

From One to All: Learning to Match Heterogeneous and Partially Overlapped Graphs

Wei jie Liu,^{1,2,3} Hui Qian,^{2,3,6} Chao Zhang,^{2,3} Jiahao Xie,^{2,3} Zebang Shen,⁴ Nenggan Zheng^{1,2,3,5*}

¹ Qiushi Academy for Advanced Studies, Zhejiang University

² College of Computer Science and Technology, Zhejiang University

³ Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies

⁴ University of Pennsylvania

⁵ Zhejiang Lab

⁶ State Key Lab of CAD&CG, Zhejiang University

{westonhunter,qianhui,zczju,xiejh}@zju.edu.cn, zebang@seas.upenn.edu, zng@cs.zju.edu.cn

Abstract

Recent years have witnessed a flurry of research activity in graph matching, which aims at finding the correspondence of nodes across two graphs and lies at the heart of many artificial intelligence applications. However, matching heterogeneous graphs with partial overlap remains a challenging problem in real-world applications. This paper proposes the first practical learning-to-match method to meet this challenge. The proposed unsupervised method adopts a novel partial OT paradigm to learn a transport plan and node embeddings simultaneously. In a *from-one-to-all* manner, the entire learning procedure is decomposed into a series of easy-to-solve sub-procedures, each of which only handles the alignment of a single type of nodes. A mechanism for searching the *transport mass* is also proposed. Experimental results demonstrate that the proposed method outperforms state-of-the-art graph matching methods.

1 Introduction

Graph matching (network alignment), aiming to determine the correspondence of nodes across two related graphs, lies at the heart of a wide range of artificial intelligence applications, including network retrieval (Berretti, Del Bimbo, and Vicario 2001; Özer, Wolf, and Akansu 2002), machine translation (Bahdanau, Cho, and Bengio 2014; Chen et al. 2020), and visual tracking (Xiong et al. 2012; Wang and Ling 2017), to name a few. Two main algorithmic components, the *node conservation* and the *matching strategy*, constitute the graph matching method, where the former measures the similarity between pairs of nodes from different networks, and the later maximizes total node conservation over aligned nodes and the amount of conserved edges (Gu et al. 2018).

Mainstream methods of graph matching have followed two directions that correspond to different matching strategies. (i) One is the *seed-and-extend* strategy, which recursively matches nodes adjacent to the currently aligned subgraphs (Narayanan and Shmatikov 2009; Pedarsani and

Grossglauser 2011; Yartseva and Grossglauser 2013; Sun et al. 2015). In general, incorrectly matched pairs could cause more mismatching in future steps, i.e., a cascade of errors (Kazemi 2016). (ii) The other is the *searching* strategy, which explores the entire alignment space and returns the one with the highest score based on a specific objective. In this paradigm, a few recent researches have shifted focus to settings in which the best searching is rephrased in the framework of *optimal transport* (OT) and the resulting algorithms have achieved the state-of-the-art performance (Xu et al. 2019; Titouan et al. 2019; Barbe et al. 2020), compared with its traditional counterparts such as *simulated annealing* (Mamano and Hayes 2017), *genetic algorithm* (Vijayan, Saraph, and Milenković 2015), or *gradient-based optimization* (Konar and Sidiropoulos 2020).

Despite this recent activity, the advances in methodology have been confined almost exclusively to the matching of *homogeneous* (containing only one type of nodes) or fully overlapped graphs, and none of existing methods has demonstrated its effectiveness on real-world alignment tasks in which graphs are both *heterogeneous* and *partially overlapped*. Such real-world tasks may include knowledge graph alignment (Li et al. 2018), matching of biological networks (Sharan and Ideker 2006), and disambiguating entities (Zhang, Swami, and Chawla 2019). We illustrate this situation in Figure 1 and depict technical details of these two challenges in the following paragraphs.

Heterogeneity. Most existing methods consider graphs with only one type of nodes. Directly applying these methods to heterogeneous graphs leads to inferior performance and type mismatch. Gu et al. (2018) propose the first practical heterogeneous graph matching method based on manually designed features, called colored graphlet degree vectors (CGDV), and extend a large body of homogeneous algorithms (Sun et al. 2015; Vijayan, Saraph, and Milenković 2015; Mamano and Hayes 2017) into heterogeneous variants. The designed features, however, are expensive to obtain since the computational complexity for generating CGDV is practically $\mathcal{O}(V^5)$, where V is the number of nodes of the graph. Apparently, as V increases, their performances will deteriorate rapidly. Besides, these methods cannot guarantee

*Corresponding author.

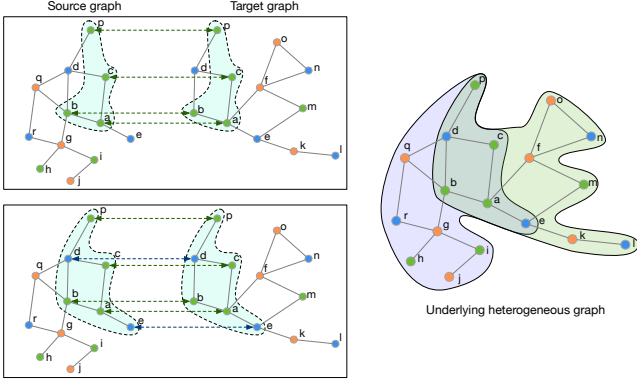


Figure 1: An illustration of the from-one-to-all matching strategy for both heterogeneous and partially overlapped graphs. The right sub-figure depicts an underlying heterogeneous graph. The color of each node indicates its type. The source graph and the target graph pictured in the left sub-figure are its two partially overlapped sub-graphs. For example, node r in the source graph does not exist in the target graph. First, type-green nodes are matched (top row of the left sub-figure). Then, type-blue nodes are matched (bottom row of the left sub-figure).

that there is no type mispairing. Thus, efficiently and effectively matching all types of nodes still remains a challenging problem.

Partial Overlapping. Partial overlapping is another challenging problem, which is generally tackled by adding *dummy nodes* that act like wildcards to absorb unmatched nodes (Swoboda et al. 2017, 2019; Sarlin et al. 2020; Rolínek et al. 2020). However, this kind of methods still suffer from two intractable issues. First, they are usually supervised methods and require a large amount of ground truth node pairs. Second, they cannot constrain the number of matching pairs even if the prior information about the degree of overlap is already known, which may lead to some extreme cases. For example, zero matching would be identified to be an optimal result when dummy nodes are adopted.

To meet these challenges from real-world tasks, we propose a novel method to match heterogeneous graphs that are possibly partially overlapped. The proposed method adopts a novel partial OT scheme to learn a transport plan and node embeddings simultaneously. For the sake of computational efficiency, the entire learning procedure is decomposed into a series of easy-to-solve sub-procedures. Specifically, each sub-procedure only handles a homogeneous but possibly partially overlapped alignment problem, with a set of node pairs which are already matched in the previous sub-procedures as seeds. Essentially, this method proceeds by first matching one type of nodes and gradually matching other types until all types are matched, that is, from one to all types, so we name it FOTA. To boost robustness against noise and further reduce the cascade of errors, we also incorporate node embeddings which encode the global topology. Our contributions are summarized as follows.

- i. FOTA is the first practical unsupervised method to match both heterogeneous and partially overlapped graphs, so far as we know, by using a hybrid strategy that mixes seed-and-extend and searching strategies. Compared to the principled method in (Gu et al. 2018), ours achieves no type mismatch at a very low cost. The overall complexities for updating the transport plan and the embeddings are $\mathcal{O}(N(T + k + d)V^2)$ and $\mathcal{O}(VBd)$ respectively¹.
- ii. A mechanism for searching the transport mass is also proposed, which endows our method a possibility to control the number of matching pairs according to some prior information. Thus the partial OT technique leveraged by our method only needs to transport an enough fraction of the mass with a minimum transportation cost.

Extensive experimental results demonstrate that the proposed method outperforms the state-of-the-art graph matching methods. The rest of the paper is organized as follows. In Sec. 2, a comprehensive review of background is given. The methodology of FOTA is presented in Sec. 3. Empirical results are demonstrated in both Sec. 4. We finally present related work in Sec. 5.

Notation. We use bold lowercase symbols, bold uppercase letters, uppercase calligraphic fonts, and Greek letters to denote vectors, matrices, spaces (sets), and measures, respectively. $\mathbf{1}^d \in \mathbb{R}^d$ is an all-ones vector. The cardinality of set \mathcal{A} is denoted by $|\mathcal{A}|$. $\mathbf{A}[i, :]$ and $\mathbf{A}[:, j]$ are the i -th row and the j -th column of matrix \mathbf{A} respectively.

