correspondences across two graphs (Chen et al. 2019; Heimann et al. 2018; Li et al. 2019). The first paradigm is to learn a shared node embedding space for two graphs, so that the embeddings of nodes in different graphs can be compared, and the node correspondences can be found by nearest neighbor matching (Derr et al. 2021). However, it is non-trivial to learn a shared embedding space especially when the feature spaces of two graphs are diverse, making it challenging to compare the node embeddings in different graphs. To avoid directly comparing node embeddings across graphs, another paradigm based on optimal transport between metric spaces is proposed for unsupervised graph alignment (Li et al. 2022). Specifically, the Gromov-Wasserstein model is built on two graphs to find the optimal transport plan across graphs, where the transport plan reveals the probabilistic correspondences between nodes in different graphs (Peyré, Cuturi, and Solomon 2016). The transport cost across graphs is measured by the difference between two metric matrices, each of which is constructed within one graph.

Although existing methods achieve encouraging performance for unsupervised graph alignment, they still suffer from the limitation of graph structure that only pairwise information between nodes is considered, while higher-order relations among multiple nodes are neglected (Ju et al. 2024). For example, a community consisting of multiple nodes contains complex dependencies among them, which cannot be captured by pairwise edges between two nodes. Compared with graphs in which one edge can connect only two nodes, the hypergraph has shown a powerful ability to characterize high-order relations among nodes, owing to the property that one hyperedge can involve multiple nodes (Gao et al. 2020). Figure 1 depicts the differences between the hypergraph and the graph representing a co-authored citation network. In the figure, a node represents a paper, and the edges in the graph can only describe the co-author relationship between the pairs of nodes, while the hyperedges in the hypergraph can contain all the papers that an author participated in. This property also endows the hypergraph with the ability to capture higher-order information.

Motivated by this, we seek to leverage hypergraphs to extract high-order structural information for unsupervised graph alignment, and design a method named **Hypergraph Learning via Optimal Transport**(HLOT). Specifically, we

\*Corresponding author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

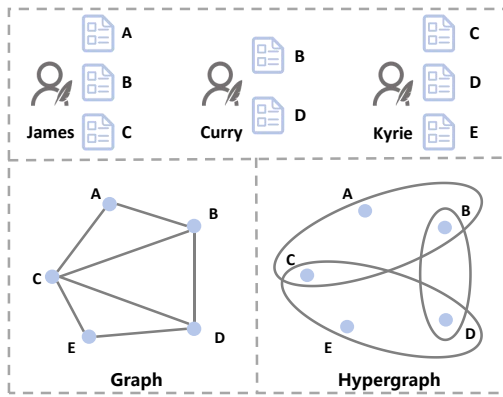


Figure 1: An example of using graph and hypergraph to represent a co-authorship citation network.

propose an optimal transport model within one graph to construct a weighted hypergraph, which can adaptively assign weights to the pairs of nodes and hyperedges, so that the different correlations can be captured. Moreover, inspired by the Dirichlet energy function of hypergraph that measures the consistency between structural and feature information, we refine our optimal transport model to derive a hypergraph learning model. Based on our constructed hypergraphs, we exploit structural and feature information on both graph and hypergraph levels, so that a better metric modeling the relations between nodes can be built for graph alignment. We conduct experiments on real-world datasets with different settings to evaluate the performance of our method, and empirically study the effects of graphs and hypergraphs.

We summarize our principal contributions as follows:

- To characterize high-order structural information among nodes, we propose an optimal transport model to learn hypergraphs, in which the consistency between structural and feature information is enhanced by the Dirichlet energy function of the hypergraph.
- To better exploit structural and feature information, we construct intra-graph metrics from the perspective of both graphs and hypergraphs.
- To evaluate the performance of our method and the effects of graphs and hypergraphs, we conduct experiments and ablation studies on real-world data sets with different settings.

## Related Works

### Unsupervised Graph Alignment

Graph alignment is usually treated as a supervised learning task, relying on the known node correspondence to enhance prediction performance (Zhang et al. 2015, 2021; Fey et al. 2020; Yan, Zhang, and Tong 2021). However, obtaining these correspondences is often expensive and could be impractical, leading to the development of unsupervised graph alignment models. (Zhang and Tong 2016) is a pioneer study that proposed the consistency principle based on

the similarity of nodes, which guides the evolution of unsupervised graph alignment algorithms. Existing methods of unsupervised graph alignment can be divided into two main-stream paradigms.

The first paradigm, known as “embed-then-cross-compare alignment”, first embeds the representation of nodes, and then determines the graph alignment based on similarity measurement. For example, REGAL (Heimann et al. 2018) uses matrix decomposition to generate node representations and match similar node pairs. WAlign (Gao, Huang, and Li 2021) employs a graph convolutional network (Kipf and Welling 2017) architecture to learn node embeddings, updating them incrementally using a Wasserstein distance discriminator. GCNAlign (Wang et al. 2018) integrates structural and attribute information through graph convolution networks, updating embeddings based on distance loss and identifying alignments via cross-graph similarity. GAlign (Trung et al. 2020) enhances graph structures by introducing perturbations and dynamically adjusting weights between high-confidence node pairs.

The second paradigm introduces the concept of Gromov-Wasserstein distance (Mémoli 2011), transforming the graph alignment problem into an optimal transport problem between metric spaces. GWL (Xu et al. 2019a) applies optimal transport for graph alignment, using the graph adjacency matrix to represent the cost matrix. FusedGW (Titouan et al. 2019a) extends the Gromov-Wasserstein distance to incorporate both structural and feature information of the graphs. However, since aligned graphs are typically constructed from different sources, there usually exists an inconsistency in structure and features between the two graphs.

