



Enhancing robust semi-supervised graph alignment via adaptive optimal transport

Songyang Chen¹ · Youfang Lin¹ · Yu Liu¹ · Yuwei Ouyang¹ · Zongshen Guo¹ · Lei Zou²

Received: 24 December 2024 / Revised: 12 February 2025 / Accepted: 18 February 2025 /
Published online: 26 February 2025

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2025

Abstract

The semi-supervised graph alignment problem aims to find the node correspondence across different graphs given a set of anchor links. Most existing methods employ the notion of alignment consistency or embedding-based techniques but overlook the global structure of graph data. Recently, an Optimal Transport (OT)-based method has been proposed for semi-supervised graph alignment by integrating structure-based embedding and OT distance, demonstrating its effectiveness in problem modeling. However, graphs to be aligned often exhibit significant structural differences, and a non-learnable transport cost design struggles to maintain generality when faced with such variations, especially in noisy real-world scenarios. Meanwhile, the challenge of efficiently incorporating anchor links into the cost design has not been thoroughly explored. In this paper, we propose RESAlign, a robust semi-supervised graph alignment framework that addresses the cross-domain alignment problem from both direct and indirect perspectives. By integrating multiple objective functions and an anchor-assisted heterogeneous graph learning module into the design of the transport cost, our framework adapts to structural differences across various graphs. Moreover, an additional weight-sharing mechanism is introduced to address node alignment from a distinct perspective, enabling effective generalization to unsupervised scenarios. Finally, compared to eleven representative methods, the proposed model not only achieves outstanding performance but also demonstrates excellent robustness and efficiency.

✉ Yu Liu
yul@bjtu.edu.cn

Songyang Chen
songyangchen@bjtu.edu.cn

Youfang Lin
yflin@bjtu.edu.cn

Yuwei Ouyang
yuweiouyang@bjtu.edu.cn

Zongshen Guo
zongshenguo@bjtu.edu.cn

Lei Zou
zoulei@pku.edu.cn

¹ Beijing Jiaotong University, Beijing 100044, China

² Peking University, Beijing 100871, China

Keywords Semi-supervised learning · Graph alignment · Optimal transport

1 Introduction

Given attributed graphs from two different sources and a subset of *anchor links* indicating the correspondence between nodes in two graphs, *semi-supervised* graph alignment (also known as network alignment) aims to predict the matching relationships for the remaining nodes. It has a wide range of applications, such as linking the same identity across different social networks [1, 2], matching scholar accounts between multiple academic platforms [3], and aligning tissue-specific protein interaction networks to improve gene prioritization [4].

Solving the exact graph alignment problem is challenging because of its NP-hardness [5]. Therefore, most existing methods have been devoted to matching graphs heuristically. Specifically, consistency-based methods [6–8] are built upon the assumption that if two nodes are aligned, their neighborhoods are likely to be aligned. With a set of anchor links, embedding-based methods [3, 9, 10] aim to preserve the topological information of both networks through node embeddings in a unified latent space, in which anchor node pairs have close embeddings and unmatched node pairs are kept far apart. These methods typically focus on the relationships between positive and sampled negative node pairs, failing to capture the comprehensive structure of graph data [4].

Another line of research [4, 11, 12] treats graphs as probability distributions embedded in a specific metric space, using optimal transport (OT) [13] to model the graph alignment problem. Given a ground cost, conventional optimal transport aims to find the optimal plan that minimizes the expected total cost, the output of which is formulated as a doubly stochastic matrix indicating the pairwise coherency between the distributions. Since graph alignment is inherently a cross-domain alignment problem, there is no predefined cost when modeling this task using optimal transport, thus specific cost design is crucial for alignment accuracy (node pairs that are more easily aligned correspond to lower cost values [14]). However, the recent study [4] adopts a non-learnable cost by performing Random Walk with Restart (RWR) on the product graph to propagate cross-graph information. Although this approach demonstrates good performance, the graphs to be aligned often exhibit significant structural differences (e.g., Douban Online-Offline [8]), a fixed transport cost design in PARROT [4] is inherently limited in its ability to adapt to these diverse scenarios, particularly when dealing with real-world graphs, which are typically noisy [12]. In such cases, fixed transport cost strategies often face serious robustness issues. On the other hand, we observe that its use of anchor links is relatively limited, as it only employs known anchors to derive positional embedding vectors via RWR, failing to simultaneously leverage anchors to integrate both graph structure and node features, thereby capturing richer information. As a result, this leads to suboptimal performance.

Based on these observations, we propose RESAlign, a robust semi-supervised graph alignment framework based on optimal transport. Our method enables fully adaptive cost learning under OT-based problem formation via a *dual-stream framework*. Specifically, to address the cross-domain alignment problem, we use Gromov Wasserstein Discrepancy (GWD) [15], which is directly applicable to cross-domain alignment tasks. By integrating richer graph structure, we enable structural alignment in an indirect way. Furthermore, using known anchor links, we reformulate the cross-domain alignment challenge as a heterogeneous graph learning problem with Wasserstein Distance (WD) [16], facilitating direct node alignment. These two approaches are mutually complementary. To enhance the adapt-

ability of the transport cost and better accommodate the diverse structural characteristics of different graphs, we propose a multi-objective function that integrates anchor links into the learning process of the transport cost, both directly and indirectly. Finally, a weight-sharing mechanism is introduced to enhance the model's generalization ability for node alignment in unsupervised settings. To summarize, our main contributions are as follows:

- We propose RESAlign, a robust semi-supervised graph alignment framework based on optimal transport, which integrates node embeddings that capture rich graph structures and node information to implement a flexible dual-cost design. This innovative design enables RESAlign to effectively adapt to diverse structural differences between graphs. As a result, the framework significantly enhances robustness when handling variations across different graphs.
- We enhance the optimized utilization of anchor node pairs through multi-objective optimization and heterogeneous graph learning, respectively. Furthermore, anchor-assisted heterogeneous graph learning and the weight-sharing mechanism are introduced to address the cross-domain alignment problem from different perspectives, the latter is also applicable to unsupervised scenarios
- Extensive experiments are conducted across six datasets to compare our algorithm with eleven representative methods. Our model consistently outperforms all baselines, achieving an absolute performance improvement of 2 to 10 percentage points in terms of Hits@1 and MRR over the state-of-the-art methods on real-world datasets. It also delivers a 3-4x improvement in efficiency. Moreover, our model is capable of generalizing to unsupervised scenarios, demonstrating outstanding performance.

2 Preliminary

2.1 Problem statement

We denote an undirected and attributed graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where \mathcal{V} is the set of nodes with size n , \mathcal{E} is the set of edges represented by the adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ represents the node features. A summary of frequently used notations is listed in Table 1.

Definition 1 (Semi-Supervised Graph Alignment) Given source graph \mathcal{G}_s and target graph \mathcal{G}_t , without loss of generality, we assume that $n_s = |\mathcal{V}_s|$, $n_t = |\mathcal{V}_t|$, and $n_s \leq n_t$. Given a

Table 1 Table of notations

Notation	Description
$\mathcal{G}_s, \mathcal{G}_t$	The source graph and target graph
$\mathbf{A}_p, \mathbf{X}_p$	Adjacency matrix and node feature matrix ($p = s, t$)
u_i, u_j, v_k, v_l	Nodes with $u_i, u_j \in \mathcal{G}_s$ and $v_k, v_l \in \mathcal{G}_t$
$\mathbf{Z}_s, \mathbf{Z}_t$	Node embeddings for source graph and target graph
$\mathbf{C}_s, \mathbf{C}_t$	The intra-graph cost of source graph and target graph
\mathbf{T}	The alignment matrix
\mathbf{C}_{ot}	The cross-domain cost
\mathcal{S}	Anchor links
\mathcal{G}_h	The merged heterogeneous graph

certain proportion of observed node correspondences (e.g., 20% of nodes), also referred to as the anchor links \mathcal{S} , the semi-supervised graph alignment problem returns an alignment matrix $\mathbf{T} \in \mathbb{R}^{n_s \times n_t}$, where $\mathbf{T}(i, k)$ represents the probability that node $u_i \in \mathcal{G}_s$ is aligned to node $v_k \in \mathcal{G}_t$. Similar to existing works [11, 12, 17], this work primarily focuses on attributed graphs.

2.2 Optimal transport

Due to its favorable properties, optimal transport has gained widespread attention across multiple domains, including machine learning [13], computer vision [18], and data mining [4]. It assumes that two sets of samples (i.e., the node sets), $\mathcal{V}_s = \{u_i\}_{i=1}^{n_s}$ and $\mathcal{V}_t = \{v_k\}_{k=1}^{n_t}$, are generated from the probability distributions $\boldsymbol{\mu} \in \Delta^{n_s-1}$ and $\boldsymbol{\nu} \in \Delta^{n_t-1}$ respectively, where Δ^{n-1} represents the $(n - 1)$ -Simplex. Optimal Transport (OT) and the related Wasserstein distance (WD) [19] are then used to quantify the difference between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ [20], as described by OT’s Kantorovich formulation [19]:

$$W(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{T \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \mathbf{C}, \mathbf{T} \rangle. \tag{1}$$

Here, $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu}) = \left\{ \mathbf{T} \in \mathbb{R}_+^{n_s \times n_t} : \mathbf{T} \mathbf{1}_{n_t} = \boldsymbol{\mu}, \mathbf{T}^\top \mathbf{1}_{n_s} = \boldsymbol{\nu} \right\}$ is the set of joint probability distributions with marginals $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, and it holds that $\sum_{i=1}^{n_s} \sum_{k=1}^{n_t} \mathbf{T}(i, k) = 1$. Note that $\mathbf{1}_n$ denotes the all-ones vector in \mathbb{R}^n , and $\mathbf{C} \in \mathbb{R}^{n_s \times n_t}$ represents the pairwise transport cost across two node sets.