2 Preliminaries

2.1 Graph Matching

A graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi, \mathcal{T})$, where \mathcal{V} is the set of nodes, \mathcal{E} is the set of edges, \mathcal{T} is the set of node types, and the type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{T}$ assigns each node a type. When graph \mathcal{G} contains multiple types of nodes, i.e., $|\mathcal{T}| \geq 2$, \mathcal{G} is a heterogeneous graph; otherwise, it is a homogeneous graph. Assigning each node an index $i \in \{1, \dots, |\mathcal{V}|\}$, the edge set \mathcal{E} can also be written as an adjacency matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ with entry $W_{ij} = 1$ if and only if there is an edge connecting nodes i and j .

Mathematically, graph matching finds a matching matrix $\mathbf{T} = [T_{ii'}]$ between the source graph $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s, \phi^s, \mathcal{T})$ and the target graph $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t, \phi^t, \mathcal{T})$, where $T_{ii'} = 1$ if node $i \in \mathcal{V}^s$ is matched to node $i' \in \mathcal{V}^t$, and $T_{ii'} = 0$ otherwise. Assuming \mathcal{G}^s has fewer nodes than \mathcal{G}^t , the matching matrix is generally identified by minimizing the following loss function (Caetano et al. 2009)

$$\min_{\mathbf{T} \in \mathcal{P}^f} \left[\sum_{i,j=1}^{|\mathcal{V}^s|} \sum_{i',j'=1}^{|\mathcal{V}^t|} d_{ii'jj'} T_{ii'} T_{jj'} + \sum_{i=1}^{|\mathcal{V}^s|} \sum_{i'=1}^{|\mathcal{V}^t|} k_{ii'} T_{ii'} \right], \quad (1)$$

¹Here N , T , k , d and B denote the number of main iterations, the number of iterative projections, the rank of approximation to the proximity matrix, embedding dimension, and the batch size respectively. See Sec. 3.4 for detail.

where $k_{ii'}$ is the cost for the *unary matching* $i \rightarrow i'$, $d_{ii'jj'}$ is the cost for the *pairwise matching* $(i, j) \rightarrow (i', j')$, and the feasible domain is

$$\mathcal{P}^f = \left\{ \mathbf{T} \in \{0, 1\}^{|\mathcal{V}^s| \times |\mathcal{V}^t|} \mid \mathbf{T}\mathbf{1}^{|\mathcal{V}^t|} = \mathbf{1}^{|\mathcal{V}^s|}, \mathbf{T}^\top \mathbf{1}^{|\mathcal{V}^s|} \leq \mathbf{1}^{|\mathcal{V}^t|} \right\}.$$

Such formulation matches every node in \mathcal{G}^s to exact one node in \mathcal{G}^t and hence is referred to as *full matching* (Kazemi, Yartseva, and Grossglauser 2015). For *partial matching*, generally \mathcal{P}^f is replaced with the relaxed feasible domain

$$\mathcal{P}^p = \left\{ \mathbf{T} \in \{0, 1\}^{|\mathcal{V}^s| \times |\mathcal{V}^t|} \mid \mathbf{T}\mathbf{1}^{|\mathcal{V}^t|} \leq \mathbf{1}^{|\mathcal{V}^s|}, \mathbf{T}^\top \mathbf{1}^{|\mathcal{V}^s|} \leq \mathbf{1}^{|\mathcal{V}^t|} \right\},$$

see e.g. (Kazemi, Yartseva, and Grossglauser 2015; Sarlin et al. 2020; Wang et al. 2020). Such formulation, however, may lead to matching no pairs in extreme cases. For example, when $d_{ii'jj'} > 0$ and $k_{ii'} > 0$ for all i, i', j, j' , matching no nodes ($\mathbf{T} = \mathbf{0}$) is the optimal solution. Matching two heterogeneous graphs further requires $\phi^s(i) = \phi^t(i')$ if node i is matched to i' .

2.2 Optimal Transport

Optimal Transport (OT) addresses the problem of transporting one measure toward another measure with the minimum cost (Villani 2008). The induced cost defines a distance between the two measures. A discrete measure α can be denoted by $\alpha = \sum_{i=1}^m p_i \delta_{\mathbf{x}_i}$ where $\delta_{\mathbf{x}}$ is the Dirac at position \mathbf{x} , i.e., a unit of mass infinitely concentrated at \mathbf{x} . With slight abuse of notation, we also use $\mathbf{p} = [p_i]$ to refer to α .

Wasserstein distance. The p -Wasserstein distance (Villani 2008; Cuturi 2013) between discrete measures \mathbf{p} and \mathbf{q} is defined as

$$W_p^p(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{T} \in \Pi(\mathbf{p}, \mathbf{q})} \sum_{i=1}^m \sum_{i'=1}^n K_{ii'}^p T_{ii'},$$

where $K_{ii'}$ is the ℓ_p distance between \mathbf{x}_i and $\mathbf{y}_{i'}$ and the feasible domain of *transport plan* $\mathbf{T} = [T_{ii'}]$ is given by the set $\Pi(\mathbf{p}, \mathbf{q}) = \{\mathbf{T} \in \mathbb{R}_+^{m \times n} \mid \mathbf{T}\mathbf{1}^n = \mathbf{p}, \mathbf{T}^\top \mathbf{1}^m = \mathbf{q}\}$. The Wasserstein distance requires the supports of the two measures to be in the same space.

Gromov-Wasserstein distance. Gromov-Wasserstein (GW) distance extends Wasserstein distance to compare measures supported in different spaces (Mémoli 2011). Let \mathcal{X} and \mathcal{Y} be two sample spaces. Endowing the spaces \mathcal{X} and \mathcal{Y} with metrics (distances) $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$, the GW distance is defined as

$$GW_p^p(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{T} \in \Pi(\mathbf{p}, \mathbf{q})} \sum_{i,j=1}^m \sum_{i',j'=1}^n D_{ii'jj'}^p T_{ii'} T_{jj'},$$

where $D_{ii'jj'} = |d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathcal{Y}}(\mathbf{y}_{i'}, \mathbf{y}_{j'})|$ with $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m \in \mathcal{X}$ and $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n \in \mathcal{Y}$.

OT-based graph matching. By associating each graph with a discrete probability measure, OT can be applied to graph matching. Xu et al. (2019) propose a GW learning framework called GWL for graph matching. They correspond two graphs \mathcal{G}^s and \mathcal{G}^t to discrete probability measures $\mu^s = [\mu_i^s]$ and $\mu^t = [\mu_i^t]$ respectively, where

$$\mu_i^z = \frac{\sum_{j=1}^{|\mathcal{V}^z|} W_{ij}^z}{\sum_{i=1}^{|\mathcal{V}^z|} \sum_{j=1}^{|\mathcal{V}^z|} W_{ij}^z},$$

for $z = s, t$. By replacing the strict distances with dissimilarity functions, GWL relaxes the GW distance and the Wasserstein distance to the GW discrepancy and the Wasserstein discrepancy separately. Such relaxation allows GWL to incorporate node embeddings to parameterize the discrepancies and improve the robustness to the noise of edges. GWL uses the learned transport plan to indicate the node correspondence, i.e., $i' \in \mathcal{V}^t$ that receives the most mass from $i \in \mathcal{V}^s$ is the estimated counterpart of i .

Both the Wasserstein distance and the GW distance require the two marginals \mathbf{p} and \mathbf{q} to have the same total mass, that is, $\|\mathbf{p}\|_1 = \|\mathbf{q}\|_1$, thus all the mass has to be transported. By contrast, the partial OT problem focuses on transporting only a fraction $0 \leq b \leq \min\{\|\mathbf{p}\|_1, \|\mathbf{q}\|_1\}$ of the mass with the minimum transportation cost (Figalli 2010; Caffarelli and McCann 2010; Chapel, Gasso et al. 2020), that is, the set of admissible couplings is given by

$$\Pi^b(\mathbf{p}, \mathbf{q}) = \left\{ \mathbf{T} \in \mathbb{R}_+^{m \times n} \mid \mathbf{T}\mathbf{1} \leq \mathbf{p}, \mathbf{T}^\top \mathbf{1} \leq \mathbf{q}, \mathbf{1}^\top \mathbf{T}\mathbf{1} = b \right\}.$$

By adding a dummy node into the target graph, the mass of the nodes which have no counterparts in the target graph can be considered as being transported to the dummy node.

3 Methodology

We first introduce the underlying model in real-world scenarios for matching graphs that are heterogeneous and partially overlapped. Next, we derive a practical *from-one-to-all* model by decomposing the learning procedure into a series of interrelated sub-procedures. A recursive line search mechanism is then proposed to search for the transport mass in order to conduct partial matching. Finally, we analyze the overall complexity of the proposed method.