To address this, SLOAlign (Tang et al. 2023) integrated multi-view structures containing rich information and introduced a learnable weight coefficient to balance the contributions of optimal transport from different views. Our work further incorporates the concept of hypergraphs into the unsupervised graph alignment framework based on optimal transport. By leveraging the hypergraph to model higher-order node relations, our model is better equipped to capture intricate complex structures and information, thereby improving both the alignment accuracy and robustness of the alignment process.

### Hypergraph

Hypergraph an extension of the traditional graph allows edges to connect an arbitrary number of nodes, thus capturing complex interactions and higher-order information among nodes. Hypergraphs have been effectively applied in various domains (Zhou, Huang, and Schölkopf 2006), including citation networks (Jiang et al. 2019), social networks (Yang et al. 2019), medical data analysis (Di et al. 2021), and recommendation systems (Wang et al. 2020). To leverage the hypergraph structure to learn node embeddings, HGNN (Feng et al. 2019) generalizes convolutional operators from Graph Convolutional Networks to hypergraph structures, significantly enhancing the representational power of hypergraph models. Building on this, subsequent works have introduced novel hypergraph convolutional operators and aggregation methods, further advancing the representation

learning of hypergraphs (Yan et al. 2024; Wu, Yan, and Ng 2022; Dong, Sawin, and Bengio 2020).

In this work, we combine the optimal transport framework with the hypergraph energy function to construct the hypergraph with a finer granularity, and then discover the higher-order relationship between the source graph and the target graph, and further enhance the effectiveness and robustness of the model.

## Optimal Transport

Optimal transport (OT) (Villani 2008) is a mathematical framework designed to find an optimal transport plan that minimizes the cost of transferring a distribution from a source to a target, given their marginal distributions and a predefined cost function. Optimal transport has drawn significant attention for its applicability in various domains such as computer vision (Solomon et al. 2015), domain adaptation (Courty et al. 2016), node classification (Titouan et al. 2019b) and graph classification (Wang et al. 2024).

The Gromov-Wasserstein distance, originally designed to compare metric spaces by aligning their underlying structures, captures both the structural and distributional differences between two distributions (Mémoli 2011; Peyré, Cuturi, and Solomon 2016). The Gromov-Wasserstein distance can identify the optimal mapping between nodes that preserves the structural consistency of the graphs, making it a natural fit for unsupervised graph alignment tasks (Tang et al. 2023). This insight has led to the reformulation of unsupervised graph alignment as an OT problem, thereby introducing the above optimal transport graph alignment paradigm without requiring any ground-truth node correspondence.

## Preliminary

**Graph** An undirected graph can be represented as  $G = (V, \mathbf{A}, \mathbf{X})$ , where  $V = \{v_1, v_2, \dots, v_{|V|}\}$  denotes the set of nodes in graph  $G$ ,  $\mathbf{X} \in \mathbb{R}^{|V| \times d}$  is the node attribute matrix of the graph  $G$ , with  $d$  as the dimension of the attribute. The structure of the graph  $G$  is represented by the adjacency matrix  $\mathbf{A} \in \{0, 1\}^{|V| \times |V|}$ , where  $A_{ij} = 1$  if there is an edge connecting  $v_i$  and  $v_j$ , otherwise  $A_{ij} = 0$ . Let  $\mathbf{D} \in \mathbb{R}^{|V| \times |V|}$  be the degree matrix of the graph  $G$ , where each diagonal value  $D_{ii} = \sum_j A_{ij}$  represents the number of neighbors of  $v_i$ .

**Hypergraph** A hypergraph is the generalization of the graph, where each hyperedge can be viewed as a set of nodes. Formally, a hypergraph can be denoted as  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{W}_e)$ , where  $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$  is the set of nodes.  $\mathcal{E} = \{e_1, e_2, \dots, e_{|\mathcal{E}|}\}$  is the set of hyperedges, with each element  $e$  being a subset of  $\mathcal{V}$ . The matrix  $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$  represents the feature matrix of the nodes, where the feature of node  $v_i$  is represented  $\mathbf{x}_i \in \mathbb{R}^d$  which is the transpose of the  $i$ -th row of  $\mathbf{X}$ . The diagonal matrix  $\mathbf{W}_e \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$  represents the weight of the hyperedge, where the  $j$ -th diagonal element  $w_j$  indicating the importance of the hyperedge  $e_j$ . In general, an incidence matrix  $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$  can be used to represent the structure of the hypergraph  $\mathcal{G}$ , where each

element  $H_{ij}$  indicates the connection between node  $v_i$  and hyperedge  $e_j$ . Specifically,  $H_{ij}$  equals to 1 if the node  $v_i$  is contained in the hyperedge  $e_j$ , otherwise 0. Additionally, the diagonal matrices  $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$  and  $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$  are used to denote the degree of nodes and hyperedges, respectively, where the diagonal values of  $\mathbf{D}_v$  and  $\mathbf{D}_e$  are given by  $\sum_j w_j H_{ij}$  and  $\sum_i H_{ij}$ , respectively.

## Problem Statement

In this study, we consider an unsupervised strategy to solve the alignment problem of two attribute graphs. Specifically, Given two attribute undirected graphs, source graph  $G_s = (V_s, A_s, X_s)$  and target graph  $G_t = (V_t, A_t, X_t)$ , where  $v_i^s, v_j^t$  represent the  $i$ -th node of the source graph  $G_s$ , and the  $j$ -th node in target graph  $G_t$ , respectively, and  $n_s, n_t$  are the numbers of nodes in  $G_s$  and  $G_t$ , respectively. There is a naturally occurring set of alignment node pairs  $\mathcal{M} = \{(v_i^s, v_j^t) | (v_i^s, v_j^t) \in V_s \times V_t\}$ , where  $v_i^s$  and  $v_j^t$  represent the same node in two different graphs. The goal of the unsupervised alignment task is to find a mapping function  $\pi(v_i^s, v_j^t)$  that accurately identifies the correspondence between nodes without using any observable ground-truth node correspondence.