Equation (1) is usually solved with an additional entropic constraint [21]:

$$\min_{T \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \mathbf{C}, \mathbf{T} \rangle - \varepsilon H(\mathbf{T}), \tag{2}$$

where $H(\mathbf{T}) = -\sum_{i,k} \mathbf{T}(i, k) \log \mathbf{T}(i, k)$ denotes the regularized entropy term with coefficient ε . This is not only for the reason of computational efficiency but also from the probabilistic perspective of the OT problem, i.e., it is favorable to have most terms in \mathbf{T} be non-zero [21]. To this end, the OT problem can be solved by the Sinkhorn-Knopp algorithm [22], an iterative matrix scaling process to compute the doubly stochastic matrix \mathbf{T} .

2.3 The state-of-the-art method

As the state-of-the-art method for semi-supervised graph alignment, PARROT [4] proposes the *position-aware regularized optimal transport* framework to enhance the OT-based formation of graph alignment with position-aware node embedding and consistency-based regularization.

For the position-aware transport cost design, it treats anchor nodes as landmarks and captures intra-network topology by performing Random Walk with Restart (RWR) from each anchor node pair in \mathcal{S} on the source and target graphs separately. This yields two embedding matrices \mathbf{R}_s and \mathbf{R}_t of size $n_s \times |\mathcal{S}|$ and $n_t \times |\mathcal{S}|$, respectively. Then, the resulting RWR embeddings are combined with initial features to derive the initial transport cost \mathbf{C}_{node} :

$$\mathbf{C}_{\text{node}} = \alpha e^{-\mathbf{R}_s \mathbf{R}_t^\top} + (1 - \alpha) e^{-\mathbf{X}_s \mathbf{X}_t^\top}, \tag{3}$$

where $\mathbf{R}_s, \mathbf{R}_t$ represent the RWR embedding matrices, $\mathbf{X}_s, \mathbf{X}_t$ represent the feature matrices, and α denotes the weight parameter. To further incorporate cross-graph information, PARROT conducts RWR on the product graph, which is constructed by the Kronecker product of the

source and target graphs. The initial transport cost C_{node} is taken as the personalized vector of the product graph, and the final transport cost C_{rwr} is computed as

$$C_{rwr} = (1 + \beta)C_{node} + (1 - \beta)\gamma W_s C_{rwr} W_t^T, \tag{4}$$

where W_s and W_t are the transpose of the row normalized matrices of A_s and A_t , γ denotes the discounted factor, and β is the teleportation parameter. This step is equivalent to conducting a pairwise random walk of identical lengths simultaneously on the source and target graphs.

Finally, the Sinkhorn algorithm [21, 22] is employed to minimize $\langle C_{rwr}, T \rangle$ to compute the alignment matrix T , with several consistency-based regularization terms. In particular, the edge consistency directly reflects the Gromov-Wasserstein discrepancy, while the neighborhood consistency poses constraints on the smoothness of the predicted alignment probability by comparing to those of the neighbors. The alignment preference is further included to regularize the alignment matrix with known anchors. Generally speaking, PARROT focuses on the direct regularization of the alignment matrix T with topology and anchor information.

3 Methodology

3.1 Overview

Figure 1 depicts the model architecture of our proposed method RESAlign. Given two attributed graphs $\mathcal{G}_s = (\mathcal{V}_s, A_s, X_s)$, $\mathcal{G}_t = (\mathcal{V}_t, A_t, X_t)$ and a set of anchor links \mathcal{S} , our model predicts an alignment probability matrix T of size $|\mathcal{V}_s| \times |\mathcal{V}_t|$. Since the two graphs to be aligned come from different domains, our model leverages optimal transport to address the cross-domain alignment problem from two perspectives. The direct cost learning module conducts cross-graph comparison with the help of anchor links, for example, via the merged graph (two graphs are merged through shared entities to form a single heterogeneous graph) and using heterogeneous graph neural networks. This process can be modeled by the Wasserstein Distance (WD) [19]. Using the Gromov-Wasserstein discrepancy [15], we enable cross-domain comparisons from an indirect perspective. Specifically, this involves

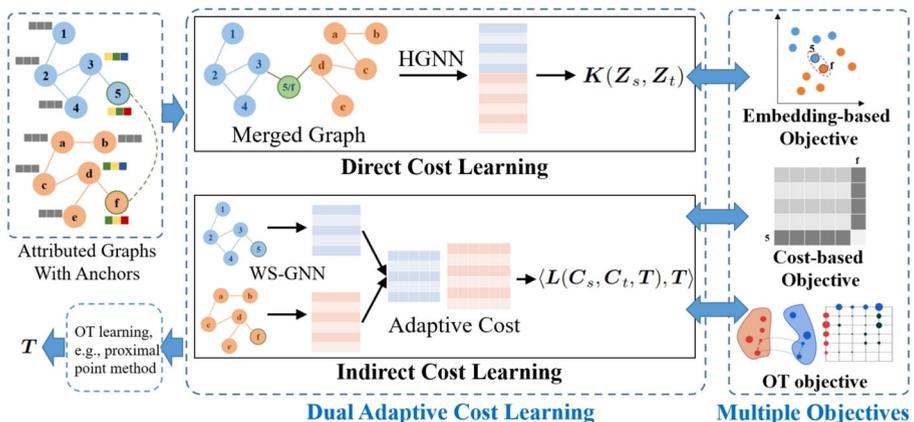


Figure 1 Overview of the RESAlign framework

designing a flexible intra-graph cost by integrating rich graph structures and node features. They can be put together to establish an OT-based objective for graph alignment, by which we compute T . Correspondingly, we construct multi-objective functions that incorporate limited anchor links into the learning process of the optimal transport cost. By more efficiently using anchor links, the flexible cost design better adapts to differences in graph structures, effectively facilitating the alignment process. The following subsections provide a detailed explanation of the model.

3.2 Dual adaptive cost modeling

As pointed out by [12], the critical problem of OT-based graph alignment lies in the design of transport cost. To address this, we propose a dual adaptive cost modeling approach, which includes both direct cost learning and indirect cost learning, enabling multi-view alignment, specifically, node alignment and structural alignment. First, we focus on modeling node alignment using the Wasserstein distance.

Definition 2 (Wasserstein Distance (WD) [19]) We are given two distributions μ and ν on two finite sets \mathcal{X} and \mathcal{Y} , and a cost function $C : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. Kantorovich’s formulation of the optimal transport problem finds a solution T^* such that:

$$T^* = \arg \min_{T \in \Pi(\mu, \nu)} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} C(x, y) T(x, y) = \operatorname{argmin}_{T \in \Pi(\mu, \nu)} \langle C, T \rangle. \tag{5}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the inner product and $\Pi(\mu, \nu)$ denotes all possible joint distributions with the corresponding marginals equal to μ and ν . T denotes the alignment probability matrix, C represents the cross-domain transport cost matrix, where a negative correlation exists between T and C , specifically, nodes that are more easily aligned correspond to smaller values in C [14]. Please refer to Sect. 3.3 for the solution of cross-domain issue in modeling node alignment.

Next, we focus on structural alignment modeling using indirect cost learning. Motivated by [15], the Gromov-Wasserstein discrepancy can be directly applied to cross-domain comparison problems where the focus is put on the relationship of node pairs within each single graph. Specifically, we compute two intra-graph cost matrices $C_s \in \mathbb{R}^{|\mathcal{V}_s| \times |\mathcal{V}_s|}$ and $C_t \in \mathbb{R}^{|\mathcal{V}_t| \times |\mathcal{V}_t|}$ for \mathcal{G}_s and \mathcal{G}_t , which contain the pair-wise measures for all node pairs. The cross-graph comparison is then computed *indirectly* following the Gromov-Wasserstein discrepancy.

Definition 3 (The Gromov-Wasserstein Discrepancy (GWD) [15]) Given the distribution μ (resp. ν) over \mathcal{V}_s (resp. \mathcal{V}_t), the GW discrepancy between μ and ν is defined as:

$$d_{GW}(\mu, \nu) = \min_{T \in \Pi(\mu, \nu)} \langle L(C_s, C_t, T), T \rangle, \tag{6}$$

s.t. $T \mathbf{1}_{|\mathcal{V}_t|} = \mu, T^\top \mathbf{1}_{|\mathcal{V}_s|} = \nu$.

We have $L(C_s, C_t, T)_{i,k} = \sum_j \sum_l |C_s(i, j) - C_t(k, l)|^2 T(j, l)$, where T denotes the alignment probability matrix, and $\mathbf{1}_n$ denotes the all-ones vector in \mathbb{R}^n . For likely matched node pairs (u_i, v_k) and (u_j, v_l) , $|C_s(i, j) - C_t(k, l)|^2$ should be small, i.e., the values of $C_s(i, j)$ and $C_t(k, l)$ are close [11, 12].

