PARROT: Position-Aware Regularized Optimal Transport for Network Alignment

Zhichen Zeng
zhichenz@illinois.edu
University of Illinois at Urbana-Champaign
IL, USA

Si Zhang
sizhang@meta.com
Meta
CA, USA

Yinglong Xia
yxia@meta.com
Meta
CA, USA

Hanghang Tong
htong@illinois.edu
University of Illinois at Urbana-Champaign
IL, USA

ABSTRACT

Network alignment is a critical steppingstone behind a variety of multi-network mining tasks. Most of the existing methods essentially optimize a Frobenius-like distance or ranking-based loss, ignoring the underlying geometry of graph data. Optimal transport (OT), together with Wasserstein distance, has emerged to be a powerful approach accounting for the underlying geometry explicitly. Promising as it might be, the state-of-the-art OT-based alignment methods suffer from two fundamental limitations, including (1) *effectiveness* due to the insufficient use of topology and consistency information and (2) *scalability* due to the non-convex formulation and repeated computationally costly loss calculation. In this paper, we propose a position-aware regularized optimal transport framework for network alignment named PARROT. To tackle the effectiveness issue, the proposed PARROT captures topology information by random walk with restart, with three carefully designed consistency regularization terms. To tackle the scalability issue, the regularized OT problem is decomposed into a series of convex subproblems and can be efficiently solved by the proposed constrained proximal point method with guaranteed convergence. Extensive experiments show that our algorithm achieves significant improvements in both effectiveness and scalability, outperforming the state-of-the-art network alignment methods and speeding up existing OT-based methods by up to 100 times.

KEYWORDS

Network alignment, optimal transport, alignment consistency

ACM Reference Format:
Zhichen Zeng, Si Zhang, Yinglong Xia, and Hanghang Tong. 2023. PARROT: Position-Aware Regularized Optimal Transport for Network Alignment. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583357

1 INTRODUCTION

Mining multiple networks collected from different resources is an important task in many applications, including social network analysis, e-commerce recommendation, computer vision, and financial fraud detection [8]. A fundamental step of multi-network mining is to find the node correspondence across different networks, also known as the network alignment problem. For example, aligning users across different social networks helps improve the quality of online recommender systems [1]. Integrating knowledge graphs, such as Wikipedia and WorkNet, is essential to construct a coherent knowledge base [3]. In bioinformatics, aligning tissue-specific protein interaction networks improves gene prioritization [18].

Most existing methods, including consistency and embedding-based methods, essentially optimize a Frobenius-like distance or ranking-based loss. To be specific, consistency-based methods are built upon the linearity and/or consistency assumptions [44]. Most consistency-based methods [11, 26, 39, 40, 42], explicit or implicitly, assume a linear transformation between two networks and aim to minimize the Frobenius distance between the adjacency matrices of one network and the permutation of another [45]. However, it has been pointed out that the Frobenius distance may overlook the underlying geometry of graph data [14]. Embedding-based methods, on the other hand, aim to find two nonlinear projectors that project two networks into a unified low-dimensional space where positive node pairs are close while negative node pairs are far away from each other, often enforced by ranking-based loss functions [5, 21, 41, 44]. Nevertheless, the ranking-based loss only addresses the relationship of positive and sampled negative node pairs but ignores relationships between the rest node pairs, and hence fails to capture the holistic structure of graph data.

More recently, optimal transport (OT), together with Wasserstein distance (WD), has emerged to be a powerful approach addressing the underlying structure of two distributions [22]. Given a transport cost measuring the distance between all sample pairs in two distributions, OT seeks for the best coupling, or alignment, between two distributions minimizing the expected cost, hence successfully depicting the holistic structure of two distributions. Besides, the OT framework can filter out noises in the cost measure and provide more deterministic alignment results [16, 27]. Thus, OT provides a potentially better approach to benefit the network alignment task.