## Methodology

Figure 2 illustrates the overview of our proposed method HLOT. We first learn hypergraphs for source and target graphs, respectively. After that, we leverage both hypergraphs and graphs to extract structural information and feature information, and then construct intra-graph metrics to characterize the relations between nodes. Finally, we find node correspondences between two graphs based on a probabilistic coupling matrix. In the following, we detailedly describe our proposed method.

## Hypergraph Level Learning

**Hypergraph Generation** To capture high-order relations among nodes, we construct hypergraphs for  $G_s$  and  $G_t$  by generating hyperedges that connect multiple nodes. Traditional methods usually find the nearest neighbors for a node to generate a hyperedge (Feng et al. 2019). However, it is heuristic to determine the number of nodes in a hyperedge. Different from them, we propose to employ a doubly stochastic matrix  $\mathbf{H} \in \mathcal{H}$  to represent the hypergraph structure, where the domain  $\mathcal{H}$  is defined as  $\mathcal{H} = \{\mathbf{H} \in \mathbb{R}_+^{|\mathcal{V}| \times |\mathcal{E}|} | \mathbf{H}\mathbf{1} = \frac{1}{|\mathcal{V}|}\mathbf{1}, \mathbf{H}^\top \mathbf{1} = \frac{1}{|\mathcal{E}|}\mathbf{1}\}$ . The constraints mean that the sums of all the rows (*resp.*, columns) are equal, indicating that all the nodes (*resp.*, hyperedge) have the same weight. Based on this, we propose the following model to adaptively find nearest neighbors and construct hyperedges

$$\min_{\mathbf{H} \in \mathcal{H}} \langle \mathbf{C}, \mathbf{H} \rangle + \epsilon \Omega(\mathbf{H}), \quad (1)$$

where  $C_{ij}$  is the squared Euclidean distance between two nodes, *i.e.*,

$$C_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2, \quad (2)$$

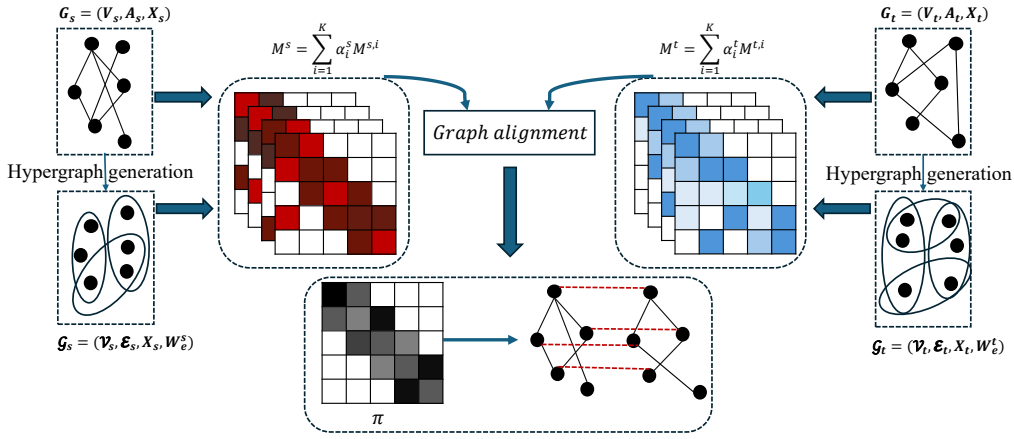


Figure 2: The overview of our proposed method HLOT.

and  $\Omega(\mathbf{H})$  is the negative entropy of  $\mathbf{H}$  defined as

$$\Omega(\mathbf{H}) = \sum_{ij} H_{ij} (\log H_{ij} - 1), \quad (3)$$

which is used to smoothen the hypergraph structure  $\mathbf{H}$  so that one hyperedge can involve multiple nodes. Since each hyperedge is built based on one node,  $C_{ii} = 0$  will induce the trivial solution that the diagonal elements  $H_{ii}$  dominate the others. To avoid this, we set  $C_{ii}$  as a sufficiently large value  $\rho$  to construct a matrix  $\tilde{\mathbf{C}} = \mathbf{C} + \rho \mathbf{I}$  where  $\mathbf{I}$  is the identity matrix, so that sufficiently small  $H_{ii}$  will be induced.

Moreover, we refine the hypergraph structure from the perspective of the total variation (or the Dirichlet energy function) of the hypergraph, which can be used to analyze the properties of a hypergraph or used as a regularization term for hypergraph learning (Hein et al. 2013). By extending the total variation of a hypergraph with 0/1 incidence matrix  $\mathbf{H}$  in (Hein et al. 2013), we define the total variation of a hypergraph with a weighted incidence matrix  $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$  as follows,

$$\text{TV}(\mathbf{H}) = \sum_{e_k \in \mathcal{E}} w_k \max_{i,j \in e_k} H_{ik} H_{jk} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2, \quad (4)$$

which is based on the node pair in one hyperedge with the largest distance. For two nodes with a strong correlation in one hyperedge (*i.e.*, a large  $H_{ik} H_{jk}$ ), they are expected to have similar features and a small distance. Therefore, a small  $\text{TV}(\mathbf{H})$  comes from the consistency between structural information in  $\mathbf{H}$  and distance information on features  $\mathbf{X}$ .