Given both the direct and indirect ways to construct the transport cost, a natural question is how to combine them effectively. Since real-world graphs are often noisy and there is no predefined transport cost for cross-domain graph alignment, a non-learnable transport cost

fails to adapt to the diverse graph structures and noise, resulting in a loss of generalization. To address this, we propose a flexible, fully learnable cost function that can accommodate structural differences across graphs, thereby enhancing both generalization and robustness. For the cross-graph node alignment, the fully learnable cost achieves a more flexible integration of graph structure and node features. The cost is computed as:

$$\mathbf{C} = -\mathbf{K}(\mathbf{Z}_s^{wd}, \mathbf{Z}_t^{wd}), \quad (7)$$

where $\mathbf{Z}_s^{wd} \in \mathbb{R}^{|\mathcal{V}_s| \times d}$ and $\mathbf{Z}_t^{wd} \in \mathbb{R}^{|\mathcal{V}_t| \times d}$ are the node representations for two graphs, e.g., computed by GNNs. $\mathbf{K}(\cdot, \cdot)$ is a general function and can be simply implemented as the cosine similarity. The transport cost between nodes should be smaller for more similar nodes [14], thus we directly invert the similarity to represent the cost matrix.¹ For indirect cost modeling, we derive the intra-graph cost terms from the fully learnable node embeddings, which include rich graph structure and node features:

$$\mathbf{C}_s = f(\mathbf{Z}_s^{gwd}), \mathbf{C}_t = f(\mathbf{Z}_t^{gwd}). \quad (8)$$

In particular, $f(\cdot) : \mathbb{R}^{|\mathcal{V}| \times d} \rightarrow \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ denotes the function mapping node embeddings to pairwise metrics, and we set:

$$f(\mathbf{Z}^{gwd}) = \beta^{(1)} \mathbf{A} + \beta^{(2)} \mathbf{K}(\mathbf{Z}^{gwd}, \mathbf{Z}^{gwd}), \quad (9)$$

with $\beta = (\beta^{(1)}, \beta^{(2)})$ being learnable parameters. Recall that by the definition of GWD, for matched node pairs (u_i, v_k) and (u_j, v_l) , $|\mathbf{C}_s(i, j) - \mathbf{C}_t(k, l)|^2$ should be small, i.e., the values of $\mathbf{C}_s(i, j)$ and $\mathbf{C}_t(k, l)$ are close [11]. Since both graphs might not have the same topological structures, the second term essentially uses the node representation-based similarity to construct an augmentation to the adjacency matrix. By making β and \mathbf{Z}^{gwd} fully learnable, we are more capable of holding $\mathbf{C}_s(i, j) \approx \mathbf{C}_t(k, l)$ for matched pairs (u_i, v_k) and (u_j, v_l) . Finally, our dual adaptive cost modeling is formalized as:

$$\mathbf{C}_{ot} = \mathbf{L}(\mathbf{C}_s, \mathbf{C}_t, \mathbf{T}) - \alpha \mathbf{K}(\mathbf{Z}_s^{wd}, \mathbf{Z}_t^{wd}), \quad (10)$$

where α is a predefined hyperparameter. The remaining challenge to obtain high-quality node representations, including \mathbf{Z}^{wd} and \mathbf{Z}^{gwd} , will be elaborated in the following.

3.3 Node representation-based cost learning

Since the graphs to be aligned originate from different domains, they may have distinct topological structures, and their node embeddings reside in separate spaces, making direct comparison infeasible. To address this, we propose two approaches for cross-domain cost learning.

3.3.1 GNN with weight-sharing mechanism

To tackle the cross-domain alignment across graphs, a weight-sharing mechanism (i.e., sharing identical model parameters) is introduced to efficiently map the nodes of different graphs into a unified embedding space. Specifically, we have

$$\mathbf{Z}_p^q = \text{GNN}^q(\mathbf{A}_p, \mathbf{X}_p, \Theta_{WS}^q), \quad (11)$$

¹ Since we have standardized the features, the node embeddings may contain negative values, which can lead to negative values in the similarity matrix. This does not affect the optimal solution as the optimization objective remains to minimize the inner product of the transport cost and the alignment matrix.

for $q \in \{wd, gwd\}$ and $p \in \{s, t\}$. For simplicity, we adopt GCN [23] and aggregate the outputs of all convolution layers to serve as the node embeddings, since each layer learns hidden features at a different range of neighborhood structures and node features:

$$\begin{aligned} \mathbf{Z}_p^{q,k} &= \sigma(\tilde{\mathbf{D}}_p^{-\frac{1}{2}} \tilde{\mathbf{A}}_p \tilde{\mathbf{D}}_p^{-\frac{1}{2}} \cdot \mathbf{Z}_p^{q,k-1} \cdot \mathbf{W}^{q,k}), \\ \mathbf{Z}_p^q &= \sum_{k=1}^K \mathbf{Z}_p^{q,k}, \end{aligned} \tag{12}$$

where $\mathbf{Z}_p^{q,0} = \mathbf{X}_p$, $\Theta_{WS}^q = \{\mathbf{W}^{q,1}, \dots, \mathbf{W}^{q,K}\}$, $\tilde{\mathbf{D}}_p$ denote the diagonal degree matrix, $\tilde{\mathbf{A}}_p = \mathbf{A}_p + \mathbf{I}_{|\mathcal{V}_p| \times |\mathcal{V}_p|}$, and σ denotes the activation function (e.g., ReLU).

3.3.2 Anchor-assisted heterogeneous graph learning

Semi-supervised graph alignment seeks to establish correspondences between nodes in two graphs from different domains, with the help of known anchor links. By connecting an edge between each anchor node pair, which represents the same entity [3] (e.g., the same user in different social networks), the two graphs can be merged. This operation not only enhances the utilization of anchor nodes but also transforms cross-domain cost learning into intra-graph cost learning. The merged graph $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h, \mathbf{X}_h)$ is inherently heterogeneous, the node set of which can be categorized into anchor nodes (denoted as $\mathcal{V}_{s \cap t}$) and non-anchor nodes (denoted as $\mathcal{V}_{s \oplus t} = \mathcal{V}_s \cup \mathcal{V}_t \setminus \mathcal{V}_{s \cap t}$). Correspondingly, we have three types of edges, including edges between anchor nodes (A-A) and between non-anchor nodes (N-N), as well as those bridging non-anchor and anchor nodes (N-A).

To this end, we can directly conduct message-passing between nodes of two original graphs through the anchor nodes, which can be adopted in direct cost learning. Similar to (11) & (12), we can apply any heterogeneous graph neural network (HGNN) to learn the node representation for \mathcal{G}_h , which is denoted as:

$$\mathbf{Z}_h^{wd} = \text{HGNN}(\mathbf{A}_h, \mathbf{X}_h, \Theta_{HG}). \tag{13}$$

In this paper, we choose R-GCN [24] for simplicity and leave the investigation of more sophisticated HGNNs as future work.

Remark It is important to note that while both of the aforementioned strategies can be applied to semi-supervised graph alignment, they exhibit certain biases toward different scenarios. Commonly, existing semi-supervised graph alignment methods [3, 4, 6, 8, 10] assume an anchor node ratio of around 20%. In such scenarios, anchor-assisted heterogeneous graph learning can better leverage the anchors, leading to improved prediction accuracy. However, when only a few anchor nodes are available, or in the absence of anchor nodes, the effectiveness of this approach is reduced. In this scenario, we recommend employing the weight-sharing mechanism, which is applicable to unsupervised settings. To demonstrate the effectiveness of our model, we also conduct a comprehensive comparison with the unsupervised methods.

3.3.3 Making fixed transport cost a special case

Interestingly, we observe that [4] can be considered as a special case of RESAlign, as [4] employs fixed transport costs for the WD and GWD terms. In particular, we have the following theorem.

Theorem 1 *By choosing specific graph neural networks and properly setting the learnable parameters, RESAlign can simulate the transport cost proposed in [4].*

Proof We focus on the WD and the GWD terms used in [4]. Firstly, for the WD term, we demonstrate that the specific form of transport costs in [4], i.e., C_{node} and C_{rwr} , can be simulated by setting the GNN model as GCNII [25]. According to [25], we have

$$\mathbf{H}^{(l+1)} = \sigma \left(\left((1 - a_l) \mathbf{P} \mathbf{H}^{(l)} + a_l \mathbf{H}^{(0)} \right) \left((1 - b_l) \mathbf{I} + b_l \mathbf{W}^{(l)} \right) \right). \tag{14}$$

To simulate C_{node} in [4], we set $\mathbf{P} = \mathbf{A}^\top \mathbf{D}^{-1}$, $a_l = \beta$ and $b_l = 1$ for each layer l . Note that by omitting the non-linear activation and simply letting $\mathbf{W}^{(l)} = \mathbf{I}$, the node representation $\mathbf{H}^{(L)}$ is essentially an estimation of the RWR distribution. To guarantee the RWR process only starts from the anchor nodes, during message passing, each node only aggregates neighborhood information from those neighbors in the anchor node set. This can be implemented by setting the node features of non-anchor nodes to $\mathbf{0}$ during GNN computation. To this end, we can simulate C_{node} by setting $C_{\text{node}} = \langle \mathbf{H}_s^{(L)}, \mathbf{H}_t^{(L)} \rangle$.