The existing OT-based network alignment methods can be categorized into either continuous or discrete approaches. Continuous

WWW '23, April 30-May 4, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583357

approaches [13–15] represent each graph by a multivariate Gaussian distribution with a graph Laplacian-like covariance matrix, while discrete approaches [2, 20, 31] associate each graph with a discrete uniform distribution over its node set. However, most, if not all, of existing OT-based methods suffer from effectiveness and/or scalability limitations. First (effectiveness), the design of transport cost is crucial to the OT problem, however, most existing OT-based methods [2, 25, 31] only embrace node attributes or local graph structure for cost design. Besides, the cost measure only depicts node relationships, while edge and neighborhood relationships are largely ignored. Second (scalability), although continuous approaches encode graph structure by the graph Laplacian-like covariance matrix [13–15], the resulting problems are non-convex, inevitably involving gradient descent for optimization with repeated computationally costly loss calculation.

In this paper, we propose a novel algorithm named PARROT to address the above limitations. For the effectiveness limitation, random walk with restart (RWR) is performed on separated and product graphs to encode graph topology for a position-aware transport cost. To incorporate alignment consistency, we neatly design the regularization terms for *edge consistency*, *neighborhood consistency*, and *alignment preference*. For the scalability limitation, we formulate the network alignment problem as a regularized OT problem and propose a constrained proximal point method for a fast solution, which, as we theoretically prove, has guaranteed convergence.

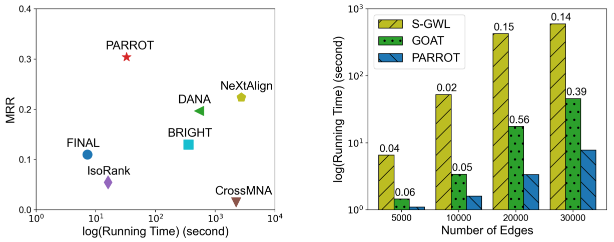
The main contributions of this paper are summarized as follows:

- **Problem Formulation.** We formulate the network alignment problem as an OT problem with alignment consistency regularization at multiple levels.
- **Algorithm and Analysis.** We propose a fast and scalable algorithm named PARROT for network alignment, which, as we theoretically prove, has guaranteed convergence.
- **Experimental Results.** We perform extensive experiments on both plain and attributed networks, and the results show that our method achieves up to 13% improvement on plain networks and 3% improvement on attributed networks in terms of MRR compared with the best competitor. Besides, PARROT is up to 100×+ faster than existing OT-based methods. See Figure 1 for comparison.

The rest of the paper is organized as follows. Section 2 defines the OT problem in the context of network alignment and introduces the preliminaries. Section 3 formulates the optimization problem. Section 4 presents the proposed PARROT algorithm and relevant analyses. Section 5 shows the experiment results. Related works and conclusions are given in Sections 6 and 7 respectively.

2 PROBLEM DEFINITION

Table 1 summarizes the main symbols and notations used throughout the paper. We use bold uppercase letters for matrices (e.g., $\mathbf{A}$), bold lowercase letters for vectors (e.g., $\mathbf{s}$), and lowercase letters for scalars (e.g., α). The element at the i-th row and j-th column of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}(i,j)$. The transpose of $\mathbf{A}$ is denoted by the superscript T (e.g., $\mathbf{A}^\mathsf{T}$). An attributed network is represented by a triplet $\mathcal{G} = \{\mathcal{V}, \mathbf{A}, \mathbf{X}\}$ where $\mathcal{V}, \mathbf{A}, \mathbf{X}$ denote the node set, adjacency matrix, and node attribute matrix respectively. Given two attributed networks $\mathcal{G}_1 = \{\mathcal{V}_1, \mathbf{A}_1, \mathbf{X}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}_2, \mathbf{A}_2, \mathbf{X}_2\}$, and a set of anchor node pairs $\mathcal{L}$ indicating which nodes are aligned a priori, the semi-supervised attributed network alignment task aims to find the best alignment matrix $\mathbf{S} \in \mathbb{R}^{n_1 \times n_2}$, where $\mathbf{S}(x,y)$ indicates how likely node x in $\mathcal{V}_1$ and node y in $\mathcal{V}_2$ are aligned.