3.1 Proposed Model

Our model is a learning-to-match model which estimates the transport plan and the node embeddings simultaneously. The optimization formulation can be phrased as follows:

$$\begin{aligned} \min_{\mathbf{Z}^s, \mathbf{Z}^t} \min_{\mathbf{T} \in \Omega(\mathbf{b}, \mathcal{G}^s, \mathcal{G}^t)} & \underbrace{\sum_{i,j=1}^{|\mathcal{V}^s|} \sum_{i',j'=1}^{|\mathcal{V}^t|} (C_{ij}^s - C_{i'j'}^t)^2 T_{ii'} T_{jj'}}_{\text{partial GW discrepancy}} \\ & + \underbrace{\alpha \sum_{i=1}^{|\mathcal{V}^s|} \sum_{i'=1}^{|\mathcal{V}^t|} -\kappa(\mathbf{z}_i^s, \mathbf{z}_{i'}^t) T_{ii'}}_{\text{partial Wasserstein discrepancy}} + \underbrace{\beta (R(\mathbf{Z}^s) + R(\mathbf{Z}^t))}_{\text{regularizers}}, \end{aligned} \quad (2)$$

where $\mathbf{Z}^s = [\mathbf{z}_i^s]$ and $\mathbf{Z}^t = [\mathbf{z}_{i'}^t]$ are node embeddings of \mathcal{G}^s and \mathcal{G}^t respectively, $\mathbf{b} = [b_r]$ specifies the transport mass for each type. The feasible domain $\Omega(\mathbf{b}, \mathcal{G}^s, \mathcal{G}^t)$ is defined as

$$\begin{aligned} \Omega(\mathbf{b}, \mathcal{G}^s, \mathcal{G}^t) = \left\{ \mathbf{T} \in \mathbb{R}_+^{|\mathcal{V}^s| \times |\mathcal{V}^t|} \mid \right. & \mathbf{T}\mathbf{1} \leq \mu^s, \mathbf{T}^\top \mathbf{1} \leq \mu^t, \\ & T_{ii'} = 0 \text{ if } \phi^s(i) \neq \phi^t(i') \forall (i, i') \in \mathcal{V}^s \times \mathcal{V}^t, \\ & \left. \sum_{i: \phi^s(i)=r} \sum_{i'=1}^{|\mathcal{V}^t|} T_{ii'} = b_r \forall r \in \mathcal{T} \right\}, \end{aligned}$$

where the third constraint guarantees that nodes of different types are not matched and the fourth indicates that the transport mass for type r is b_r . α and β are constant scalars and the three components of the objective are described as follows.

Partial GW Discrepancy. The partial GW discrepancy is adopted to measure the sum of the pairwise matching costs. In heterogeneous graphs, nodes may not be adjacent to nodes of its own type. To take into account the in-type topological structure, we consider multi-hop connections. Specifically, we calculate a matrix that characterizes the K^{th} -order proximity as

$$\mathbf{B}^z = \sum_{l=1}^K ((\mathbf{D}^z)^{-1} \mathbf{W}^z)^l + (\mathbf{W}^z (\mathbf{D}^z)^{-1})^l, \quad (3)$$

where $z = s, t$ and $\mathbf{D}^z = [D_{ij}^z]$ is the diagonal degree matrix with the i^{th} entry given by $D_{ii}^z = \sum_{j=1}^{|\mathcal{V}^z|} W_{ij}^z$. The $(i, j)^{\text{th}}$ entry of matrix $((\mathbf{D}^z)^{-1} \mathbf{W}^z)^l$ is the probability that, at the l^{th} step, j is visited by a random walk starting from i (Qiu et al. 2018). Similar explanation applies to the $(i, j)^{\text{th}}$ entry of $(\mathbf{W}^z (\mathbf{D}^z)^{-1})^l$. We use *proximity matrix* $\mathbf{C}^z = [C_{ij}^z]$ to model the proximity between nodes i and j where

$$C_{ij}^z = \begin{cases} B_{ij}^z, & \text{if } i \neq j, \\ \bar{C}, & \text{otherwise,} \end{cases} \quad (4)$$

where $z = s, t$ and \bar{C} is a large enough constant.

Partial Wasserstein Discrepancy. The partial Wasserstein discrepancy is the sum of the unary matching costs based on the learned node embeddings. Herein, $\kappa(\mathbf{z}_i^s, \mathbf{z}_{i'}^t) = (\mathbf{z}_i^{s\top} \mathbf{z}_{i'}^t) / (\|\mathbf{z}_i^s\| \|\mathbf{z}_{i'}^t\|)$ is the normalized inner-product between node embeddings \mathbf{z}_i^s and $\mathbf{z}_{i'}^t$, and measures the similarity between nodes i and i' . By incorporating node embeddings, we improve the robustness to the structure noise. Note that, for attributed graphs in which each node is associated to a feature vector, heterogeneous graph neural networks (e.g., (Zhang et al. 2019)) can be used to incorporate attributes information and parameterize the node embeddings. For simplicity, this paper only considers unattributed graphs.

Regularizers. $R(\mathbf{Z}^z)$ for $z = s$ or t , regularizes the embedding by penalizing the distance between the inner-product of node embeddings and the proximity, and is defined as

$$R(\mathbf{Z}^z) = \sum_{i=1}^{|\mathcal{V}^z|} \sum_{j: \phi^z(i) = \phi^z(j), j \neq i} (C_{ij}^z - \mathbf{z}_i^{z\top} \mathbf{z}_j^z)^2.$$

The regularizers guarantee that node embeddings capture the global topological structure via preserving the total high-order proximity (Tang et al. 2015; Cao, Lu, and Xu 2015).

With the composition of the above three components, an appropriate algorithm can learn embedding vectors and node correspondence in a simultaneous manner. This joint learning formula has been shown to substantially improve the robustness to the noise (Xu et al. 2019; Karakasis, Konar, and Sidiropoulos 2021).

3.2 Model Decomposition: From One to All

We decompose the learning procedure into a series of easy-to-solve sub-procedures due to the following facts. **(i)** The complicated constraint $\Omega(\mathbf{b}, \mathcal{G}^s, \mathcal{G}^t)$ makes the model corresponding to (2) difficult to train. **(ii)** In several scenarios such as user alignment (Wang et al. 2019b), the matching of only a few primary types are of interest.

Before delving into the decomposition of the learning procedure, we first introduce some additional notations. Without loss of generality, we assume the node types are represented by integers, i.e., type $r \in \{1, 2, \dots, |\mathcal{T}|\}$. $\zeta^{z,r} = [\zeta_i^{z,r}]$ is the mask vector for type r and is defined as

$$\zeta_i^{z,r} = \begin{cases} 1, & \text{if } \phi^z(i) = r, \\ 0, & \text{otherwise.} \end{cases}$$

Based on the mask vectors, \mathbf{T} can be written as the sum of $|\mathcal{T}|$ matrices $\mathbf{T} = \sum_{r=1}^{|\mathcal{T}|} \mathbf{T}^r$ where $\mathbf{T}^r = \mathbf{T} \odot (\zeta^{s,r} \zeta^{t,r\top})$ is the matching for type- r nodes and \odot is the element-wise multiplication. Similarly, μ^s and μ^t can be written as the sum of *type measures*, i.e., $\mu^z = \sum_{r=1}^{|\mathcal{T}|} \mu^{z,r}$ where the type- r measure is $\mu^{z,r} = \mu^z \odot \zeta^{z,r}$. We further denote the set of type- r nodes and the corresponding embeddings by $\mathcal{V}^{z,r} = \{i | \phi^z(i) = r\}$ and $\mathbf{Z}^{z,r} = \{\mathbf{z}_i^z | \phi^z(i) = r\}$ respectively.

We now decompose the learning procedure for (2) into $|\mathcal{T}|$ sub-procedures, which is formally stated in the following proposition.

Proposition 1. *The learning procedure (2) can be decomposed into $|\mathcal{T}|$ sub-procedures. The r^{th} sub-procedure corresponds to the learning of type- r matching \mathbf{T}^r and the embeddings $\mathbf{Z}^{s,r}$ and $\mathbf{Z}^{t,r}$. The optimization problem is $\min_{\mathbf{Z}^{s,r}, \mathbf{Z}^{t,r}} \min_{\mathbf{T}^r} f^r(\mathbf{T}^r, \mathbf{Z}^{s,r}, \mathbf{Z}^{t,r})$ and the objective is given by*

$$\begin{aligned} f^r(\mathbf{T}^r, \mathbf{Z}^{s,r}, \mathbf{Z}^{t,r}) &= \sum_{i,j=1}^{|\mathcal{V}^s|} \sum_{i',j'=1}^{|\mathcal{V}^t|} (C_{ij}^s - C_{i'j'}^t)^2 T_{ii'}^r (T_{jj'}^r + 2 \sum_{a=1}^{r-1} \hat{T}_{jj'}^a) \\ &+ \alpha \sum_{i \in \mathcal{V}^{s,r}} \sum_{i' \in \mathcal{V}^{t,r}} -\kappa(\mathbf{z}_i^s, \mathbf{z}_{i'}^t) T_{ii'}^r + \beta (R^r(\mathbf{Z}^{s,r}) + R^r(\mathbf{Z}^{t,r})), \end{aligned} \quad (5)$$

where $\mathbf{T}^r \in \Pi^{b_r}(\mu^{s,r}, \mu^{t,r})$, $\hat{\mathbf{T}}^a = [\hat{T}_{jj'}^a]$ is the obtained matching for type- a nodes, and the regularizers are $R^r(\mathbf{Z}^{z,r}) = \sum_{i,j \in \mathcal{V}^{z,r}, i \neq j} (C_{ij}^z - \mathbf{z}_i^{z\top} \mathbf{z}_j^z)^2$, $z = s, t$.