Motivated by this, we seek to learn a hypergraph structural  $\mathbf{H}$  by minimizing the total variation of it. However, the maximum operation in Eq. (4) is non-differentiable and difficult to address in optimization. To tackle this, we instead design the following loss function to approximate Eq. (4)

$$\Psi(\mathbf{H}) = \sum_k w_k \sum_{ij} H_{ik} H_{jk} C_{ij} = \text{tr}(\mathbf{W}_e \mathbf{H}^\top \mathbf{C} \mathbf{H}), \quad (5)$$

where the weighted distances between nodes in one hyperedge are considered. By minimizing this, the consistency between structural information and feature information is enhanced.

Finally, we achieve the following optimization problem to learn the hypergraph structure

$$\min_{\mathbf{H} \in \mathcal{H}} \langle \tilde{\mathbf{C}}, \mathbf{H} \rangle + \theta \Psi(\mathbf{H}) + \epsilon \Omega(\mathbf{H}), \quad (6)$$

where  $\theta$  and  $\epsilon$  are the trade-off parameters.

We employ the projected gradient descent method to solve this problem (Peyré, Cuturi, and Solomon 2016). The technical details are given in the appendix. After obtaining the weighted hypergraph  $\mathbf{H}$ , we construct two views to extract structural and feature information, respectively.

**Hypergraph Structure View** First, we leverage the hypergraph structure to model the relations between nodes. To this end, we accumulate the incidence degrees of all the hyperedges that are shared by two nodes to measure the relations between these two nodes. Formally, the relation between nodes  $v_i$  and  $v_j$  is calculated as  $M_{ij}^{hs} = \sum_k H_{ik} H_{jk}$ , and the intra-graph relation matrix between nodes is calculated as  $\mathbf{M}^{hs} = \mathbf{H} \mathbf{H}^\top$ , where the superscript *hs* means hypergraph structure view. Different from a classical 0/1 adjacency matrix, the matrix  $\mathbf{M}^{hs}$  contains continuous values, which reflect the strengths of the relations between two nodes.

**Hypergraph Feature View** Second, we leverage the hypergraph structure to aggregate the features of nodes and their neighbors. To achieve this, we apply the hypergraph convolutional operation (Feng et al. 2019) to obtain the embeddings of nodes, and the embeddings of the  $(l+1)$ -th layer is updated as

$$\mathbf{Z}_h^{(l+1)} = \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W}_e \mathbf{D}_e^{-1} \mathbf{H}^\top \mathbf{D}_v^{-\frac{1}{2}} \mathbf{Z}_h^{(l)}, \quad (7)$$

where  $\mathbf{Z}_h^0 = \mathbf{X}$  is the input feature for the hypergraph,  $\mathbf{D}_v$  is the node degree matrix,  $\mathbf{D}_e$  is the edge degree matrix, and  $\mathbf{W}_e$  is a diagonal matrix with diagonal elements indicating the weights of the hyperedges. In practice, to filter weak incidence in  $\mathbf{H}$  and speed up the calculation, for each node, we adopt  $\kappa$  hyperedges with the highest incidence values, and set the values of them as one while the others as zero. As a result, we obtain a sparse incidence matrix  $\mathbf{H}$  with strong relations between nodes and hyperedge preserved.

---

**Algorithm 1: HLOT**


---

**Input:** Source graph  $G_s = (U_s, A_s, X_s)$ , target graph  $G_t = (V_t, A_t, X_t)$ ,  
**Initialize:**  $\alpha_i^s = \alpha_i^t = \frac{1}{K}$ ,  $\alpha^1 = [\alpha^s, \alpha^t]$ ,  $\pi_{ij}^1 = \frac{1}{n_s n_t}$ .  
1: Construct hypergraph  $\mathbf{H}$  through solving Problem (6);  
2: Construct candidate structure bases  $\mathbf{M}^s$  and  $\mathbf{M}^t$ ;  
3: **repeat**  
4: Update  $\alpha^{k+1}$  by solving Problem (15);  
5: Update  $\pi^{k+1}$  by solving Problem (16);  
6: **until** convergence  
7: Generate node pairs according to  $\pi$ .

---

After obtaining the final embeddings  $\mathbf{Z}_h$ , we construct the intra-graph relations between nodes based on the similarity metric, which is implemented by the inner product here, *i.e.*,  $\mathbf{M}^{hf} = \mathbf{Z}_h \mathbf{Z}_h^\top$ , where the superscript *hf* means **hypergraph feature view**.

### Graph Level Learning

In this part, we construct intra-graph metric matrices based on the graph structure. Following the above approach for hypergraphs, two views are constructed to extract structural and feature information, respectively.

**Graph Structure View** The graph structural information is well captured by the adjacency matrix  $\mathbf{A}$ . Therefore, we construct the graph structure view as  $\mathbf{M}^{gs} = \mathbf{A}$ , where the superscript *gs* means **graph structure view**.

**Graph Feature View** To leverage feature information of nodes by the graph structure, we apply a parameter-free graph convolutional operation (Wu et al. 2019) to aggregate features of nodes and their neighbors. Specifically, given the node embeddings of the (*l*)-th layer, the embeddings are updated as

$$\mathbf{Z}_g^{(l+1)} = \mathbf{D}^{-\frac{1}{2}} \tilde{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}} \mathbf{Z}_g^{(l)}, \quad (8)$$

where  $\mathbf{Z}_g^{(0)}$  is the input features for the graph,  $\mathbf{D}$  is the degree matrix of the graph,  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  is the adjacency matrix with self loops, where  $\mathbf{I}$  is an identity matrix. After obtaining the embeddings  $\mathbf{Z}_g$  from GCN, we use a similarity metric to calculate node similarity:  $\mathbf{M}^{gf} = \mathbf{Z}_g \mathbf{Z}_g^\top$ , where the superscript *gf* means **graph feature view**.