(a) (b)

Figure 1: Mean Reciprocal Rank (MRR) and running time of different methods. (a) Comparison with consistency and embedding-based methods; (b) Comparison with OT-based methods: numbers above bars indicate the ratio of MRR of baselines and that of PARROT. Our proposed PARROT (1) achieves much higher MRR than consistency-based methods (FINAL and IsoRank) with a comparable running time and (2) outperforms all embedding and OT-based methods in terms of both MRR and speed.

Table 1: Symbols and Notations.

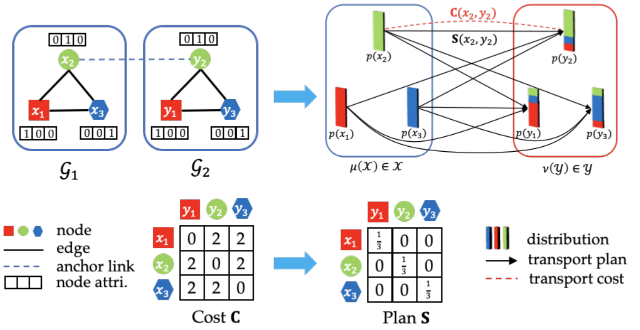
Symbol	Definition
$\mathcal{G}_1, \mathcal{G}_2$	input networks
$\mathcal{V}_1, \mathcal{V}_2$	the sets of nodes of $\mathcal{G}_1$ and $\mathcal{G}_2$
$\mathbf{A}_1, \mathbf{A}_2$	adjacency matrices of $\mathcal{G}_1$ and $\mathcal{G}_2$
$\mathbf{X}_1, \mathbf{X}_2$	node attribute matrices of $\mathcal{G}_1$ and $\mathcal{G}_2$
n_i, m_i	number of nodes/edges in $\mathcal{G}_i$
$\mathcal{L}$	the set of anchor node pairs
$\mathbf{I}, \mathbf{1}$	an identity matrix and a column vector of all 1s
$\mathbf{s} = \text{vec}(\mathbf{S})$	vectorization of matrix S in the column order
$\mathbf{D} = \text{diag}(\mathbf{d})$	diagonal matrix of a vector $\mathbf{d}$
$\otimes$	Kronecker product
$\odot$	Hadamard product
$\langle \cdot, \cdot \rangle$	inner product
Π	probabilistic coupling
$[\cdot \Vert \cdot]$	horizontal concatenation of vectors

2.1 Optimal Transport

OT has recently been revisited in various domains including image processing, data mining and machine learning [22]. We follow the Kantorovich formulation which can be formally defined in terms of two distributions and a cost matrix as follows:

Definition 1. *Optimal Transport and Wasserstein distance [17]. Given two discrete distributions $\boldsymbol{\mu}, \boldsymbol{\nu}$ defined on probability simplex Δ_1, Δ_2 and a cost matrix $\mathbf{C} \in \mathbb{R}^{n_1 \times n_2}$ measuring the distance between all pairs $(x_i, y_j) \in \Delta_1 \times \Delta_2$ across two distributions. The OT problem seeks for an optimal coupling/transport plan $\mathbf{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$ between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ that minimizes the expected cost over the coupling as follows*

$$\mathbf{S} = \underset{\mathbf{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})}{\arg\min} \sum_{x_i, y_j} \mathbf{C}(x_i, y_j)\mathbf{S}(x_i, y_j) = \underset{\mathbf{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})}{\arg\min} \langle \mathbf{C}, \mathbf{S} \rangle \tag{1}$$