The proof is deferred to the long version due to the limit of space. Intuitively, to match each type, the nodes already matched serve as seeds. Node embeddings that encode global topology are incorporated to enhance robustness to edge noise and reduce the cascade of errors.

$\Pi^{b_r}(\mu^{s,r}, \mu^{t,r})$ is a relatively simpler feasible domain. Hence, the sub-problem can be practically solved via alternating optimization, that is, alternatingly updating the transport plan and node embeddings, which is detailed as follows.

Updating the transport plan. Given current node embeddings $\mathbf{Z}^{s,r,(m)}$ and $\mathbf{Z}^{t,r,(m)}$, we solve the following problem,

$$\min_{\mathbf{T}^r \in \Pi^{b_r}(\mu^{s,r}, \mu^{t,r})} f^r(\mathbf{T}^r, \mathbf{Z}^{s,r,(m)}, \mathbf{Z}^{t,r,(m)}). \quad (6)$$

Minimization (6) is a partial OT problem and can be effectively solved using off-the-shelf OT algorithms.

Updating the embeddings. Given the calculated transport plan $\hat{\mathbf{T}}^{r,(m)}$, we update the embeddings. The sub-problem for updating embeddings is

$$\min_{\mathbf{Z}^{s,r}, \mathbf{Z}^{t,r}} \alpha \sum_{i \in \mathcal{V}^{s,r}} \sum_{i' \in \mathcal{V}^{t,r}} -\kappa(\mathbf{z}_i^s, \mathbf{z}_{i'}^t) T_{ii'}^{r,(m)} + \beta (R^r(\mathbf{Z}^{s,r}) + R^r(\mathbf{Z}^{t,r})), \quad (7)$$

which can be solved by SGD or its variants.

We summarize FOTA in Algorithm 1 and make some important remarks as follows.

1. **Extended transport plan.** The final learned transport plan $\hat{\mathbf{T}}$ is used to construct an extended transport plan $\tilde{\mathbf{T}} = [\hat{\mathbf{T}}, \boldsymbol{\mu}^s - \hat{\mathbf{T}}\mathbf{1}] \in \mathbb{R}^{|\mathcal{V}^s| \times (|\mathcal{V}^t|+1)}$. Nodes matched to the dummy node are considered to have no counterpart in the target graph. By choosing the largest \tilde{T}_{ij} for each i , we find the correspondence.
2. **Refinement.** The current algorithm uses $|\mathcal{T}|$ iterations to obtain the matching of each type, which is analogous to that of a single round of block coordinate descent. Since approximation errors exist in each iteration, empirically, we can repeat Line 6 to 12 in Alg 1 to refine the matching.

Algorithm 1: FOTA

```

1: Input:  $M$  rounds,  $K$ ,  $\alpha$ ,  $\beta$ , graphs  $\mathcal{G}^s$  and  $\mathcal{G}^t$ .
2: Output: Correspondence set  $\mathcal{D}$ .
3: Calculate  $\mathbf{C}^s$  and  $\mathbf{C}^t$  as Eq. (4).
4: Obtain transport mass for each type  $\mathbf{b}$  via Algorithm 2.
5: Initialize  $\mathbf{Z}^s$  and  $\mathbf{Z}^t$  via SVD for  $\mathbf{C}^s$  and  $\mathbf{C}^t$  respectively.
6: for  $r = 1, \dots, |\mathcal{T}|$  do
7:   for  $m = 1, \dots, M$  do
8:     Update transport plan by solving (6).
9:     Update node embeddings by solving (7).
10:   end for
11:    $\hat{\mathbf{T}}^r = \hat{\mathbf{T}}^{r,(M)}$ .
12: end for
13:  $\hat{\mathbf{T}} = \sum_{r=1}^{|\mathcal{T}|} \hat{\mathbf{T}}^r$ .
14:  $\tilde{\mathbf{T}} = [\hat{\mathbf{T}}, \boldsymbol{\mu}^s - \hat{\mathbf{T}}\mathbf{1}]$ .
15: Initialize correspondence set  $\mathcal{D} = \emptyset$ .
16: for  $i \in \{1, \dots, |\mathcal{V}^s|\}$  do
17:    $i' = \arg \max_{i'} \tilde{T}_{ii'}$ .
18:   if  $i' \neq (|\mathcal{V}^t| + 1)$  then
19:      $\mathcal{D} = \mathcal{D} \cup \{(i, i')\}$ .
20:   end if
21: end for

```

3.3 Transport Mass Search

Intuitively, the partial GW discrepancy is very small if the chosen b_r is less than the underlying transport mass. It soars when b_r begins to exceed the underlying transport mass, since matching more non-overlapped nodes with disparate topology incurs larger penalty. Therefore, for each type r , one can conduct a line search over

Algorithm 2: LineSearch

```

1: Input: type  $r$ , line search resolution  $\delta$ , number  $n$ , search range  $(b_{\min}, b_{\max}]$ , threshold  $\gamma$ .
2: Output:  $\hat{b}_r$ .
3: for  $i = 1, \dots, n$  do
4:    $b_i = b_{\min} + \frac{i}{n}(b_{\max} - b_{\min})$ .
5:   Obtain the approximation  $\hat{A}(b_i)$  for  $A(b_i)$ .
6:   if  $\hat{A}(b_i) \geq \frac{1+\gamma}{i} \sum_{j=1}^i \hat{A}(b_j)$  then
7:     if  $b_{\max} - b_{\min} > \delta$  then
8:       LineSearch( $r, \delta, n, (b_{i-1}, b_i], \gamma$ ).
9:     else
10:       $\hat{b}_r = b_{\min}$ .
11:    return
12:  end if
13: end if
14: end for

```

Datasets	$\rho=1.0$	$\rho=0.8$	$\rho=0.6$	Description
Arenas Email	$ \mathcal{V}^s $	1133	986	844
	$ \mathcal{E}^s $	10902	8694	6760
	$ \mathcal{V}^t $	1133	992	868
	$ \mathcal{E}^t $	10902	9022	7346
PPI Yeast	$ \mathcal{V}^s $	1004	835	628
	$ \mathcal{E}^s $	16646	13406	8474
	$ \mathcal{V}^t $	1004	844	661
	$ \mathcal{E}^t $	16646	13270	10396
Arxiv	$ \mathcal{V}^s $	18772	15667	13757
	$ \mathcal{E}^s $	396160	318846	261206
	$ \mathcal{V}^t $	18772	15933	14073
	$ \mathcal{E}^t $	396160	327403	267646

Table 1: Statistics of homogeneous graphs used in our experiments.

$(0, \min\{\|\boldsymbol{\mu}^{s,r}\|_1, \|\boldsymbol{\mu}^{t,r}\|_1\})$ and find the turning point \hat{b}_r of the partial GW discrepancy

$$A(b_r) := \min_{\mathbf{T} \in \Pi^{b_r}} \sum_{i,j,i',j'} (C_{ij}^s - C_{i'j'}^t)^2 T_{ii'} T_{jj'}. \quad (8)$$

Specifically, we adopt a recursive strategy. We search the range $(b_{\min}, b_{\max}]$ by evaluating $A(b_i)$ where $b_i = b_{\min} + \frac{i}{n}(b_{\max} - b_{\min})$ and n is a preset number of samples. When $A(b_i) > \frac{1+\gamma}{i} \sum_{j=1}^i A(b_j)$ where γ is a given threshold, the optimal \hat{b}_r is believed to fall into the range $(b_{i-1}, b_i]$. Then we set $b_{\min} = b_{i-1}$ and $b_{\max} = b_i$, and repeat the above procedure until $b_{\max} - b_{\min} < \delta$ where δ is a small number. In the beginning of the recursive line search, $b_{\min} = 0$ and $b_{\max} = \min\{\|\boldsymbol{\mu}^{s,r}\|_1, \|\boldsymbol{\mu}^{t,r}\|_1\}$. We summarize this line search strategy in Algorithm 2.