### Graph Alignment

Given structural and feature views obtained from both hypergraphs and graphs, we are ready to combine them to construct the optimal representations of two graphs and find node correspondences between them. For simplicity, for the source graph  $G_s$  (*resp.*, the target graph  $G_t$ ), we build matrices  $\mathbf{M}^{s,1}$ ,  $\mathbf{M}^{s,2}$ ,  $\mathbf{M}^{s,3}$ ,  $\mathbf{M}^{s,4}$  (*resp.*,  $\mathbf{M}^{t,1}$ ,  $\mathbf{M}^{t,2}$ ,  $\mathbf{M}^{t,3}$ ,  $\mathbf{M}^{t,4}$ , ) based on  $\mathbf{M}^{hs}$ ,  $\mathbf{M}^{hf}$ ,  $\mathbf{M}^{gs}$ ,  $\mathbf{M}^{gf}$ , respectively. We construct intra-graph metric matrices by the convex combinations of the above views, *i.e.*,

$$\mathbf{M}^s = \sum_i \alpha_i^s \mathbf{M}^{s,i}, \quad \mathbf{M}^t = \sum_i \alpha_i^t \mathbf{M}^{t,i}, \quad (9)$$

where the weight vectors  $\alpha^s$  and  $\alpha^t$  are in the following domain

$$\Sigma_K = \{\mathbf{a} \in \mathbb{R}^K \mid \mathbf{a}^\top \mathbf{1} = 1, a_i \in [0, 1] \forall i\}. \quad (10)$$

Following (Chen et al. 2020; Xu et al. 2019b) we use a probabilistic coupling matrix  $\pi$  to indicate the node correspondences between these two graphs, and apply the Gromov-Wasserstein model to learn the optimal coupling  $\pi$ . Here,  $\pi_{ij}$  represents the matching degree between  $v_i^s$  and  $v_j^t$ , and  $\pi$  is in the following domain

$$\Pi = \{\pi \in \mathbb{R}_+^{n_s \times n_t} \mid \pi \mathbf{1} = \frac{1}{n_s} \mathbf{1}, \pi^\top \mathbf{1} = \frac{1}{n_t} \mathbf{1}\}, \quad (11)$$

and the Gromov-Wasserstein model aims to minimize the following objective function

$$F(\pi, \alpha^s, \alpha^t) = \sum_{i,i',j,j'} L_{i,i',j,j'} \cdot \pi_{ij} \cdot \pi_{i'j'}, \quad (12)$$

where the loss tensor is defined as

$$L_{i,i',j,j'} = \frac{1}{2} (M_{ii'}^s - M_{jj'}^t)^2, \quad (13)$$

which measures the transport cost between node pairs  $(v_i^s, v_{i'}^s)$  and  $(v_j^t, v_{j'}^t)$ .

Combining the learning of the weight vectors  $\alpha^s$  and  $\alpha^t$ , we solve the following optimization problem to find the optimal weight vectors and node correspondences jointly

$$\begin{aligned} \min_{\pi, \alpha^s, \alpha^t} F(\pi, \alpha^s, \alpha^t) \\ \text{s.t. } \pi \in \Pi, \alpha^s \in \Sigma_K, \alpha^t \in \Sigma_K. \end{aligned} \quad (14)$$

We alternately update two blocks of the variables, *i.e.*,  $\pi$  and  $\alpha = [\alpha^s; \alpha^t]$ . At the  $\tau$ -th iteration, for the weight variables  $\alpha$ , we follow (Tang et al. 2023) to apply a projected proximal gradient method to find  $\alpha \in \Sigma_{2K}$  that minimizes the linearized problem, *i.e.*,

$$\alpha^{\tau+1} = \arg \min_{\alpha \in \Sigma_{2K}} \langle \nabla_\alpha F(\pi^\tau, \alpha^\tau), \alpha \rangle + \lambda \|\alpha - \alpha^\tau\|_2^2, \quad (15)$$

where  $\nabla_\alpha F(\pi^\tau, \alpha^\tau)$  is the gradient of  $F(\pi^\tau, \alpha^\tau)$  with respect to  $\alpha$ .

For the probabilistic coupling matrix  $\pi$ , to handle the doubly stochastic constraint defined in  $\Pi$ , we apply a projected

Data	Type	#Node	#Edges	#Attr.
Cora	Citation Network	2708	5028	1433
Citeseer	Citation Network	3327	4732	3703
PPI	Protein Interaction	1767	16159	171
Facebook	Social Network	4039	44117	1476
ACM	Co-Author Network	9872	39561	17
DBLP	Co-Author Network	9916	44808	17
Douban -Online	Social Network	3906	16328	538
Douban -Offline	Social Network	1118	3022	538

Table 1: Statistical information of the datasets.

Model	Douban Online-Offline				ACM-DBLP			
	Hit@1	Hit@5	Hit@10	Hit@30	Hit@1	Hit@5	Hit@10	Hit@30
KNN	3.31	10.38	16.64	30.05	49.25	59.46	63.42	69.61
REGAL	30.32	54.83	-	-	34.09	46.58	51.35	56.34
GCNAlign	20.93	34.44	39.62	50.72	38.43	68.46	77.64	86.89
GATAlign	23.70	36.94	44.01	57.16	14.21	34.07	42.12	49.00
WAlign	35.69	57.87	69.05	83.09	50.61	72.87	80.84	89.47
GWD	3.04	7.96	9.21	11.90	56.24	77.14	82.20	84.92
FusedGW	29.61	62.79	66.46	68.07	30.80	38.39	39.26	39.60
SLOTAlign	51.43	73.43	77.73	82.02	<b>66.04</b>	84.06	87.95	90.32
HLOT	<b>57.33</b>	<b>79.70</b>	<b>84.17</b>	<b>87.12</b>	65.94	<b>84.70</b>	<b>88.76</b>	<b>91.32</b>