For C_{rwr} , please note that it is defined as conducting another RWR-based computation on the product graph $G_s \otimes G_t$ with the personalized vector $\text{vec}(C_{\text{node}})$. By [26] and [27], this process is equivalent to performing a pairwise random walk with restart with the same walk lengths on \mathcal{G}_s and \mathcal{G}_t , respectively. Subsequently, we set $a_l = 0$ and $b_l = 1$ for GCNII, and record $\mathbf{H}_p^{(0)}, \mathbf{H}_p^{(1)}, \dots, \mathbf{H}_p^{(L)}$ separately for $p = s, t$, as in this case $\mathbf{H}_p^{(l)}$ simulates the l -hop walking probability on each single graph. For the RWR process on the product graph, each node pair $(x, y) \in (\mathcal{V}_s, \mathcal{V}_t)$ with $C_{\text{node}}(x, y)$ contributes to the value of $C_{\text{rwr}}(u, v)$, where the weight is determined by the step-wise paired random walk probabilities. To simulate these probabilities, let $\mathbf{H}_{s,t}^{(l)} = \text{Concat}(\mathbf{H}_s^{(l)}, \mathbf{H}_t^{(l)})$, and we can thus simulate C_{rwr} with $C_{\text{rwr}}(u, v) = C_{\text{node}}(u, v) + \sum_{x \in \mathcal{V}_s, y \in \mathcal{V}_t} C_{\text{node}}(x, y) \cdot \sum_{l=0}^L \gamma \beta (1 - \beta)^l \langle \mathbf{H}_{s,t}^{(l)}, \mathbf{H}_{x,y}^{(l)} \rangle$.

Secondly, for the GWD term, which satisfies that $C_s = e^{-X_s X_s^\top} \odot A_s$ and $C_t = e^{-X_t X_t^\top} \odot A_t$, we can adopt an additional learning objective:

$$\min_{\beta, \Theta_{GNN}} \|\beta^{(1)} \mathbf{A} + \beta^{(2)} \mathbf{K}(\mathbf{Z}^{gwd}, \mathbf{Z}^{gwd}) - e^{-X X^\top} \odot \mathbf{A}\|_F, \tag{15}$$

for both graphs. Summing all up, the major cost terms in [4] are simulated via GNN-based transport cost, and the theorem follows. \square

Compared to the fixed cost design based on RWR, GNNs can replicate its functionality under specific parameter settings while offering the advantage of expressing a wider range of functions. This is due to their learnable parameters, inherent non-linearity, and adaptability to various tasks. Since RESAlign has a larger learning space, it is potentially more powerful than the choice of fixed transport cost.

3.4 Multi-objective loss function

To further enhance the adaptivity of using optimal transport for graph alignment, we incorporate anchor links into both node representation learning and transport cost learning. First of all, we employ the OT-based objective in (10), which learn the alignment probability matrix \mathbf{T} and the parameters for cost modeling jointly, denoted as $\Theta = \{\Theta_{WS}, \Theta_{HG}, \beta_s, \beta_t\}$. Next, we propose two additional regularization terms. First, a straightforward objective is to

minimize the distance of node embeddings for anchor links (referred to as the *embedding-based objective*):

$$D(\mathbf{Z}_s, \mathbf{Z}_t, S) = \frac{1}{|S|} \sum_q \sum_{(u_i, v_k) \in S} \|Z_s^q(u_i) - Z_t^q(v_k)\|^2, \tag{16}$$

for $q = \{wd, gwd\}$. As node embeddings are integral to the design of the transport cost, this operation further facilitates the dual adaptive cost learning process. Second, a *cost-based objective* is applied to explicitly constrain the transport cost of the anchor links. We minimize the following regularization term:

$$R(C_{ot}, S) = \frac{1}{|S|} \sum_{(u_i, v_k) \in S} C_{ot}(i, k). \tag{17}$$

To be precise, for each anchor link (u_i, v_k) , the OT-based cost should be small. Compared to [4], we do not pose constraints on the alignment matrix directly but choose to regularize the node embeddings and OT-based cost terms. To sum up, we use the following multi-objective loss:

$$\begin{aligned} \min_{\Theta} \min_{T \in \Pi(\mu, \nu)} & \underbrace{\langle L(C_s, C_t, T), T \rangle}_{\text{Gromov-Wasserstein discrepancy}} - \alpha \underbrace{\langle K(Z_s^{wd}, Z_t^{wd}), T \rangle}_{\text{Wasserstein discrepancy}} \\ & + \lambda \underbrace{D(\mathbf{Z}_s, \mathbf{Z}_t, S)}_{\text{embedding-based objective}} + \gamma \underbrace{R(C_{ot}, S)}_{\text{cost-based objective}} \end{aligned} \tag{18}$$

3.5 Learning algorithm

Equation (18) is a nonconvex optimization problem and can be solved via a nested iterative framework [11, 12] in which the alignment probability matrix T and model parameters Θ are learned alternatively. In the r -th iteration, we update $T^{(r+1)}$ with a proximal term based on the Kullback–Leibler (KL) divergence [28]:

$$\begin{aligned} T^{(r+1)} = \arg \min_T & \langle L(C_s^{(r)}, C_t^{(r)}, T^{(r)}) \\ & - \alpha K(Z_s^{wd, (r)}, Z_t^{wd, (r)}), T \rangle + \varepsilon \text{KL}(T \| T^{(r)}), \end{aligned} \tag{19}$$

where $\text{KL}(T \| T^{(r)}) = \sum_{i,k} T(i, k) \log \frac{T(i,k)}{T^{(r)}(i,k)} - T(i, k) + T(i, k)^{(r)}$ serves as a regularizer of T , whose significance is controlled by the weight hyperparameter ε . This leads to a proximal gradient algorithm and improves the smoothness upon updating T [20]. By fixing the cost terms, this sub-problem is essentially the regularized OT problem and can be solved by the Sinkhorn algorithm [21, 22]. To update $\Theta = \{\Theta_{WS}, \Theta_{HG}, \beta_s, \beta_t\}$, the following optimization objective is established (Cf. (10) & (18)):

$$\min_{\Theta} \mathcal{L}(\Theta) = \langle C_{ot}(T^{(r)}; \Theta) \rangle + \lambda D(\Theta) + \gamma R(T^{(r)}; \Theta). \tag{20}$$

To be specific, given $T^{(r)}$, we update Θ by gradient descent:

$$\begin{aligned} \Theta^{(r+1)} &= \arg \min \left\{ \nabla_{\Theta} \mathcal{L}(\Theta^{(r)})^\top \Theta + \frac{1}{2\tau} \|\Theta - \Theta^{(r)}\|^2 \right\} \\ &= \text{Proj} \left(\Theta^{(r)} - \tau \nabla_{\Theta} \mathcal{L}(\Theta^{(r)}) \right). \end{aligned} \tag{21}$$

The pseudocode of RESAlign is illustrated in Algorithm 1. The following theorem states that by adopting the simplified GNN with one layer of linear transformation, the learning algorithm theoretically guarantees convergence to a critical point, as shown in [11, 12].

Theorem 2 Let $F(\mathbf{T}, \Theta)$ denote the learning objective in (18) with $\Theta = \{\Theta_{WS}, \Theta_{HG}, \beta_s, \beta_t\}$. Denote by $\Theta_{GNN} = (\Theta_{WS}, \Theta_{HG})$ and $\beta = (\beta_s, \beta_t)$. Suppose that $0 < \varepsilon < \frac{1}{L_f}, 0 < \tau < \frac{1}{L_f^\beta}$, and $0 < lr < \frac{1}{L_f^{\Theta_{GNN}}}$, where L_f^T, L_f^β , and $L_f^{\Theta_{GNN}}$ are the gradient Lipschitz continuous modulus of $F(\mathbf{T}, \Theta)$ respectively. If the following conditions hold, i.e.,

$$\mathcal{T} = \{\mathbf{T} \geq 0 : \mathbf{T}\mathbf{1} = \boldsymbol{\mu}, \mathbf{T}^\top \mathbf{1} = \mathbf{v}\}, \tag{22}$$

$$\mathcal{B} = \{(\beta_s, \beta_t) \geq 0 : \sum_{i=1}^2 \beta_p^{(i)} = 1, p = s, t\}, \tag{23}$$

$$\Theta_{GNN} = \{\sum_{i=1} W_{ij} = 1, \forall j \in [1, d]\}, \tag{24}$$

then the GW learning process converges to a critical point of $\bar{F}(\mathbf{T}, \Theta)$, with $\bar{F}(\mathbf{T}, \Theta) = F(\mathbf{T}, \Theta) + \mathbb{I}_{\mathcal{T}}(\mathbf{T}) + \mathbb{I}_{\mathcal{B}}(\beta) + \mathbb{I}_W(\Theta_{GNN})$, where $\mathbb{I}_C(\cdot)$ denotes the indicator function on the set C .