3.4 Complexity Analysis

The computational costs can be divided into four parts. (i) The cost for calculating the K^{th} -order proximity matrices \mathbf{C}^s and \mathbf{C}^t is $\mathcal{O}(KVE)$ where $V = \max\{|\mathcal{V}^s|, |\mathcal{V}^t|\}$ and $E = \max\{|\mathcal{E}^s|, |\mathcal{E}^t|\}$. (ii) When updating the transport plan, the gradient of $f^r(\mathbf{T}^r, \mathbf{Z}^{s,(m)}, \mathbf{Z}^{t,(m)})$ in (6) takes the

Datasets		$\rho=1.0$	$\rho=0.8$	$\rho=0.6$
Movies	\mathcal{V}^s	0:348, 1:389, 2:257, 3:6	0:300, 1:306, 2:202, 3:3	0:237, 1:234, 2:159, 3:3
	\mathcal{E}^s	4618	3476	2612
	\mathcal{V}^t	0:348, 1:389, 2:257, 3:6	0:307, 1:329, 2:212, 3:3	0:261, 1:230, 2:150, 3:4
	\mathcal{E}^t	4618	3986	3080
PubMed	\mathcal{V}^s	0:1059, 1:1096, 2:1176, 3:669	0:908, 1:944, 2:1028, 3:572	0:808, 1:801, 2:908, 3:520
	\mathcal{E}^s	18982	15895	13527
	\mathcal{V}^t	0:1059, 1:1096, 2:1176, 3:669	0:925, 1:966, 2:1038, 3:591	0:830, 1:851, 2:935, 3:519
	\mathcal{E}^t	18982	16552	13801
DBLP	\mathcal{V}^s	0:3067, 1:7278, 2:1598, 3:57	0:1558, 1:4035, 2:956, 3:47	0:1416, 1:3385, 2:820, 3:31
	\mathcal{E}^s	118063	51415	39016
	\mathcal{V}^t	0:3067, 1:7278, 2:1598, 3:57	0:1654, 1:4083, 2:951, 3:43	0:1453, 1:3559, 2:825, 3:48
	\mathcal{E}^t	118063	50847	45727

Table 2: Statistics of heterogeneous graphs used in our experiments. The information of node types is included. For example, 0:348 means 348 nodes in this graph are of type 0.

form (Peyré, Cuturi, and Solomon 2016; Xu et al. 2019)

$$\begin{aligned} \nabla f^r(\mathbf{T}^{r,(m)}, \mathbf{Z}^{s,(m)}, \mathbf{Z}^{t,(m)}) &= h(\mathbf{C}^s) \mathbf{Y}^{r,(m)} \mathbf{1}^{|\mathcal{V}^t|} \mathbf{1}^{|\mathcal{V}^t|^\top} \\ &+ \mathbf{1}^{|\mathcal{V}^s|} \left(\mathbf{Y}^{r,(m)} \mathbf{1}^{|\mathcal{V}^s|} \right)^\top h(\mathbf{C}^t) \\ &- 2\mathbf{C}^s \mathbf{Y}^{r,(m)} \mathbf{C}^t - \alpha \mathbf{K}^{(m)}, \end{aligned} \quad (9)$$

where $h(\cdot)$ is the element-wise square operation and we use matrix notations $\mathbf{Y}^{r,(m)} = \mathbf{T}^{r,(m)} + 2 \sum_{a=1}^{r-1} \hat{\mathbf{T}}^a$ and $\mathbf{K}^{(m)} = [\kappa(\mathbf{z}_i^{s,(m)}, \mathbf{z}_{i'}^{t,(m)})]$. Adopting the k -rank approximations for \mathbf{C}^s and \mathbf{C}^t , the cost for computing $\nabla f^r(\mathbf{T}^{r,(m)}, \mathbf{Z}^{s,(m)}, \mathbf{Z}^{t,(m)})$ is $\mathcal{O}((k+d)V^2)$ (Scetbon, Peyré, and Cuturi 2021), where d is the dimension of node embeddings. The complexity for obtaining the k -rank approximation for \mathbf{C}^s and \mathbf{C}^t via SVD is $\mathcal{O}(kV^2)$ (Golub and Van Loan 1989; Halko, Martinsson, and Tropp 2011). Problem (6) can be solved by *mirror descent* (Bubeck et al. 2015; Peyré, Cuturi, and Solomon 2016) which involves *iterative Bregman projections* (Benamou et al. 2015). If we run mirror descent for N iterations in total for learning the transport plan, each of which involves T matrix-vector multiplications in the projection, the complexity for updating the transport plan is $\mathcal{O}(N(T+k+d)V^2)$. (iii) For learning the embeddings, by selecting the size of node batch as $B \ll V$, the complexity for updating the embeddings is $\mathcal{O}(VBd)$ (Xu et al. 2019) and can be ignored compared to that of learning the transport plan. (iv) With reasonable δ and n , the complexity for line search is of the same order as for learning the transport plan. Therefore, the overall complexity is $\mathcal{O}(KVE + N(T+k+d)V^2)$.

4 Experiments

We compare FOTA with state-of-the-art methods on both homogeneous and heterogeneous graphs. The experiments are conducted on a Ubuntu 18.04 server with a 24-core 2.70GHz Intel Xeon Platinum 8163 CPU, an NVIDIA Tesla V100 GPU, and 92 GB RAM. The source code is written in Python 3.6 and C++.

4.1 Experimental Setup

Baselines. The baselines can be divided into three families: 1) State-of-the-art methods for matching homogeneous graphs including REGAL (Heimann et al. 2018), GDD (Scott and Mjolsness 2021), GRAMPA (Fan et al. 2020), GWL (Xu et al. 2019), MM (Konar and Sidiropoulos 2020), SpectralPivot (Karakasis, Konar, and Sidiropoulos 2021); 2) SANA (Gu et al. 2018), which extends Mamano and Hayes (2017) by adopting colored graphlet degree vector features to match heterogeneous graphs²; 3) Methods that treat heterogeneous graphs as *homogeneous attributed* graphs, including VELSET (Dutta, Nayek, and Bhattacharya 2017) and G-Finder (Liu et al. 2019). We also conduct ablation studies. The variants of FOTA include 1) FOTA-GW which only uses the partial GW discrepancy by setting $\alpha = 0$ and thus does not involve the embedding learning, and 2) FOTA-W which only uses the partial Wasserstein discrepancy.

Metrics. For evaluation of partial graph matching, we compute the commonly used indicators (see e.g. (Sarlin et al. 2020; Wang et al. 2019b)),

$$\begin{aligned} \text{recall} &= \frac{\#\{\text{correct matching}\}}{\#\{\text{ground truth matching}\}}, \\ \text{precision} &= \frac{\#\{\text{correct matching}\}}{\#\{\text{total predicted matching}\}}, \\ \text{F1} &= \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}. \end{aligned}$$

On experiments of matching heterogeneous graphs, we also report the ratio of type mismatch

$$q = \frac{|\{(i, \hat{i}) | \phi^s(i) \neq \phi^t(\hat{i})\}|}{\#\{\text{ground truth matching}\}},$$

where \hat{i} is the output counterpart of node i predicted by each method. We run each method for 5 times and report both the average values and standard deviations.

²Gu et al. (2018) extend three homogeneous graph matching methods to the heterogeneous variants. SANA is shown to achieve the best performance among them.