Table 2: Hit@k results on two real-world graph alignment datasets.

proximal gradient method with the Kullback-Leibler (KL) divergence, *i.e.*,

$$\pi^{\tau+1} = \arg \min_{\pi \in \Pi} \langle \nabla_{\pi} F(\pi^{\tau}, \alpha^{\tau+1}), \pi \rangle + \gamma \text{KL}(\pi | \pi^{\tau}), \quad (16)$$

where the KL divergence is defined as

$$\text{KL}(\pi | \pi^{\tau}) \stackrel{\text{def.}}{=} \sum_{ij} \pi_{ij} \log \left( \frac{\pi_{ij}}{\pi_{ij}^{\tau}} \right) - \pi_{ij} + \pi_{ij}^{\tau}. \quad (17)$$

As a result, the Sinkhorn algorithm (Peyré, Cuturi, and Solomon 2016) can be employed to efficiently update  $\pi$ . Algorithm 1 summarizes our proposed method. The complexity analysis is given in the appendix.

## Experiments

### Datasets

We list the statistical information of the datasets in Table 1, and describe the datasets in the following.

- **Douban** (Tang et al. 2023) is a social network dataset and contains online and offline graphs. Nodes and edges have the same meaning as the Facebook dataset. The location information of a user is used as node features for the Douban dataset.
- **ACM-DBLP** (Tang et al. 2023) is a co-author network dataset and contains two graphs extracted from the publication information in four research areas. It consists of a collection of papers and their authors. The node features of ACM-DBLP are bag-of-words vector representations.
- **Citeseer** (Sen et al. 2008) is a co-citation dataset. It consists of the collection of papers and their citation links. The node features of Citeseer are bag-of-words vector representations.
- **PPI** (Zitnik and Leskovec 2017) is a Protein-Protein interaction network dataset. The nodes represent the proteins in the network and the edges represent the Protein-Protein interactions.
- **Cora** (Sen et al. 2008) is a citation network dataset. The node features of Cora are based on the term frequency-inverse document frequency (TF-IDF).

- **Facebook** (Leskovec and McAuley 2012) is a social network dataset. The nodes represent users in a social network and edges represent the correspondence between users. Facebook employs user profile information as the node features.

Among the above six datasets, the first two are real-world graph alignment datasets, and the other four are semi-synthetic datasets used to evaluate the graph alignment methods. The detailed settings are described in the following parts.

### Compared Methods

**KNN**: This method directly matches nodes with their closest  $K$  nodes as candidates in feature space. **REGAL** (Heimann et al. 2018): It is a graph alignment method based on embedding that can be used for graphs with or without node features. **WAlign** (Gao, Huang, and Li 2021): This method uses a lightweight Graph Convolutional Network with a Wasserstein distance discriminator to obtain a matching matrix between the source and target graphs. **GCNAlign** (Wang et al. 2018): This method uses a graph convolutional network to calculate node embeddings for both the source and target graphs. **GATAlign** (Velickovic et al. 2017): This method is similar to GCNAlign, while a Graph Attention Network is used to obtain node embeddings for source and target graphs. **GWD** (Xu et al. 2019a): This method uses the Gromov-Wasserstein distance for graph alignment, which only pays attention to structural information by using the graph adjacency matrices. **FuesdGW** (Titouan et al. 2019b): FuesdGW contains both the Wasserstein distance and the Gromov-Wasserstein distance. It takes into account both structural and feature information for graph alignment. **SLOTAlign** (Tang et al. 2023): SLOTAlign proposes a multi-view structure to learn graph representations and reduce the effect of structural and feature inconsistency.

### Experiment Settings

For our proposed method, we employ a one-layer parameter-free graph convolutional network and one one-layer parameter-free HGNN network for all datasets except ACM-DBLP. Since ACM-DBLP has more nodes and edges than

Model	Permutation				Compression			
	Cora	Citeseer	PPI	Facebook	Cora	Citeseer	PPI	Facebook
KNN	89.55(0.027)	57.44(0.019)	6(0.003)	20.85(0.042)	0(0.004)	0.03(0.004)	0.06(0.004)	0.02(0.002)
GWD	50.37(0.001)	14.52(0.002)	84.95(0.003)	70.17(0.002)	50.37(0.02)	14.52(0.014)	84.95(0.32)	70.17(0.25)
FusedGW	97.45(0.002)	99.58(0.002)	78.53(0.002)	64.17(0.002)	0.85(0.004)	0.57(0.001)	0.85(0.003)	0.87(0.002)
SLOTAlign	99.25(0.001)	99.25(0.002)	89.64(0.001)	98.71(0.003)	98.45(0.027)	<b>99.76(0.02)</b>	86.47(0.3)	98.61(0.043)
HLOT	<b>99.56(0.004)</b>	<b>99.73(0.003)</b>	<b>89.98(0.005)</b>	<b>99.46(0.004)</b>	<b>98.56(0.003)</b>	99.73(0.007)	<b>87.44(0)</b>	<b>98.71(0.002)</b>

Table 3: Hit@1 results of four disturbance datasets.

Model	Douban Online-Offline			
	Hit@1	Hit@5	Hit@10	Hit@30
HLOT-w/o $M_{h,s}$	45.35	63.42	67.62	72.18
HLOT-w/o $M_{h,f}$	52.95	77.91	81.84	85.87
HLOT-w/o $M_{g,s}$	1.07	5.1	8.77	19.32
HLOT-w/o $M_{g,f}$	47.14	64.4	68.25	73.08
HLOT	<b>57.33</b>	<b>79.7</b>	<b>84.17</b>	<b>87.12</b>

Table 4: Hit@k results of ablation studies.

other datasets, we use two-layer parameter-free GCN and two-layer parameter-free HGNN to capture information.