Proof of Theorem 2 Note that by our definition, \mathcal{T}, \mathcal{B} , and Θ_{GNN} are bounded sets, and $F(\mathbf{T}, \Theta)$ is a bi-quadratic function with respect to \mathbf{T}, β , and Θ_{GNN} . To guarantee that Θ_{GNN} satisfies the above constraint in each iteration, it is sufficient to apply an activation function (e.g., ReLU) followed by column-wise normalization for the GNN parameters. The proof then generally follows that of Theorem 5 in [12], where we can compute the gradient of F w.r.t. \mathbf{T}, β , and Θ_{GNN} to update the parameters with provable convergence. \square

Algorithm 1 The pseudocode of RESAlign.

Input: Attributed graphs $\mathcal{G}_s(\mathcal{V}_s, \mathcal{E}_s, \mathbf{X}_s), \mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$, anchor links \mathcal{S} , and hyperparameters $R, \alpha, \lambda, \gamma, \varepsilon$.

Output: The alignment matrix \mathbf{T} ;

- 1 Initialize $\beta_s, \beta_t \leftarrow (1, 1)^\top, \mathbf{T}^{(0)} \leftarrow \frac{1}{|\mathcal{V}_s||\mathcal{V}_t|}$;
 - 2 **for** $r = 0$ to $R - 1$ **do**
 - 3 Compute $\mathbf{Z}_s^{wd}, \mathbf{Z}_t^{wd}$ by (11) & (12) (or 13);
 - 4 Compute $\mathbf{Z}_s^{gwd}, \mathbf{Z}_t^{gwd}$ by (11) & (12);
 - 5 Construct the direct cost by (7);
 - 6 Construct the indirect cost by (8) & (9);
 - 7 Construct the multi-objective loss by (18);
 - 8 Update $\Theta^{(r+1)} \leftarrow \Theta^{(r)}$ by (21);
 - 9 Update $\mathbf{T}^{(r+1)} \leftarrow \mathbf{T}^{(r)}$ by (19);
 - 10 **return** the alignment matrix $\mathbf{T}^{(R)}$;
-

3.6 Complexity analysis

We analyze the complexity of RESAlign in brief. Suppose \mathcal{G}_s has n_s nodes and m_s edges while \mathcal{G}_t has n_t nodes and m_t edges, and d is the feature dimension. Let $n = \max(n_s, n_t)$. The

complexity of intra-graph cost computation for C_s, C_t in (9) is $O(n^2d)$, which is the same for the computational cost of C in (7). According to [11, 12], the GWD term in (10) takes $O(n^3)$ time, thus cost of β -update and T -update in (19) is $O(n^3)$. Since $d \ll \min(n_s, n_t)$, the overall complexity is $O(n^3)$, which is of the same order as other optimal transport-based alignment methods [11, 12, 17].

Note that the bottleneck lies in the Gromov-Wasserstein learning process, which relies on the proximal gradient algorithm [28, 29] to update the alignment matrix. We admit that scalability remains an open problem for the graph alignment problem and is the key focus of our future work.

4 Experimental analysis

4.1 Experimental setup

4.1.1 Datasets

As shown in Table 2, we use commonly employed datasets from the related work [4, 12, 17, 30, 31] for comparison, including three real-world datasets, Douban Online-Offline [8], ACM-DBLP [32], and Allmovie-Imdb [31], as well as three synthetic ones, i.e., Cora [33], Citeseer [33], and PPI [34]. Among these, ACM-DBLP is the largest dataset used in the state-of-the-art methods for both the semi-supervised setting [4] and the unsupervised scenario [12].

Douban Online-Offline [8] In this context, we align two social network graphs of Douban, the online graph and the offline graph. In the online graph, nodes correspond to users, and edges depict interactions between users (such as replies to messages) on the website. The offline graph is constructed based on user co-occurrence in social gatherings, with the location of a user serving as node features in both graphs. The online graph is larger and encompasses all users in the offline graph. For this dataset, 1,118 users appearing in both graphs are utilized as the ground truth.

Table 2 Datasets and their statistics. Ground truth represents the actual matches between the source and target graphs in the dataset

Dataset	$ \mathcal{V}_s , \mathcal{V}_t $	$ \mathcal{E}_s , \mathcal{E}_t $	Features	Ground Truth
Douban Online-Offline	1,118	3,022	538	1,118
	3,906	16,328		
ACM-DBLP	9,872	39,561	17	6,325
	9,916	44,808		
Allmv-Imdb	6,011	124,709	14	5,174
	5,713	119,073		
PPI-1	1,767	37,493	50	1,767
PPI-2	1,767	28,973		
Cora-1	2,708	6,334	1,433	2,708
Cora-2	2,708	4,542		
Citeseer-1	3,327	10,150	3,703	3,327
Citeseer-2	3,327	7,844		

ACM-DBLP [32] The co-author networks, ACM and DBLP, are derived from publication data in four research areas. In both networks, nodes represent authors and edges signify co-author relations. Node features indicate the number of papers published in various venues. There are 6,325 common authors shared between the two networks.

Allmovie-Imdb [31] The Allmovie network is derived from the Rotten Tomatoes website, where two films are connected by an edge if they share at least one common actor. The Imdb network is constructed from the Imdb website using a similar approach. The alignment indicates film identity, incorporating 5,174 anchor links.

Cora [35] This citation network consists of nodes representing publications and edges denoting the citations between them. Cora-1 and Cora-2 are two modified versions of this network, with noise introduced. Specifically, 10% of the edges are added to Cora-1, and 15% are subsequently removed from Cora-2. Both networks are attributed, with node attributes represented by binary feature vectors in a bag-of-words model. Notably, there are 2,708 common publications shared between the two networks.

Citeseer [33] A citation network where nodes correspond to scientific publications and edges represent citation links. Each publication node in the graph is characterized by a one-hot vector for words, indicating the absence/presence of the corresponding word from the dictionary. Similar to the Cora dataset, we construct two new datasets, Citeseer-1 and Citeseer-2, by adding 10% of the edges and removing 15% of the edges from the original dataset, respectively.

PPI [34] It is a protein-protein interaction network where nodes represent proteins and edges represent interactions between them. The node features include motif gene sets and immunological signatures. We construct two new datasets, PPI-1 and PPI-2, in a way similar to Cora and Citeseer.

4.1.2 Baselines

We compare our method with eleven representative baselines, including (1) semi-supervised methods: FINAL [8], NextTAlign [3], WAlign [30], NetTrans [10], BRIGHT [6] and PARROT [4], which is the state-of-the-art model, (2) unsupervised methods: KNN, GAlign [31], GTCAlign [36], GWL [11] and SLOAlign [12]. We omit other existing approaches as they have been outperformed by these baselines. For fairness, in the semi-supervised setting, we use 20% anchor nodes across all methods, while in the unsupervised setting, no anchor nodes are used.

For our method, we provide two model variants, which differ in the implementation of direct cost learning (more specifically, the computation of \mathbf{Z}_s^{wd} and \mathbf{Z}_t^{wd}). In particular, for the semi-supervised setting, the model RESAlign adopts anchor-assisted heterogeneous graph learning via R-GCN [24], while RESAlign-WS employs the weight sharing mechanism with GCN [23]. Note that the latter mainly focuses on the unsupervised setting where anchor links are absent.

4.1.3 Metrics and experimental configuration

Following [12, 30], we adopt Hits@ k with $k = \{1, 5, 10\}$ and Mean Reciprocal Rank (MRR) to evaluate the effectiveness of all models. Given a node $u \in \mathcal{G}_s$, if the aligned node $v \in \mathcal{G}_t$ is among the top- k nodes with the largest alignment probability, it is regarded as a hit.

Table 3 Comparison of model performance in the semi-supervised setting

Datasets	Metrics	FINAL	WAlign	NetTrans	BRIGHT	NeXTAlign	PARROT	RESAlign-WS	RESAlign
Douban Online-Offline	Hits@1	61.54	40.34	47.69	35.19	46.59	70.61	78.87	80.67
	Hits@5	91.68	62.68	61.32	52.29	77.54	93.63	<u>94.32</u>	95.64
	Hits@10	95.43	71.73	66.40	58.65	81.56	96.31	<u>96.54</u>	97.21
	MRR	74.33	51.05	58.91	43.51	58.50	80.59	<u>87.63</u>	89.62
	Time(s)	7.78	13.56	66.64	29.77	62.21	7.79	2.82	2.61
ACM-DBLP	Hits@1	39.72	62.69	70.36	47.88	51.68	72.11	<u>74.38</u>	75.79
	Hits@5	72.79	83.81	90.45	78.41	81.03	92.84	<u>93.11</u>	94.07
	Hits@10	83.34	89.74	94.28	85.63	87.87	96.01	<u>96.33</u>	96.76
	MRR	54.14	71.74	77.04	60.72	63.54	80.56	<u>82.50</u>	83.18
	Time(s)	21.75	238.27	586.54	751.29	1335.23	226.24	59.91	60.46
Allmv-Imdb	Hits@1	83.48	56.52	73.03	69.29	57.05	95.12	<u>97.12</u>	97.25
	Hits@5	95.87	72.58	82.46	80.72	67.56	96.57	<u>97.44</u>	97.92
	Hits@10	96.84	78.02	85.63	84.17	70.53	96.96	<u>97.62</u>	98.24
	MRR	89.06	64.22	77.00	74.52	61.44	95.79	<u>97.42</u>	97.62
	Time(s)	6.74	94.46	828.55	971.05	758.7	55.46	15.76	14.36
Cora	Hits@1	71.02	99.08	98.94	86.47	72.68	99.56	<u>99.60</u>	99.63
	Hits@5	86.25	<u>99.95</u>	99.86	98.84	84.45	100	100	100
	Hits@10	88.14	<u>99.97</u>	99.95	99.40	85.74	100	100	100
	MRR	77.34	99.47	99.35	92.23	77.68	99.71	<u>99.75</u>	99.78
	Time(s)	23.38	24.63	10.85	7.08	16.57	7.94	2.31	2.47
Citeseer	Hits@1	63.11	98.87	95.11	84.67	66.68	99.77	<u>99.79</u>	99.86
	Hits@5	69.08	100	97.10	<u>99.84</u>	80.58	100	100	100
	Hits@10	69.31	100	97.37	<u>99.96</u>	84.79	100	100	100