	y_1	y_2	y_3
x_1	0	2	2
x_2	2	0	2
x_3	2	2	0

Cost **C**

	y_1	y_2	y_3
x_1	$\frac{1}{3}$	0	0
x_2	0	$\frac{1}{3}$	0
x_3	0	0	$\frac{1}{3}$

Plan **S**

Figure 2: An example of OT-based network alignment: $\mathcal{G}_1, \mathcal{G}_2$ are represented as uniform distributions $\mu = \frac{\mathbf{1}_{n_1}}{n_1}, \nu = \frac{\mathbf{1}_{n_2}}{n_2}$ on node sets. Bars in right boxes indicate the probability p of data points x_i and y_j in the distribution. We may calculate the cost matrix C and the transport plan S based on L_1 norm.

where S *is the optimal transport plan and corresponding* $\langle \mathrm{C}, \mathrm{S} \rangle$ *is the Wasserstein distance between* $\boldsymbol{\mu}$ *and* $\boldsymbol{\nu}$*.*

Remark. To adopt the OT framework for network alignment, a key question is how to represent networks as distributions. Figure 2 gives an example of the OT-based network alignment. Specifically, by considering nodes as samples of a distribution, the input networks $\mathcal{G}_1, \mathcal{G}_2$ can be represented as discrete uniform distributions $\boldsymbol{\mu} = \frac{\mathbf{1}_{n_1}}{n_1}, \boldsymbol{\nu} = \frac{\mathbf{1}_{n_2}}{n_2}$ over their corresponding node sets [2]. Then, together with the transport cost $\mathbf{C} \in \mathbb{R}^{n_1 \times n_2}$, the transport plan S obtained by solving Eq. (1) indicates the node alignments.

2.2 Position-aware Node Embedding

Several recent works address the necessity of capturing node's positional information within a network to generate position-aware node embeddings. P-GNN [37] encodes positional information by the distance between the target node and a set of landmarks, but suffers from the space disparity issue due to the different landmark selections of different graphs. BRIGHT [34] performs RWR on separated graphs and generates a unified RWR embedding space with the help of anchor node pairs. By regarding anchor node pairs as one identical landmark in the RWR embedding space, RWRs on $\mathcal{G}_1$ and $\mathcal{G}_2$ encode the positional information w.r.t. same landmarks, and hence construct a unified RWR embedding space.

3 OPTIMIZATION FORMULATION

In this section, we present our regularized OT-based network alignment formulation. To leverage the topology information, we first design a position-aware transport cost in Section 3.1. Then we introduce three regularization terms to incorporate the alignment consistency principle into the OT framework to form a regularized OT-based network alignment formulation in Section 3.2.

3.1 Position-aware Transport Cost

The effectiveness of OT-based methods largely relies on the quality of the transport cost. To fully exploit multi-aspect information underlying multiple networks, our key idea is to exploit RWR on separated networks capturing intra-network topology information, as well as RWR on the product graph depicting cross-network node pair relationships. Together with node attributes, we obtain a position-aware transport cost that simultaneously encodes both structural and attribute information of multiple networks.

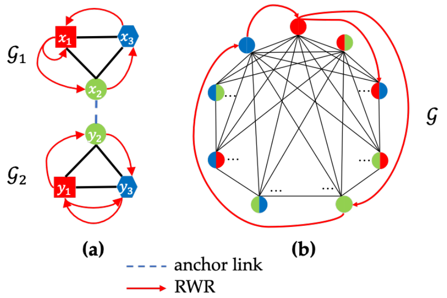


Figure 3: An illustrative example of RWR: (a) RWR on separated graphs: RWRs encode positional information for $x_1, x_3 \in \mathcal{G}_1$ and $y_1, y_3 \in \mathcal{G}_2$ w.r.t. the unified landmark nodes x_2, y_2; (b) RWR on the product graph: the co-occurrence of RWR$_1$ = $(x_1, x_2, x_3, x_1, x_1)$ on $\mathcal{G}_1$ and RWR$_2$ = $(y_1, y_2, y_3, y_1, y_3)$ on $\mathcal{G}_2$ is equivalent to the occurrence of RWR$=((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_1, y_1), (x_1, y_3))$ on $\mathcal{G} = \mathcal{G}_1 \otimes \mathcal{G}_2$.