Following (Tang et al. 2023), for each experiment, we tune the hyperparameters  $\theta$  in the range  $\{0.001, 0.01, 0.1, 0.2, 0.5, 1\}$ ,  $\epsilon$  in the range  $\{0.001, 0.01, 0.1, 1, 10, 100\}$ , and select the highest Hit@k value that performs best as a result. For the updating of  $\alpha^s, \alpha^t$ , we train the learnable weight for 500 epochs by gradient descent for all the datasets except that the epochs of ACM-DBLP are set to 1000.

We run all the methods on the Pytorch platform and the experiments are conducted on a Linux server with an NVIDIA RTX 4090 (24GB) graphics card. The running time results are reported in the appendix.

**Evaluation Metrics** We adopt Hit@k to evaluate the performance of the graph alignment methods. This metric calculates the percentage of nodes in  $\mathcal{V}_t$  whose ground-truth alignment results in  $\mathcal{V}_s$  are among the top-k candidates.

## Results and Discussions

For fair comparisons, we strictly follow the experimental setting in (Tang et al. 2023), in which we directly align two graphs in noisy real-world graph alignment datasets to test the performance of graph alignment of our method. We align the source graph with the target graph. Table 2 presents the results of graph alignment in terms of the Hit@k, which shows that our method outperforms the other methods. Compared with the state-of-the-art method SLOTAlign, HLOT has 5.1% improvement in Hit@30 on the Douban data, and 1.0% improvement in Hit@30 on the ACM-DBLP data. It verifies that high-order information involved in hypergraphs is helpful for graph alignment.

We also use two kinds of disturbing ways to disturb the source graph to obtain the target graph, which are used for graph alignment. The first type is a permutation, meaning we randomly select  $p\%$  feature columns and then rearrange the feature columns at random. The second type is compression. We use some dimensionality reduction methods such as PCA to reduce dimension to  $(100 - p)\%$ . We set the  $p = 50$  in the following experiments. Table 3 presents the results, where the results of the mean and standard derivation of 10 times are reported. Our method also outperforms other methods and maintains a high performance of all disturbance settings for all datasets. It means that our method can steadily perform hypergraph learning and graph learning in the case of feature disturbance. We also conduct experiments by randomly removing 10% nodes from the source graph and report the results in the appendix.

## Ablation Studies

To investigate the impacts of structural and feature information of hypergraphs and graphs, we conduct ablation studies on Douban and report the Hit@k results of different variants of our method in Table 4, in which each variant removes a view. We observe that the results of Douban drop significantly after the graph structure view is removed, which is consistent with the observation in (Tang et al. 2023) for SLOTAlign, indicating that the structural information  $M_{g,s}$  is important to the datasets. Compared with the variants without the hypergraph level information, *i.e.*, without  $M_{h,s}$  or  $M_{h,f}$ , HLOT achieves better performance, which verifies that hypergraphs are beneficial for extracting high-order information for better graph alignment. We also test a version that replaces the hypergraph view with a similarity metric, and show our results in the appendix.

## Conclusion

In this paper, we propose to learn hypergraphs to capture high-order information for unsupervised graph alignment. We apply an optimal transport model to learn hypergraphs, which are refined by the total variation function of the hypergraph for structure and feature consistency. Both hypergraphs and graphs are applied to model the relations between nodes to find the node correspondences across graphs. We conduct experiments with different settings to demonstrate the effectiveness of our proposed method.

## Acknowledgments

This work was supported in part by the National Science and Technology Major Project (2021ZD0111501), National Natural Science Foundation of China (62206061, U24A20233), National Science Fund for Excellent Young Scholars (62122022), Guangdong Basic and Applied Basic Research Foundation (2024A1515011901), and Guangzhou Basic and Applied Basic Research Foundation (2023A04J1700). The work of Michael Ng was supported in part by the Hong Kong Research Grant Council GRF (17201020, 17300021), C7004-21GF, and Joint NSFC-RGC (N-HKU76921).