Table 3 continued

Datasets	Metrics	FINAL	WAlign	NetTrans	BRIGHT	NextAlign	PARROT	RESAlign-WS	RESAlign
PPI	MRR	65.76	99.42	96.09	91.92	72.61	99.86	<u>99.89</u>	99.91
	Time(s)	84.43	39.05	26.85	3.96	11.64	10.23	4.96	3.62
	Hits@1	69.31	88.40	70.08	66.12	62.02	88.24	<u>88.53</u>	88.68
	Hits@5	75.18	91.64	77.93	77.08	78.43	91.32	92.56	<u>92.43</u>
	Hits@10	75.32	93.42	82.24	80.48	82.74	93.51	93.71	<u>93.64</u>
	MRR	71.89	90.16	76.64	71.23	69.02	90.40	<u>90.47</u>	90.54
	Time(s)	2.48	10.65	63.30	3.36	380.63	4.92	2.54	1.62

For a dataset with $|\mathcal{S}^*|$ ground truth anchor links, we have $\text{Hits}@k = \frac{\# \text{ of hits}}{|\mathcal{S}^*|}$. MRR is computed by averaging the inverse of ground truth ranking for each node, i.e., $\text{MRR} = \frac{1}{|\mathcal{S}^*|} \sum_{(u,v) \in \mathcal{S}^*} \frac{1}{\text{rank}(u,v)}$. Note that $\text{Hits}@1$ and MRR directly reveal the accuracy of graph alignment.

All experimental evaluations are performed using a 3090 GPU endowed with 24GB of memory. We adhere to the default configurations of all baselines. To ensure a fair comparison, we reimplement PARROT from its original MATLAB code into Python. In terms of the semi-supervised setting, we follow the default setting of 20% anchor nodes as adopted by the baselines. When varying different anchor ratios, we randomly sample 5%, 25%, 50%, and 75% of the anchor set to obtain the anchor ratio of 1%, 5%, 10%, and 15% for our experimental evaluation. This approach ensures a systematic and rigorous analysis of the impact of anchor node ratios on the results. The source code will be made publicly available upon the acceptance of the paper.

4.2 Comparison of model performance

First, we present the comparison in the semi-supervised setting. The results are shown in Table 3, in which the bold font represents the best results, and the underlined numbers denote the second-best results. We employ both metrics, i.e., $\text{Hits}@k$ and MRR, to evaluate the alignment accuracy, leading to similar qualitative conclusions. The proposed RESAlign and its variant RESAlign-WS outperform all baselines by a notable margin on the real-world datasets, demonstrating the effectiveness of our methods. On the Douban dataset, the RESAlign improves the accuracy by nearly 10 points for both $\text{Hits}@1$ and MRR over PARROT, the state-of-the-art approach. Additionally, we observe a 2-3 percentage point absolute improvement on the ACM-DBLP and Allmv-Imdb datasets. The proposed methods also show certain advantages over the baselines on synthetic datasets. Compared to RESAlign-WS, RESAlign achieves superior performance on real-world datasets, which we attribute to its heterogeneous graph learning module that more effectively leverages anchor nodes.

It can be concluded that OT-based methods consistently surpass embedding-based [6, 10, 30] and consistency-based methods [3, 8], coinciding with the analysis in [4], demonstrating optimal transport as a powerful tool for modeling the graph alignment problem. Moreover, compared to the non-learnable transport cost design [4], we incorporate anchor node information from multiple perspectives into the learnable cost, making the cost function more flexible. When aligning two graphs with significant differences (e.g., Douban), both the direct and indirect cost learning modules can leverage flexible node embeddings to integrate rich graph information, thereby enhancing the model's adaptability.

Meanwhile, Table 4 presents the performance of our model in the unsupervised setting across three real-world datasets. The bold font represents the best results, and the underlined numbers denote the second-best results. Our model consistently outperforms all baseline methods, which can be attributed to our dual approach that simultaneously addresses both node alignment and structural alignment. By integrating richer graph structural information into the design of the transport cost, we enhance the learning of node representations, thereby achieving superior results.

Furthermore, we compare the total cost of the anchor node pairs, as shown in Table 5. Intuitively, anchor node pairs correspond to smaller transport costs. In contrast to the fixed cost design in PARROT, which remains a constant, as our model converges, the cost associated with the anchor node pairs decreases significantly, underscoring the effectiveness of the adaptive cost design in RESAlign.

Table 4 Comparison of model performance in the unsupervised setting across five methods

Datasets	Metrics	KNN	GAlign	GTCAlign	GWL	SLOTAlign	RESAlign-WS
Douban Online-Offline	Hits@1	27.55	45.26	61.79	3.29	<u>51.43</u>	64.58
	Hits@5	42.31	67.71	76.83	8.32	<u>73.43</u>	82.56
	Hits@10	49.28	78.00	82.29	9.93	<u>77.73</u>	85.60
	MRR	35.01	56.32	69.77	5.79	<u>61.29</u>	76.70
ACM-DBLP	Hits@1	36.25	<u>70.20</u>	60.92	56.36	66.04	71.45
	Hits@5	66.83	<u>87.23</u>	75.60	77.09	85.84	89.91
	Hits@10	76.22	<u>91.36</u>	79.97	82.18	87.76	93.15
	MRR	38.59	<u>77.49</u>	67.67	64.82	73.76	79.53
Allmv-Imdb	Hits@1	30.36	82.14	84.73	87.82	<u>90.60</u>	95.47
	Hits@5	47.14	86.35	89.89	92.31	<u>92.75</u>	97.14
	Hits@10	54.25	90.03	91.32	92.83	<u>93.14</u>	97.41
	MRR	38.59	84.96	87.12	89.64	<u>91.61</u>	96.61

4.3 Further analysis of the proposed model

4.3.1 Robustness analysis

To test the robustness of our model, we conduct comparative experiments with PARROT on three real-world datasets, Douban, ACM-DBLP and Allmv-Imdb. We randomly perturb 0-50% of the edges in the target graph. The proposed model significantly surpasses PARROT with different levels of noise, and the performance gap progressively increases with the noise level (Figure 2). Especially for ACM-DBLP and Allmv-Imdb, when the perturbation rate reaches 35%, our model remains an identical performance of PARROT on the original graph. We attribute the robustness of our model to its adaptive cost design. Fixed transport costs are often susceptible to structural differences across various graph datasets, as they fail to adapt to the unique characteristics of different graph structures, leading to a lack of generalization. In contrast, a flexible node embedding approach, augmented by the use of anchor nodes, enables the model to adapt to different graph noises, thereby enhancing stability and robustness.

4.3.2 Running time and convergence analysis

We report the runtime when the model converges, as shown in Table 3. The OT-based methods, PARROT and RESAlign, achieve shorter runtime while maintaining higher accuracy. Compared to PARROT, the proposed RESAlign achieves a 3-4x improvement in efficiency.

Table 5 The total cost of anchor node pairs across three real-world datasets, with smaller values being preferable

Datasets	Douban Online-Offline	ACM-DBLP	Allmv-Imdb
PARROT	84.21	469.42	382.03
RESAlign	0.38	1.22	3.24

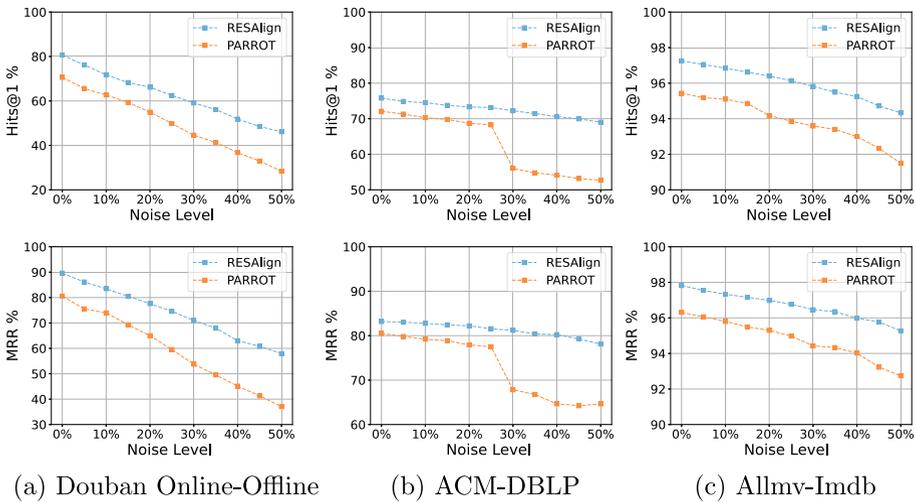


Figure 2 Prediction accuracy vs. noise level

This is primarily attributed to the strong expressive power of our method, which allows the model to converge with few epochs (in Figure 3), and the fact that the matrix multiplication in Gromov-Wasserstein learning can fully leverage the computing power of GPUs. However, the efficiency of PARROT is severely constrained by the computationally intensive Random Walk with Restart (RWR) process on the product graph. In contrast, while FINAL achieves shorter runtime on some datasets, it demonstrates suboptimal performance.