RWR on separated graphs can capture intra-network topology information w.r.t. shared landmarks [34]. An example is shown in Figure 3 (a). Given the k-th anchor node pair $(x_{l_k}, y_{l_k}) \in \mathcal{L}$ where $x_{l_k} \in \mathcal{V}_1, y_{l_k} \in \mathcal{V}_2$ and $k = 1, \cdots, |\mathcal{L}|$, the RWR score vectors that depict the relative positions of nodes w.r.t. the corresponding anchor node (e.g., nodes in $\mathcal{V}_1$ w.r.t. x_{l_k}) can be obtained by [29]

$$\mathbf{r}_{x_{l_k}} = (1-\beta)\mathbf{W}_1 \mathbf{r}_{x_{l_k}} + \beta \mathbf{e}_{x_{l_k}}, \ \mathbf{r}_{y_{l_k}} = (1-\beta)\mathbf{W}_2 \mathbf{r}_{y_{l_k}} + \beta \mathbf{e}_{y_{l_k}} \quad (2)$$

where β is the restart probability, $\mathbf{W}_1 = (\mathbf{D}_1^{-1}\mathbf{A}_1)^{\top}$ is the transpose of the row normalized matrix of $\mathbf{A}_1$, $\mathbf{e}_{x_{l_k}}$ is an n_1-dimensional one-hot vector with $\mathbf{e}_{x_{l_k}}(x_{l_k}) = 1$, and similarly for $\mathbf{W}_2, \mathbf{e}_{y_{l_k}}$. With the set of anchor node pairs $\mathcal{L}$, we compute the RWR embedding matrices $\mathbf{R}_1 \in \mathbb{R}^{n_1 \times |\mathcal{L}|}$ and $\mathbf{R}_2 \in \mathbb{R}^{n_2 \times |\mathcal{L}|}$ by concatenating the RWR scores w.r.t. different anchor node pairs via $\mathbf{R}_1 = [\mathbf{r}_{x_{l_1}} \| \cdots \| \mathbf{r}_{x_{l_{|\mathcal{L}|}}}], \mathbf{R}_2 = [\mathbf{r}_{y_{l_1}} \| \cdots \| \mathbf{r}_{y_{l_{|\mathcal{L}|}}}]$. Note that these node embeddings $\mathbf{R}_1, \mathbf{R}_2$ naturally integrate the structural information of each input network by RWR. To compute the transport cost, we further leverage node attributes $\mathbf{X}_1, \mathbf{X}_2$ by a linear combination with the cost based on RWR embeddings, which is computed by

$$\mathbf{C}_{\text{node}} = \alpha e^{-\mathbf{R}_1 \mathbf{R}_2^{\top}} + (1-\alpha) e^{-\mathbf{X}_1 \mathbf{X}_2^{\top}} \quad (3)$$

where α is the weight parameter and $\mathbf{C}_{\text{node}}(x_i, y_j)$ describes the cost of transporting/aligning node $x_i \in \mathcal{G}_1$ to $y_j \in \mathcal{G}_2$.

This transport cost $\mathbf{C}_{\text{node}}$ is computed solely based upon each network individually, and hence may overlook the structural correlations among node pairs across different networks. To encode this cross-network information, we assume that not only the aligned node pairs, but also their neighboring node pairs should be similar. To this end, instead of measuring the cost of transporting node x_1 and y_1, we turn to measure the cost of transporting two synchronized RWRs starting from x_1 and y_1. To obtain the co-occurrence of RWR starting from node $x_1 \in \mathcal{G}_1$, named RWR$_1(x_1)$ on $\mathcal{G}_1$, and RWR$_2(y_1)$ on $\mathcal{G}_2$, from the view of product graph, it is equivalent to measure the occurrence of RWR$((x_1, y_1))$ on the product graph $\mathcal{G} = \mathcal{G}_1 \otimes \mathcal{G}_2$. An example is given in Figure 3 (b). For simplicity, we denote the node pair visited at step i as $s_i = (x_i, y_i)$. Given the start node pair s_1, the RWR-level cross-network cost $\mathbf{C}_{\text{rwr}}(s_1)$ can be