## References

- Chen, C.; Xie, W.; Huang, W.; Rong, Y.; Ding, X.; Huang, Y.; Xu, T.; and Huang, J. 2019. Progressive Feature Alignment for Unsupervised Domain Adaptation. arXiv:1811.08585.
- Chen, L.; Gan, Z.; Cheng, Y.; Li, L.; Carin, L.; and Liu, J. 2020. Graph Optimal Transport for Cross-Domain Alignment. arXiv:2006.14744.
- Courty, N.; Flamary, R.; Tuia, D.; and Rakotomamonjy, A. 2016. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9): 1853–1865.
- Derr, T.; Karimi, H.; Liu, X.; Xu, J.; and Tang, J. 2021. Deep adversarial network alignment. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 352–361.
- Di, D.; Shi, F.; Yan, F.; Xia, L.; Mo, Z.; Ding, Z.; Shan, F.; Song, B.; Li, S.; Wei, Y.; et al. 2021. Hypergraph learning for identification of COVID-19 with CT imaging. *Medical image analysis*, 68: 101910.
- Dong, Y.; Sawin, W.; and Bengio, Y. 2020. Hnhn: Hypergraph networks with hyperedge neurons. arXiv preprint arXiv:2006.12278.
- Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 3558–3565.
- Fey, M.; Lenssen, J. E.; Morris, C.; Masci, J.; and Kriege, N. M. 2020. Deep graph matching consensus. arXiv preprint arXiv:2001.09621.
- Gao, J.; Huang, X.; and Li, J. 2021. Unsupervised graph alignment with wasserstein distance discriminator. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 426–435.
- Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S.; and Zou, C. 2020. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5): 2548–2566.
- Heimann, M.; Shen, H.; Safavi, T.; and Koutra, D. 2018. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 117–126.
- Hein, M.; Setzer, S.; Jost, L.; and Rangapuram, S. S. 2013. The total variation on hypergraphs-learning on hypergraphs revisited. *Advances in Neural Information Processing Systems*, 26.
- Jiang, J.; Wei, Y.; Feng, Y.; Cao, J.; and Gao, Y. 2019. Dynamic hypergraph neural networks. In *IJCAI*, 2635–2641.
- Ju, W.; Mao, Z.; Yi, S.; Qin, Y.; Gu, Y.; Xiao, Z.; Wang, Y.; Luo, X.; and Zhang, M. 2024. Hypergraph-enhanced Dual Semi-supervised Graph Classification. arXiv:2405.04773.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.
- Leskovec, J.; and McAuley, J. 2012. Learning to discover social circles in ego networks. *Advances in neural information processing systems*, 25.
- Li, C.; Wang, S.; Wang, H.; Liang, Y.; Yu, P. S.; Li, Z.; and Wang, W. 2019. Partially shared adversarial learning for semi-supervised multi-platform user identity linkage. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 249–258.
- Li, J.; Tang, J.; Kong, L.; Liu, H.; Li, J.; So, A. M.-C.; and Blanchet, J. 2022. Fast and Provably Convergent Algorithms for Gromov-Wasserstein in Graph Data. arXiv preprint arXiv:2205.08115.
- Liu, L.; Cheung, W. K.; Li, X.; and Liao, L. 2016. Aligning Users across Social Networks Using Network Embedding. In *Ijcai*, volume 16, 1774–80.
- Liu, X.; Hong, H.; Wang, X.; Chen, Z.; Kharlamov, E.; Dong, Y.; and Tang, J. 2022. Selfkg: Self-supervised entity alignment in knowledge graphs. In *Proceedings of the ACM Web Conference 2022*, 860–870.
- Mémoli, F. 2011. Gromov-Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4): 417–487.
- Peyré, G.; Cuturi, M.; and Solomon, J. 2016. Gromov-Wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, 2664–2672.
- Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.
- Solomon, J.; De Goes, F.; Peyré, G.; Cuturi, M.; Butscher, A.; Nguyen, A.; Du, T.; and Guibas, L. 2015. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4): 1–11.
- Tang, J.; Zhang, W.; Li, J.; Zhao, K.; Tsung, F.; and Li, J. 2023. Robust Attributed Graph Alignment via Joint Structure Learning and Optimal Transport. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 1638–1651.
- Titouan, V.; Courty, N.; Tavenard, R.; and Flamary, R. 2019a. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, 6275–6284. PMLR.
- Titouan, V.; Courty, N.; Tavenard, R.; Laetitia, C.; and Flamary, R. 2019b. Optimal Transport for structured data with application on graphs. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6275–6284. PMLR.

- Trung, H. T.; Van Vinh, T.; Tam, N. T.; Yin, H.; Weidlich, M.; and Hung, N. Q. V. 2020. Adaptive network alignment with unsupervised and multi-order convolutional networks. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 85–96. IEEE.
- Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph Attention Networks. *ArXiv*, abs/1710.10903.
- Villani, C. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.
- Wang, J.; Ding, K.; Hong, L.; Liu, H.; and Caverlee, J. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 1101–1110.
- Wang, Y.; Zhao, Y.; Wang, D. Z.; and Li, L. 2024. GALOPA: graph transport learning with optimal plan alignment. *Advances in Neural Information Processing Systems*, 36.
- Wang, Z.; Lv, Q.; Lan, X.; and Zhang, Y. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 349–357.
- Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.
- Wu, H.; Yan, Y.; and Ng, M. K.-P. 2022. Hypergraph collaborative network on vertices and hyperedges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3245–3258.
- Xu, H.; Luo, D.; Zha, H.; and Duke, L. C. 2019a. Gromov-Wasserstein Learning for Graph Matching and Node Embedding. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6932–6941. PMLR.
- Xu, H.; Luo, D.; Zha, H.; and Duke, L. C. 2019b. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, 6932–6941. PMLR.
- Yan, Y.; Chen, Y.; Wang, S.; Wu, H.; and Cai, R. 2024. Hypergraph Joint Representation Learning for Hypervertices and Hyperedges via Cross Expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9232–9240.
- Yan, Y.; Zhang, S.; and Tong, H. 2021. Bright: A bridging algorithm for network alignment. In *Proceedings of the web conference 2021*, 3907–3917.
- Yang, D.; Qu, B.; Yang, J.; and Cudre-Mauroux, P. 2019. Revisiting user mobility and social relationships in lbsns: a hypergraph embedding approach. In *The world wide web conference*, 2147–2157.
- Zhang, S.; and Tong, H. 2016. Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1345–1354.
- Zhang, S.; Tong, H.; Jin, L.; Xia, Y.; and Guo, Y. 2021. Balancing consistency and disparity in network alignment. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2212–2222.
- Zhang, Y.; Tang, J.; Yang, Z.; Pei, J.; and Yu, P. S. 2015. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1485–1494.
- Zhou, D.; Huang, J.; and Schölkopf, B. 2006. Learning with Hypergraphs: Clustering, Classification, and Embedding. In Schölkopf, B.; Platt, J.; and Hoffman, T., eds., *Advances in Neural Information Processing Systems*, volume 19. MIT Press.
- Zitnik, M.; and Leskovec, J. 2017. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14): i190–i198.