ρ	Methods	Arenas Email			PPI Yeast			Arxiv		
		recall	precision	F1	recall	precision	F1	recall	precision	F1
1.0	REGAL	97.3±0.0	97.3±0.0	97.3±0.0	81.1±0.0	81.1±0.0	81.1±0.0	77.9±0.0	77.9±0.0	77.9±0.0
	GDD	24.2±0.0	24.2±0.0	24.2±0.0	28.3±0.0	28.3±0.0	28.3±0.0	18.7±0.0	18.7±0.0	18.7±0.0
	GRAMPA	40.9±0.0	40.9±0.0	40.9±0.0	32.4±0.0	32.4±0.0	32.4±0.0	/	/	/
	GWL	95.4±0.3	95.4±0.3	95.4±0.3	84.9±0.6	84.9±0.6	84.9±0.6	78.5±0.1	78.5±0.1	78.5±0.1
	MM	97.3±0.0	97.3±0.0	97.3±0.0	80.9±0.0	80.9±0.0	80.9±0.0	77.8±0.0	77.8±0.0	77.8±0.0
	SpectralPivot	97.4±0.0	97.4±0.0	97.4±0.0	84.0±0.0	84.0±0.0	84.0±0.0	73.5±0.0	73.5±0.0	73.5±0.0
	FOTA-GW	97.9±0.0	97.9±0.0	97.9±0.0	85.5±0.0	85.5±0.0	85.5±0.0	79.0±0.0	79.0±0.0	79.0±0.0
	FOTA-W	5.1±0.3	5.1±0.3	5.1±0.3	2.6±0.2	2.6±0.2	2.6±0.2	0.5±0.0	0.5±0.0	0.5±0.0
	FOTA	98.4±0.0	98.4±0.0	98.4±0.0	86.5±0.1	86.5±0.1	86.5±0.1	79.1±0.0	79.1±0.0	79.1±0.0
0.8	REGAL	28.4±0.7	25.1±0.6	26.6±0.7	27.4±1.1	24.5±1.0	25.9±1.0	24.1±0.1	21.7±0.1	22.8±0.1
	GDD	2.4±0.0	2.1±0.0	2.3±0.0	3.5±0.0	3.1±0.0	3.3±0.0	0.4±0.0	0.3±0.0	0.3±0.0
	GRAMPA	9.3±0.0	8.2±0.0	8.7±0.0	15.5±0.0	13.8±0.0	14.6±0.0	/	/	/
	GWL	91.8±0.8	81.5±1.4	86.4±0.9	55.3±1.3	49.3±1.2	52.1±1.2	62.7±0.2	56.2±0.2	59.3±0.2
	MM	26.0±0.9	22.9±0.8	24.3±0.9	25.1±1.3	22.3±1.2	23.6±1.3	22.0±0.1	19.7±0.1	20.8±0.1
	SpectralPivot	89.3±0.1	78.7±0.1	83.6±0.1	67.6±0.8	60.2±0.7	63.7±0.7	16.4±0.3	14.7±0.3	15.5±0.3
	FOTA-GW	89.8±0.0	95.2±0.0	92.4±0.0	68.8±0.0	67.6±0.0	68.2±0.0	70.4±0.0	79.2±0.0	74.6±0.0
	FOTA-W	0.8±0.0	0.8±0.0	0.8±0.0	0.2±0.0	0.2±0.0	0.2±0.0	0.2±0.1	0.2±0.1	0.2±0.1
	FOTA	90.6±0.1	96.8±0.7	93.6±0.3	69.5±0.0	67.7±0.1	68.6±0.0	71.2±0.1	79.2±0.1	75.0±0.1
0.6	REGAL	7.0±0.5	5.3±0.4	6.0±0.5	19.1±0.9	14.7±0.7	16.6±0.8	9.7±0.1	7.4±0.1	8.4±0.1
	GDD	1.2±0.0	0.9±0.0	1.1±0.0	1.9±0.0	1.4±0.0	1.6±0.0	0.2±0.0	0.2±0.0	0.2±0.0
	GRAMPA	1.1±0.0	0.8±0.0	0.9±0.0	5.8±0.0	4.5±0.0	5.0±0.0	/	/	/
	GWL	4.5±2.7	3.5±1.9	3.5±1.1	34.9±0.6	26.8±0.4	30.3±0.5	53.2±0.6	40.6±0.5	46.1±0.5
	MM	4.6±0.5	3.5±0.4	3.9±0.4	17.3±1.0	13.3±0.8	15.1±0.9	7.8±0.1	6.0±0.1	6.8±0.1
	SpectralPivot	10.7±1.6	8.1±1.2	9.2±1.4	33.6±0.7	25.8±0.5	29.2±0.6	6.2±0.2	4.7±0.2	5.4±0.2
	FOTA-GW	15.0±0.0	15.2±0.0	15.1±0.0	42.1±0.0	41.3±0.0	41.7±0.0	62.6±0.0	68.6±0.0	65.5±0.0
	FOTA-W	0.0±0.0	0.0±0.0	0.0±0.0	0.4±0.0	0.4±0.0	0.4±0.0	0.2±0.0	0.2±0.0	0.2±0.0
	FOTA	17.3±0.3	17.4±0.3	17.3±0.3	53.0±0.2	50.2±0.2	51.6±0.2	63.2±0.2	70.2±0.0	66.5±0.1

Table 3: Recall, precision and F1 scores with standard deviations on homogeneous graphs (in percent). The similarity matrix in GRAMPA incurs quadruple computational complexity and takes thousands of hours to obtain on Arxiv. Its performance is thus not reported on this dataset.

Dataset Preparation. We extract fully or partially overlapped subgraphs from benchmark datasets. Mathematically, the overlap ratio is defined as $\rho = |\mathcal{V}^s \cap \mathcal{V}^t| / |\mathcal{V}^s \cup \mathcal{V}^t|$. As ρ decreases, the matching problem becomes more difficult. We verify the efficacy of FOTA on graphs extracted from three homogeneous graphs, including Arenas Email³, PPI Yeast⁴, and Arxiv⁵. We then compare the performance of FOTA against baselines on heterogeneous graphs, including Movie⁶, PubMed⁷ and DBLP⁸. Movie contains four node types, including actors, movies, directors and composers. PubMed is a network of genes, diseases, chemicals, and species. DBLP is an academic network containing authors, papers, venues and phrases. Statistics of extracted subgraphs are listed in Table 1 and Table 2.

Parameter choices. The transport mass is selected by line search as is stated in Sec. 3.3. In all experiments, the embedding dimension is set as $d = 64$. Setting $1 \times 10^{-7} \leq \alpha \leq 1 \times 10^{-4}$ for FOTA yields improved performance over

FOTA-GW. The results in Tables 3, 4 and 5 are obtained with $\alpha = 1 \times 10^{-5}$. We tested β in $\{1, 0.1, 0.01, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}\}$. $1 \times 10^{-3} \leq \beta \leq 1$ achieves stable performance and thus we set $\beta = 0.01$.

4.2 Matching Homogeneous Graphs

The recall, precision, and F1 scores on homogeneous graphs are shown in Table 3. FOTA and FOTA-GW consistently outperform baselines in terms of the F1 indicator and the advantage becomes more significant with the overlap ratio decreasing, which demonstrates the effectiveness of partial optimal transport. GWL and SpectralPivot are closest competitors. Because GWL is a full matching method, it occasionally outperforms FOTA in terms of recall by a small margin. However, it is widely known that F1 is a better measure as it balances precision and recall (van Rijsbergen 1979; Fawcett 2006). The partial optimal transport allows FOTA to significantly boost the precision and F1. For partially overlapped graphs, FOTA improves the precision and F1 over the best baseline by *at least* 12.5% and 7.7% respectively and often much more. The ablation study indicates that the superior performance of FOTA is mainly attributed to the partial GW discrepancy, since FOTA-W has unsatisfying results. Node embeddings improve the matching performance of FOTA.