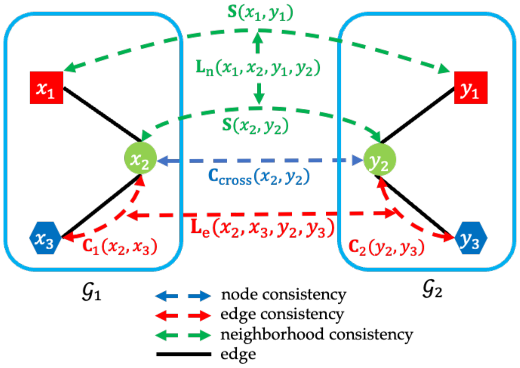


Figure 4: An example of the consistency regularization.

computed as the expected discounted sum of node cost conditioned on s_1, that is $[\mathrm{C}_{\text{node}}(s_i)|s_1]$, along RWR on the product graph.

$$\begin{aligned}\mathrm{C}_{\text{rwr}}(s_1) &= \mathbb{E}_{\mathbf{W}} \textstyle\sum_{i=1}^{\infty} \gamma^{i-1} \left[\mathrm{C}_{\text{node}}(s_i)|s_1\right] \\ &= (1{+}\beta)\mathrm{C}_{\text{node}}(s_1) + (1{-}\beta)\mathbb{E}_{\mathbf{W}} \textstyle\sum_{i=1}^{\infty} \gamma^{i} \left[\mathrm{C}_{\text{node}}(s_{i+1})|s_1\right] \\ &= (1{+}\beta)\mathrm{C}_{\text{node}}(s_1) + (1{-}\beta)\mathbb{E}_{\mathbf{W}} \textstyle\sum_{i=1}^{\infty} \gamma^{i}\mathbf{W} \left[\mathrm{C}_{\text{node}}(s_i)|s_1\right] \\ &= (1{+}\beta)\mathrm{C}_{\text{node}}(s_1) + (1{-}\beta)\gamma\mathbf{W}\mathrm{C}_{\text{rwr}}(s_1)\end{aligned}$$

where γ is the discounted factor of RWR and $\mathbf{W} = \mathbf{W}_2 \otimes \mathbf{W}_1$ is the transition matrix of the product graph $\mathcal{G}$. Since $\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^{\mathsf{T}} \otimes \mathbf{A})\mathrm{vec}(\mathbf{B})$, the above equation can be re-written as

$$\mathrm{C}_{\text{rwr}} = (1+\beta)\mathrm{C}_{\text{node}} + (1-\beta)\gamma\mathbf{W}_1\mathrm{C}_{\text{rwr}}\mathbf{W}_2^{\mathsf{T}} \tag{4}$$

Note that Eq. (4) is known as Sylvester equation and is guaranteed to converge through fixed point iteration [24].

3.2 Consistency-regularized OT

Most, if not all, of existing OT-based methods ignore the alignment consistency principle, which assumes that the alignment between two pairs of nodes across input networks should be consistent from multiple perspectives. Inspired by [4, 40], for two aligned nodes $x \in \mathcal{G}_1, y \in \mathcal{G}_2$ and their neighbor nodes $x' \in \mathcal{N}(x), y' \in \mathcal{N}(y)$, we propose three types of consistency. First, the *node consistency* assumes that aligned nodes (x, y) should be similar in terms of node attributes and topology. Second, the *edge consistency* assumes that if the corresponding neighbors x', y' of x, y are likely to be aligned, then the edges $(x, x'), (y, y')$ are likely to exhibit similar relationships. For example, in Figure 4, if neighboring nodes x_2, y_2 and x_3, y_3 are likely to be aligned respectively, then (x_2, x_3) and (y_2, y_3) are expected to exhibit similar intra-network relationships. Third, the *neighborhood consistency* assumes that close nodes in one graph are likely to match to close nodes in another graph, i.e., the alignment score of neighboring node pairs should be similar [4]. For example, in Figure 4, for neighboring nodes pairs $x_1, x_2 \in \mathcal{G}_1$ and $y_1, y_2 \in \mathcal{G}_2$, (x_1, y_1) and (x_2, y_2) are expected to have similar alignment scores.