We evaluate the convergence of the model on three real-world datasets. As shown in Figure 3, with more epochs, the prediction accuracy gradually increases along with the continuously decreasing of the loss, and the model finally converges.

4.3.3 Impact of anchor ratio

As illustrated in Figure 4, we present a comparative visualization of the performance (MRR) between the proposed RESAlign and the state-of-the-art method, PARROT, across different anchor ratios. It is evident that the performance of both models improves progressively as the anchor ratio increases. Notably, even at a low anchor ratio of 5%, RESAlign demonstrates

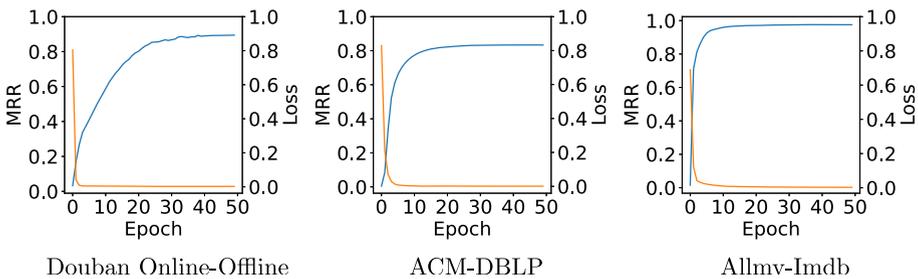


Figure 3 Convergence analysis on three real-world datasets

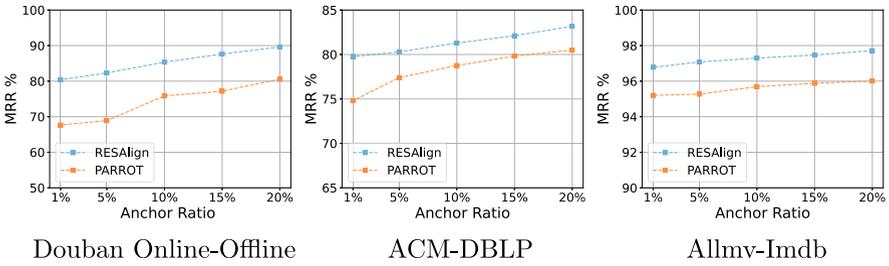


Figure 4 MRR vs. anchor ratio

competitive performance compared to PARROT, validating the effectiveness of RESAlign in scenarios with limited anchor nodes.

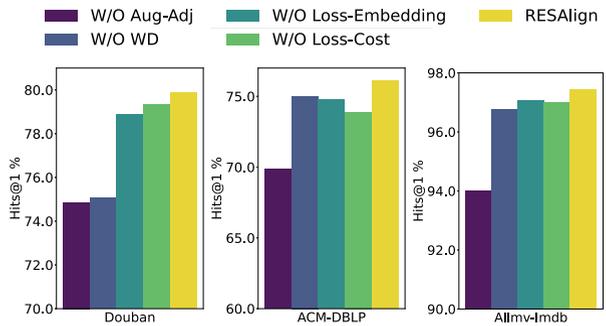
4.3.4 Ablation study

To validate the effectiveness of each component, we conduct the ablation study in Figure 5. In particular, W/O Aug-Adj represents RESAlign without the augmented graph structure for indirect cost modeling (9), while W/O WD denotes the model without the anchor-based direct cost learning module (13). W/O Loss-Embedding and W/O Loss-Cost represent the model without the embedding-based objective (16) and the cost-based objective (17), respectively. Each module contributes to the overall performance. It can also be observed that the augmented graph structure in GWD plays a crucial role. We attribute this to the inherent suitability of GWD for cross-domain alignment tasks, as it primarily models structural alignment, with the augmented graph structure further enhancing its effectiveness.

4.3.5 Sensitivity analysis

We further analyze the model’s performance across different feature dimensions and values of α and ϵ , as shown in Figure 6, with α representing the weight hyperparameter for the WD term, and ϵ being the coefficient for entropic regularization of OT. Overall, the model achieves consistently high performance, with the optimal configuration found at a feature dimension of 128, $\alpha = 0.2$, and $\epsilon = 0.03$.

Figure 5 Ablation study on three real-world datasets



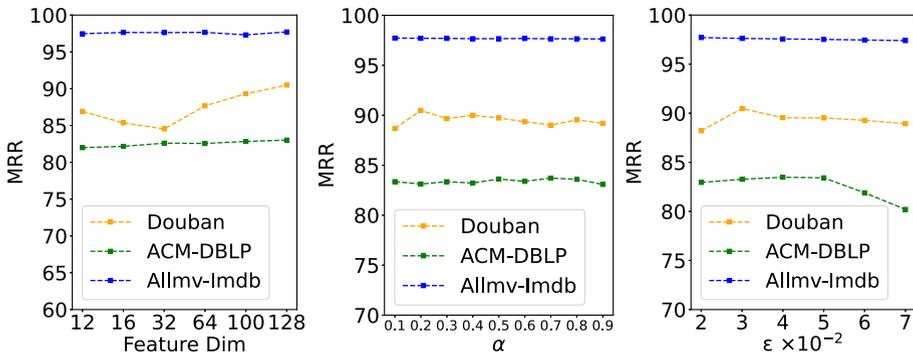


Figure 6 Impact of hyperparameters

5 Related work

5.1 Graph alignment

Existing *semi-supervised* methods can be roughly categorized into consistency-based, embedding-based, and optimal transport (OT)-based solutions. Consistency-based approaches [8, 37] usually assume the existence of a noisy permutation between the aligned graphs and put emphasis on local topology and attribute consistency. Embedding-based methods compute low-dimensional node embeddings through matrix factorization [38], Random Walk with Restart (RWR) [6, 39], and adversarial learning [40, 41], ensuring anchor node pairs to have close embeddings. A recently proposed method [4] formulates graph alignment as an optimal transport problem and devises the transport cost inspired by RWR for the alignment process. The typical setting is to exploit 20 percent of node pairs as anchor links [4]. Another line of existing studies focuses on *unsupervised* graph alignment without anchor node pairs. Well-adopted techniques include graph augmentation [31, 36], adversarial learning with GNN [30], and OT-based graph alignment [11, 12, 42, 43], which generally relies on hand-crafted cost design. Besides, the entity alignment problem has also been extensively studied for knowledge graphs [44–50], e.g., for multimodal entity-relation extraction [51].

5.2 Optimal transport

Optimal Transport (OT) is used to compare two probability distributions with a convex linear program. Given the cost that measures the distance between two distributions, it can be interpreted as moving the mass from the source distribution to the target with minimal expected total cost. In particular, the Gromov-Wasserstein distance [15] extends OT to ground spaces that are not pre-registered, which leads to a non-convex quadratic program for transport computation. OT-based distances find their applications in matching tasks, including shape matching [52] and multi-modal alignment [48, 53]. A bunch of OT-based methods for graph matching has been proposed [11, 12, 54, 55], with most of them under the unsupervised scenario.

6 Conclusion and future work

We present RESAlign, a robust semi-supervised graph alignment framework that effectively addresses cross-domain graph alignment through a dual-perspective approach with optimal transport. The RESAlign employs multi-objective loss functions and anchor-assisted heterogeneous graph learning modules, efficiently integrating anchor links into the flexible design of the transport cost. These innovations enable the model to adapt to varying graph structural differences, ensuring strong robustness. Moreover, a weight-sharing mechanism is used to map node embeddings into a unified space, facilitating compatibility with unsupervised scenarios. Finally, extensive experiments validate the effectiveness of RESAlign.

We also point out the limitations of the proposed RESAlign method. Although it demonstrates outstanding performance in modeling graph alignment using optimal transport across a wide range of datasets, the high computational complexity associated with using GWD for structural alignment presents scalability challenges. As a result, improving scalability will be a central focus of our future work, for example, by sparsifying the computation of GWD [20]. Furthermore, in semi-supervised scenarios, the selection strategy of anchor nodes and its impact on alignment accuracy represent a promising direction for future research.

Author Contributions Songyang Chen and Yu Liu proposed the method and completed the manuscript writing. Yuwei Ouyang and Zongshen Guo assisted with some of the experiments, while Youfang Lin and Lei Zou provided valuable suggestions for revising the paper.

Funding This work was supported by National Natural Science Foundation of China under grant 62202037, 62372031, 61932001, and CCF-Tencent Open Fund under grant RAGR20220131.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Ethical Approval Not applicable.

Conflict of interest The authors declare no Conflict of interest.