³<http://konect.cc/networks/arenas-email/>

⁴<https://www3.nd.edu/~cone/MAGNA++/>

⁵<http://snap.stanford.edu/data/ca-AstroPh.html>

⁶<https://github.com/eXascaleInfolab/JUST/tree/master/Datasets/Movies>

⁷<https://pubmed.ncbi.nlm.nih.gov/>

⁸<https://dblp.uni-trier.de/>

ρ	Methods	Movie			PubMed			DBLP		
		recall	precision	F1	recall	precision	F1	recall	precision	F1
1.0	REGAL	73.7±0.0	73.7±0.0	73.7±0.0	60.3±0.0	60.3±0.0	60.3±0.0	81.0±0.0	81.0±0.0	81.0±0.0
	GDD	21.9±0.0	21.9±0.0	21.9±0.0	16.4±0.0	16.4±0.0	61.3±0.0	17.7±0.0	17.7±0.0	17.7±0.0
	GRAMPA	12.3±0.0	12.3±0.0	12.3±0.0	/	/	/	/	/	/
	GWL	88.2±1.4	88.2±1.4	88.2±1.4	61.2±0.8	61.2±0.8	61.2±0.8	81.6±0.1	81.6±0.1	81.6±0.1
	MM	73.5±0.1	73.5±0.1	73.5±0.1	60.2±0.0	60.2±0.0	60.2±0.0	81.0±0.0	81.0±0.0	81.0±0.0
	SpectralPivot	87.6±0.5	87.6±0.5	87.6±0.5	59.6±0.7	59.6±0.7	59.6±0.7	78.9±0.1	78.9±0.1	78.9±0.1
	SANA	65.8±0.5	65.8±0.5	65.8±0.5	/	/	/	/	/	/
	VELSET	4.7±0.0	4.7±0.0	4.7±0.0	5.7±0.0	5.7±0.0	5.7±0.0	0.5±0.0	0.5±0.0	0.5±0.0
	G-Finder	1.7±0.0	3.3±0.0	2.2±0.0	/	/	/	/	/	/
	FOTA-GW	91.6±0.0	91.6±0.0	91.6±0.0	65.5±0.0	65.5±0.0	65.5±0.0	81.2±0.0	81.2±0.0	81.2±0.0
	FOTA-W	17.3±3.6	17.3±3.6	17.3±3.6	1.1±0.1	1.1±0.1	1.1±0.1	0.3±0.0	0.3±0.0	0.3±0.0
	FOTA	93.7±0.3	93.7±0.3	93.7±0.3	68.0±0.0	68.0±0.0	68.0±0.0	81.6±0.0	81.6±0.0	81.6±0.0
0.8	REGAL	32.8±1.2	29.6±1.1	31.1±1.1	35.0±0.3	31.5±0.3	33.2±0.3	19.5±0.3	17.5±0.3	18.5±0.3
	GDD	2.9±0.0	2.6±0.0	2.7±0.0	1.7±0.0	1.5±0.0	1.6±0.0	0.6±0.0	0.5±0.0	0.5±0.0
	GRAMPA	6.0±0.0	5.4±0.0	5.7±0.0	/	/	/	/	/	/
	GWL	73.0±2.6	66.0±2.3	69.3±2.4	48.1±0.2	43.3±0.1	45.6±0.1	39.3±1.5	34.9±1.1	37.0±1.3
	MM	29.3±1.4	26.5±1.3	27.8±1.3	33.5±0.6	30.1±0.6	31.7±0.6	16.0±0.6	14.3±0.5	15.1±0.5
	SpectralPivot	72.0±1.9	65.0±1.7	68.3±1.8	47.7±0.4	43.0±0.4	45.2±0.4	24.5±9.4	21.9±8.4	23.1±8.8
	SANA	49.8±1.5	45.0±1.4	47.3±1.5	/	/	/	/	/	/
	VELSET	4.4±0.0	4.0±0.0	4.1±0.0	4.5±0.0	4.1±0.0	4.3±0.0	0.5±0.0	0.4±0.0	0.4±0.0
	G-Finder	1.1±0.0	3.1±0.0	1.6±0.0	/	/	/	/	/	/
	FOTA-GW	74.4±0.0	71.9±0.0	73.1±0.0	49.3±0.0	50.5±0.0	49.9±0.0	32.9±0.0	46.6±0.0	38.6±0.0
	FOTA-W	2.0±0.1	2.7±0.4	2.3±0.2	0.5±0.1	0.5±0.1	0.5±0.1	0.5±0.0	0.5±0.0	0.5±0.0
	FOTA	75.7±0.0	72.2±0.2	73.9±0.1	53.7±0.0	52.6±0.0	53.1±0.0	41.7±1.0	59.7±1.3	49.1±1.2
0.6	REGAL	19.5±0.7	14.8±0.6	16.9±0.6	18.1±0.5	14.0±0.4	15.8±0.4	10.4±0.2	7.9±0.2	9.0±0.2
	GDD	2.1±0.0	1.6±0.0	1.8±0.0	0.8±0.0	0.6±0.0	0.7±0.0	0.6±0.0	0.4±0.0	0.5±0.0
	GRAMPA	3.1±0.0	2.4±0.0	2.7±0.0	/	/	/	/	/	/
	GWL	60.6±3.5	46.1±2.7	52.4±3.0	35.2±0.2	27.2±0.1	30.7±0.2	29.3±0.3	22.3±0.2	25.3±0.3
	MM	14.9±0.7	11.3±0.5	12.9±0.6	15.4±0.3	11.9±0.3	13.4±0.3	7.8±0.6	5.9±0.4	6.8±0.5
	SpectralPivot	41.1±2.6	31.2±2.0	35.5±2.2	33.0±0.8	25.5±0.6	28.7±0.7	17.8±1.4	13.5±1.1	15.4±1.2
	SANA	30.1±1.3	22.9±1.0	26.0±1.1	/	/	/	/	/	/
	VELSET	4.2±0.0	3.2±0.0	3.6±0.0	3.8±0.0	2.9±0.0	3.3±0.0	0.4±0.0	0.3±0.0	0.3±0.0
	G-Finder	0.8±0.0	2.2±0.0	1.2±0.0	/	/	/	/	/	/
	FOTA-GW	76.1±0.0	62.9±0.0	68.9±0.0	33.3±0.0	29.9±0.0	31.5±0.0	27.0±0.0	31.0±0.0	28.8±0.0
	FOTA-W	1.1±0.4	0.9±0.3	1.0±0.4	0.3±0.1	0.3±0.1	0.3±0.1	0.1±0.0	0.1±0.0	0.1±0.0
	FOTA	77.9±0.3	63.7±0.1	70.1±0.2	37.1±0.0	33.3±0.0	35.1±0.0	27.9±0.1	32.7±0.2	30.1±0.1

Table 4: Recall, precision, and F1 type mismatch ratio q scores with standard deviations on heterogeneous graphs (in percent).

4.3 Matching Heterogeneous Graphs

The performance of FOTA and baselines on heterogeneous graphs are reported in Tables 4 and 5. Due to the complexity of GRAMPA, SANA and G-Finder, we evaluate them only in the first test on the smaller Movie dataset but not in the remaining two tests on the larger PubMed and DBLP graphs. Other methods are evaluated on all three tests. FOTA and FOTA-GW outperform baselines in terms of precision and F1 indicators on all datasets.

All homogeneous graph matching methods can match nodes of different types. On Movie and PubMed, they match about half of the nodes in the source graph to nodes of different types in the target graph. Therefore, they cannot be directly used to match heterogeneous graphs and the type information should be explicitly considered. SANA, VELSET and G-Finder outperform these methods in terms of the type mismatch ratio. However, type mismatch still occurs. By contrast, type mismatch do not happen for FOTA and its variants.

5 Related Work

Matching homogeneous graphs. The matching costs $k_{ii'}$ and $d_{ii'jj'}$ in (1) are critical to the matching accuracy. In some early works, these costs are based on handcrafted features that rely heavily on expert knowledge (Mamano and Hayes 2017; Heimann et al. 2018). More recently, end-to-end deep learning frameworks for graph matching are proposed to automatically learn the node embedding-based assignment costs (Zanfir and Sminchisescu 2018; Wang, Yan, and Yang 2019; Wang et al. 2019a), which however are supervised and require a large amount of ground truth node pairs to be available. OT-based methods propose to exploit geometrical properties of the metric space of the graph in order to estimate the node correspondence in an unsupervised/semi-supervised manner and thus reduce the demand of data labeling (Maretic et al. 2019; Xu et al. 2019; Titouan et al. 2019).

Matching heterogeneous Graphs. Many existing methods for matching heterogeneous graphs are supervised (Wu et al. 2019; Ren, Meng, and Zhang 2020; Sun et al. 2020;

Datasets	Movie			PubMed			DBLP		
ρ	1.0	0.8	0.6	1.0	0.8	0.6	1.0	0.8	0.6
REGAL	12.2±0.9	27.4±1.5	34.2±1.3	29.1±0.5	50.9±0.4	63.0±1.0	0.1±0.0	3.4±0.1	6.6±0.3
GDD	37.9±0.0	55.9±0.0	55.9±0.0	61.3±0.0	72.8±0.0	73.4±0.0	33.8±0.0	49.5±0.0	52.1±0.0
GRAMPA	50.5±0.0	57.8±0.0	51.5±0.0	/	/	/	/	/	/
GWL	4.5±0.8	16.5±1.4	25.2±1.6	28.6±0.7	42.9±0.2	54.7±0.2	0.1±0.0	7.2±0.8	5.0±0.6
MM	10.9±0.0	37.1±2.0	47.5±0.9	29.1±0.5	51.8±0.6	64.4±1.0	0.3±0.0	19.2±0.9	24.2±0.9
SpectralPivot	4.8±0.3	16.7±1.1	32.3±2.0	30.5±0.4	42.9±0.5	55.3±0.3	4.5±0.3	21.0±6.3	20.7±0.3
SANA	10.0±0.6	6.9±0.7	8.9±0.5	/	/	/	/	/	/
VELSET	0.0±0.0	0.0±0.0	2.1±0.0	0.0±0.0	0.0±0.0	0.1±0.0	0.1±0.0	0.1±0.0	0.3±0.0
G-Finder	0.0±0.0	0.0±0.0	0.0±0.0	/	/	/	/	/	/
FOTA-GW	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
FOTA-W	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
FOTA	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0

Table 5: Type mismatch ratio q results with standard deviations on heterogeneous graphs (in percent).

Wang, Yang, and Ye 2020). Although Zhang et al. 2021 propose an unsupervised method, the purpose is different from ours. Concretely, the nodes represent persons captured by different cameras, and are divided into two types according to whether the appearance is clear. The node type mismatch is allowed or even encouraged, since a person is clear in one camera may not be clear in another. Some methods (Dutta, Nayek, and Bhattacharya 2017; Liu et al. 2019) treat the node type as an one-dimensional node attribute. Matching heterogeneous graphs is then converted into the problem of matching homogeneous *attributed* graphs, which may still match nodes of different types. Besides, these methods are often time-consuming due to the complex matching procedures.

Conclusion

In this paper, we propose the first practical method to match both heterogeneous graphs and partially overlapped graphs. The learning procedure is decomposed into a series of sub-procedures, each of which matches one type of nodes by solving a partial optimal transport problem. The nodes that are already matched serve as seeds. Such a matching strategy is a hybrid of the *seed-and-extend* strategy and the *searching* strategy. Empirical results demonstrate that our method outperforms state-of-the-art graph matching methods on both homogeneous and heterogeneous graphs.

Acknowledgments

This work is supported by National Key Research and Development Program of China under Grant 2020AAA0107400, Zhejiang Provincial Natural Science Foundation of China (Grant No: LZ18F020002, LR19F020005), Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, and National Natural Science Foundation of China (Grant No: 61672376, 61751209, 61472347, 61972347).

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barbe, A.; Sebban, M.; Gonçalves, P.; Borgnat, P.; and Gri-bonval, R. 2020. Graph Diffusion Wasserstein Distances. In *ECML PKDD*.

Benamou, J.-D.; Carlier, G.; Cuturi, M.; Nenna, L.; and Peyré, G. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2): A1111–A1138.

Berretti, S.; Del Bimbo, A.; and Vicario, E. 2001. Efficient matching and indexing of graph models in content-based retrieval. *IEEE TPAMI*, 23(10): 1089–1105.

Bubeck, S.; et al. 2015. Convex Optimization: Algorithms and Complexity. *Foundations and Trends® in Machine Learning*, 8(3-4): 231–357.

Caetano, T. S.; McAuley, J. J.; Cheng, L.; Le, Q. V.; and Smola, A. J. 2009. Learning graph matching. *IEEE TPAMI*, 31(6): 1048–1058.

Caffarelli, L. A.; and McCann, R. J. 2010. Free boundaries in optimal transport and Monge-Ampère obstacle problems. *Annals of mathematics*, 673–730.

Cao, S.; Lu, W.; and Xu, Q. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, 891–900.

Chapel, L.; Gasso, G.; et al. 2020. Partial Optimal Transport with applications on Positive-Unlabeled Learning. In *NeurIPS*.

Chen, L.; Gan, Z.; Cheng, Y.; Li, L.; Carin, L.; and Liu, J. 2020. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, 1542–1553. PMLR.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2292–2300.

Dutta, S.; Nayek, P.; and Bhattacharya, A. 2017. Neighbor-aware search for approximate labeled graph matching using the chi-square statistics. In *Proceedings of the 26th International Conference on World Wide Web*, 1281–1290.

Fan, Z.; Mao, C.; Wu, Y.; and Xu, J. 2020. Spectral Graph Matching and Regularized Quadratic Relaxations: Algorithm and Theory. In *International Conference on Machine Learning*, 2985–2995. PMLR.

- Fawcett, T. 2006. An introduction to ROC analysis. *Pattern recognition letters*, 27(8): 861–874.
- Figalli, A. 2010. The optimal partial transport problem. *Archive for rational mechanics and analysis*.
- Golub, G. H.; and Van Loan, C. F. 1989. *Matrix computations*. JHU press.
- Gu, S.; Johnson, J.; Faisal, F. E.; and Milenković, T. 2018. From homogeneous to heterogeneous network alignment via colored graphlets. *Scientific reports*.
- Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288.
- Heimann, M.; Shen, H.; Safavi, T.; and Koutra, D. 2018. Regal: Representation learning-based graph alignment. In *CIKM*.
- Karakasis, P. A.; Konar, A.; and Sidiropoulos, N. D. 2021. Joint Graph Embedding and Alignment with Spectral Pivot. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 851–859.
- Kazemi, E. 2016. Network alignment: Theory, algorithms, and applications. Technical report, EPFL.
- Kazemi, E.; Yartseva, L.; and Grossglauser, M. 2015. When can two unlabeled networks be aligned under partial overlap? In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*.
- Konar, A.; and Sidiropoulos, N. D. 2020. Graph Matching via the lens of Supermodularity. *IEEE Transactions on Knowledge and Data Engineering*.
- Li, S.; Li, X.; Ye, R.; Wang, M.; Su, H.; and Ou, Y. 2018. Non-translational Alignment for Multi-relational Networks. In *IJCAI*, 4180–4186.
- Liu, L.; Du, B.; Tong, H.; et al. 2019. G-finder: Approximate attributed subgraph matching. In *2019 IEEE International Conference on Big Data (Big Data)*, 513–522. IEEE.
- Mamano, N.; and Hayes, W. B. 2017. SANA: simulated annealing far outperforms many other search algorithms for biological network alignment. *Bioinformatics*.
- Maretic, H. P.; El Gheche, M.; Chierchia, G.; and Frossard, P. 2019. GOT: an optimal transport framework for graph comparison. In *NeurIPS*.
- Mémoli, F. 2011. Gromov–Wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4): 417–487.
- Narayanan, A.; and Shmatikov, V. 2009. De-anonymizing social networks. In *2009 30th IEEE symposium on security and privacy*, 173–187. IEEE.
- Özer, I. B.; Wolf, W.; and Akansu, A. N. 2002. A graph-based object description for information retrieval in digital image and video libraries. *JVCIR*.
- Pedarsani, P.; and Grossglauser, M. 2011. On the privacy of anonymized networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1235–1243.
- Peyré, G.; Cuturi, M.; and Solomon, J. 2016. Gromov-wasserstein averaging of kernel and distance matrices. In *ICML*.
- Qiu, J.; Dong, Y.; Ma, H.; Li, J.; Wang, K.; and Tang, J. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 459–467.
- Ren, Y.; Meng, L.; and Zhang, J. 2020. Scalable Heterogeneous Social Network Alignment through Synergistic Graph Partition. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, 261–270.
- Rolínek, M.; Swoboda, P.; Zietlow, D.; Paulus, A.; Musil, V.; and Martius, G. 2020. Deep Graph Matching via Blackbox Differentiation of Combinatorial Solvers. In *ECCV*.
- Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. Superglue: Learning feature matching with graph neural networks. In *CVPR*.
- Scetbon, M.; Peyré, G.; and Cuturi, M. 2021. Linear-Time Gromov Wasserstein Distances using Low Rank Couplings and Costs. *arXiv preprint arXiv:2106.01128*.
- Scott, C.; and Mjolsness, E. 2021. Graph diffusion distance: Properties and efficient computation. *Plos one*, 16(4): e0249624.
- Sharan, R.; and Ideker, T. 2006. Modeling cellular machinery through biological network comparison. *Nature biotechnology*, 24(4): 427–433.
- Sun, Y.; Crawford, J.; Tang, J.; and Milenković, T. 2015. Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *WABI*.
- Sun, Z.; Wang, C.; Hu, W.; Chen, M.; Dai, J.; Zhang, W.; and Qu, Y. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI*.
- Swoboda, P.; Mokarian, A.; Theobalt, C.; Bernard, F.; et al. 2019. A convex relaxation for multi-graph matching. In *CVPR*.
- Swoboda, P.; Rother, C.; Abu Alhaija, H.; Kainmuller, D.; and Savchynskyy, B. 2017. A study of lagrangean decompositions and dual ascent solvers for graph matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1607–1616.
- Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, 1067–1077.
- Titouan, V.; Courty, N.; Tavenard, R.; and Flamary, R. 2019. Optimal Transport for structured data with application on graphs. In *ICML*.
- van Rijsbergen, C. 1979. *Information Retrieval*, 2nd edButterworths.
- Vijayan, V.; Saraph, V.; and Milenković, T. 2015. MAGNA++: Maximizing Accuracy in Global Network Alignment via both node and edge conservation. *Bioinformatics*, 31(14): 2409–2411.
- Villani, C. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

- Wang, F.; Xue, N.; Yu, J.-G.; and Xia, G.-S. 2020. Zero-Assignment Constraint for Graph Matching with Outliers. In *CVPR*.
- Wang, F.-D.; Xue, N.; Zhang, Y.; Xia, G.-S.; and Pelillo, M. 2019a. A functional representation for graph matching. *TPAMI*.
- Wang, R.; Yan, J.; and Yang, X. 2019. Learning combinatorial embedding networks for deep graph matching. In *ICCV*.
- Wang, T.; and Ling, H. 2017. Gracker: A graph-based planar object tracker. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1494–1501.
- Wang, Y.; Feng, C.; Chen, L.; Yin, H.; Guo, C.; and Chu, Y. 2019b. User identity linkage across social networks via linked heterogeneous network embedding. *WWW*.
- Wang, Z.; Yang, J.; and Ye, X. 2020. Knowledge graph alignment with entity-pair embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1672–1680.
- Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; Yan, R.; and Zhao, D. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv preprint arXiv:1908.08210*.
- Xiong, H.; Zheng, D.; Zhu, Q.; Wang, B.; and Zheng, Y. F. 2012. A structured learning-based graph matching method for tracking dynamic multiple objects. *IEEE transactions on circuits and systems for video technology*, 23(3): 534–548.
- Xu, H.; Luo, D.; Zha, H.; and Duke, L. C. 2019. Gromov-Wasserstein Learning for Graph Matching and Node Embedding. In *ICML*.
- Yartseva, L.; and Grossglauser, M. 2013. On the performance of percolation graph matching. In *Proceedings of the first ACM conference on Online social networks*, 119–130.
- Zanfir, A.; and Sminchisescu, C. 2018. Deep learning of graph matching. In *CVPR*.
- Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019. Heterogeneous graph neural network. In *KDD*.
- Zhang, C.; Swami, A.; and Chawla, N. V. 2019. Shne: Representation learning for semantic-associated heterogeneous networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 690–698.
- Zhang, M.; Liu, K.; Li, Y.; Guo, S.; Duan, H.; Long, Y.; and Jin, Y. 2021. Unsupervised Domain Adaptation for Person Re-identification via Heterogeneous Graph Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3360–3368.