References

1. Zhang, J., Philip, S.Y.: Multiple anonymized social networks alignment. In: 2015 IEEE International Conference on Data Mining, pp. 599–608 (2015). IEEE
2. Li, C., Wang, S., Wang, H., Liang, Y., Yu, P.S., Li, Z., Wang, W.: 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, pp. 249–258 (2019)
3. Zhang, S., Tong, H., Jin, L., Xia, Y., Guo, Y.: Balancing consistency and disparity in network alignment. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2212–2222 (2021)
4. Zeng, Z., Zhang, S., Xia, Y., Tong, H.: Parrot: Position-aware regularized optimal transport for network alignment. In: Proceedings of the ACM Web Conference 2023, pp. 372–382 (2023)
5. Cheng, H., Luo, D., Xu, H.: Dhot-gm: Robust graph matching using a differentiable hierarchical optimal transport framework. (2023) [arXiv:2310.12081](https://arxiv.org/abs/2310.12081)
6. Yan, Y., Zhang, S., Tong, H.: Bright: A bridging algorithm for network alignment. In: Proceedings of the Web Conference 2021, pp. 3907–3917 (2021)
7. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceed. National Acad. Sci.* **105**(35), 12763–12768 (2008)

8. Zhang, S., Tong, H.: Final: Fast attributed network alignment. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1345–1354 (2016)
9. Chu, X., Fan, X., Yao, D., Zhu, Z., Huang, J., Bi, J.: Cross-network embedding for multi-network alignment. In: The World Wide Web Conference, pp. 273–284 (2019)
10. Zhang, S., Tong, H., Xia, Y., Xiong, L., Xu, J.: Nettrans: Neural cross-network transformation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 986–996 (2020)
11. Xu, H., Luo, D., Zha, H., Duke, L.C.: Gromov-wasserstein learning for graph matching and node embedding. In: International Conference on Machine Learning, pp. 6932–6941 (2019). PMLR
12. Tang, J., Zhang, W., Li, J., Zhao, K., Tsung, F., Li, J.: Robust attributed graph alignment via joint structure learning and optimal transport. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE), pp. 1638–1651 (2023). IEEE
13. Peyré, G., Cuturi, M., et al.: Computational optimal transport: With applications to data science. *Found. Trends® Mach. Learn.* **11**(5-6), 355–607 (2019)
14. Peyré, G., Cuturi, M., et al.: Computational optimal transport: With applications to data science. *Foundat. Trends® Mach. Learn.* **11**(5-6), 355–607 (2019)
15. Peyré, G., Cuturi, M., Solomon, J.: Gromov-wasserstein averaging of kernel and distance matrices. In: International Conference on Machine Learning, pp. 2664–2672 (2016). PMLR
16. Kantorovich, L.V.: On the translocation of masses. *J. Math. Sci.* **133**(4) (2006)
17. Cheng, H., Luo, D., Xu, H.: Dhot-gm: Robust graph matching using a differentiable hierarchical optimal transport framework. (2023) [arXiv:2310.12081](https://arxiv.org/abs/2310.12081)
18. Xu, L., Sun, H., Liu, Y.: Learning with batch-wise optimal transport loss for 3d shape recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3333–3342 (2019)
19. Kantorovich, L.: On the transfer of masses (in russian). In: *Doklady Akademii Nauk*, **37**, p. 227 (1942)
20. Li, M., Yu, J., Xu, H., Meng, C.: Efficient approximation of gromov-wasserstein distance using importance sparsification. *J. Comput. Graph. Stat.* **32**(4), 1512–1523 (2023)
21. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. *Adv. Neural. Inf. Process. Syst.* **26** (2013)
22. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.* **21**(2), 343–348 (1967)
23. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. (2016) [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
24. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June, 2018, Proceedings 15, pp. 593–607 (2018). Springer
25. Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks. In: International Conference on Machine Learning, pp. 1725–1735 (2020). PMLR
26. Jeh, G., Widom, J.: Simrank: A measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543 (2002)
27. Liu, H., Liao, N., Luo, S.: Simga: A simple and effective heterophilous graph neural network with efficient global aggregation. (2023) [arXiv:2305.09958](https://arxiv.org/abs/2305.09958)
28. Xie, Y., Wang, X., Wang, R., Zha, H.: A fast proximal point method for computing exact wasserstein distance. In: Uncertainty in Artificial Intelligence, pp. 433–453 (2020). PMLR
29. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.* **146**(1), 459–494 (2014)
30. Gao, J., Huang, X., Li, J.: Unsupervised graph alignment with wasserstein distance discriminator. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 426–435 (2021)
31. Trung, H.T., Van Vinh, T., Tam, N.T., Yin, H., Weidlich, M., Hung, N.Q.V.: Adaptive network alignment with unsupervised and multi-order convolutional networks. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 85–96 (2020). IEEE
32. Zhang, S., Tong, H.: Attributed network alignment: Problem definitions and fast solutions. *IEEE Trans. Knowl. Data. Eng.* **31**(9), 1680–1692 (2018)
33. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. *AI Magaz.* **29**(3), 93–93 (2008)
34. Zitnik, M., Leskovec, J.: Predicting multicellular function through multi-layer tissue networks. *Bioinf.* **33**(14), 190–198 (2017)
35. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: International Conference on Machine Learning, pp. 40–48 (2016). PMLR

36. Wang, C., Jiang, P., Zhang, X., Wang, P., Qin, T., Guan, X.: Gtcalign: Global topology consistency-based graph alignment. *IEEE Trans. Knowl. Data. Eng.* (2023)
37. Zhang, J., Philip, S.Y.: Integrated anchor and social link predictions across social networks. In: *Twenty-fourth International Joint Conference on Artificial Intelligence* (2015)
38. Heimann, M., Shen, H., Safavi, T., Koutra, D.: Regal: Representation learning-based graph alignment. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 117–126 (2018)
39. Yan, Y., Zhou, Q., Li, J., Abdelzaher, T., Tong, H.: Dissecting cross-layer dependency inference on multi-layered inter-dependent networks. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2341–2351 (2022)
40. Hong, H., Li, X., Pan, Y., Tsang, I.W.: Domain-adversarial network alignment. *IEEE Trans. Knowl. Data. Eng.* **34**(7), 3211–3224 (2020)
41. Zhou, Y., Zhang, Z., Wu, S., Sheng, V., Han, X., Zhang, Z., Jin, R.: Robust network alignment via attack signal scaling and adversarial perturbation elimination. In: *Proceedings of the Web Conference 2021*, pp. 3884–3895 (2021)
42. Titouan, V., Courty, N., Tavenard, R., Flamary, R.: Optimal transport for structured data with application on graphs. In: *International Conference on Machine Learning*, pp. 6275–6284 (2019). PMLR
43. Chen, S., Liu, Y., Zou, L., Wang, Z., Lin, Y., Chen, Y., Pan, A.: Combining optimal transport and embedding-based approaches for more expressiveness in unsupervised graph alignment. (2024) [arXiv:2406.13216](https://arxiv.org/abs/2406.13216)
44. Zeng, K., Li, C., Hou, L., Li, J., Feng, L.: A comprehensive survey of entity alignment for knowledge graphs. *AI Open* **2**, 1–13 (2021)
45. Liu, X., Hong, H., Wang, X., Chen, Z., Kharlamov, E., Dong, Y., Tang, J.: Selfkg: Self-supervised entity alignment in knowledge graphs. In: *Proceedings of the ACM Web Conference 2022*, pp. 860–870 (2022)
46. Zeng, K., Dong, Z., Hou, L., Cao, Y., Hu, M., Yu, J., Lv, X., Cao, L., Wang, X., Liu, H., et al.: Interactive contrastive learning for self-supervised entity alignment. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2465–2475 (2022)
47. Xin, K., Sun, Z., Hua, W., Liu, B., Hu, W., Qu, J., Zhou, X.: Ensemble semi-supervised entity alignment via cycle-teaching. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**, pp. 4281–4289 (2022)
48. Wang, L., Qi, P., Bao, X., Zhou, C., Qin, B.: Pseudo-label calibration semi-supervised multi-modal entity alignment. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, pp. 9116–9124 (2024)
49. Li, Y., Chen, L., Liu, C., Zhou, R., Li, J.: Generative adversarial network for unsupervised multi-lingual knowledge graph entity alignment. *World Wide Web* **26**(5), 2265–2290 (2023)
50. Shu, Y., Zhang, J., Huang, G., Chi, C.-H., He, J.: Entity alignment via graph neural networks: a component-level study. *World Wide Web* **26**(6), 4069–4092 (2023)
51. Yuan, L., Cai, Y., Wang, J., Li, Q.: Joint multimodal entity-relation extraction based on edge-enhanced graph alignment network and word-pair relation tagging. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, pp. 11051–11059 (2023)
52. Solomon, J., Peyré, G., Kim, V.G., Sra, S.: Entropic metric alignment for correspondence problems. *ACM Trans. Graph. (ToG)* **35**(4), 1–13 (2016)
53. Chen, L., Gan, Z., Cheng, Y., Li, L., Carin, L., Liu, J.: Graph optimal transport for cross-domain alignment. In: *International Conference on Machine Learning*, pp. 1542–1553 (2020). PMLR
54. Maretic, H.P., El Gheche, M., Chierchia, G., Frossard, P.: Fgot: Graph distances based on filters and optimal transport. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**, pp. 7710–7718 (2022)
55. Xu, H., Luo, D., Carin, L.: Scalable gromov-wasserstein learning for graph partitioning and matching. *Adv. Neural. Inf. Process. Syst.* **32** (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.