The OT problem in Eq. (1) naturally incorporates node consistency by using the transport cost matrix C_{rwr} to re-weight the alignment S by element-wise product, that is

$$L_{\text{node}} = \sum_{x,y} \mathrm{C}_{\text{rwr}}(x, y)\mathrm{S}(x, y) = \langle \mathrm{C}_{\text{rwr}}, \mathrm{S} \rangle \tag{5}$$

In addition, for edge consistency, we integrate it with an additional regularization term L_e to emphasize the consistency of node pair relationships. From the view of dual graphs, where edges are projected as nodes and vice versa, the edge consistency can be interpreted as aligning edges of two networks [2]. To measure the intra-network relationships along edges, we compute the intra-network dissimilarity matrices $\mathbf{C}_1, \mathbf{C}_2$ based on node attributes by

$$\mathbf{C}_1 = e^{-\mathbf{X}_1\mathbf{X}_1^{\mathsf{T}}} \odot \mathbf{A}_1,\ \mathbf{C}_2 = e^{-\mathbf{X}_2\mathbf{X}_2^{\mathsf{T}}} \odot \mathbf{A}_2 \tag{6}$$

which we further use to define the edge consistency regularization term as

$$\begin{aligned} L_e &= \sum_{x,x',y,y'} \mathbf{L}_e(x, x', y, y') \\ &= \sum_{\substack{x,y \\ x' \in \mathcal{N}(x), y' \in \mathcal{N}(y)}} |\mathbf{C}_1(x, x') - \mathbf{C}_2(y, y')|^2 \mathrm{S}(x, y)\mathrm{S}(x', y') \end{aligned} \tag{7}$$

Note that Eq. (7) is equivalent to the Gromov-Wasserstein distance (GWD). Based on [23], we further derive it into an inner product form for fast computation as

$$\begin{aligned} L_e &= \langle \mathbf{L}, \mathbf{S} \rangle \\ \text{where } &\begin{cases} \mathbf{L} = \mathbf{C}_1^2\boldsymbol{\mu}\mathbf{1}_{n_2}^{\mathsf{T}} + \mathbf{1}_{n_1}\boldsymbol{\nu}^{\mathsf{T}}\mathbf{C}_2^2 - 2\mathbf{C}_1\mathbf{S}\mathbf{C}_2^{\mathsf{T}} \\ \boldsymbol{\mu} = \dfrac{\mathbf{1}_{n_1}}{n_1}, \boldsymbol{\nu} = \dfrac{\mathbf{1}_{n_2}}{n_2} \end{cases} \end{aligned} \tag{8}$$

The neighborhood consistency $\mathbf{L}_n(x, x', y, y')$ measures the consistency of alignment scores among neighboring node pairs. In other words, for any node pairs (x, y), we want the alignment score $\mathrm{S}(x, y)$ to be similar to the average alignment score $\hat{\mathrm{S}}(x, y)$ among its neighbors. The average alignment score can be defined as

$$\begin{aligned} \hat{\mathrm{S}}(x, y) &= \frac{1}{|\mathcal{N}(x)||\mathcal{N}(y)|} \sum_{x' \in \mathcal{N}(x), y' \in \mathcal{N}(y)} \mathrm{S}(x', y') \\ &= \mathbf{W}_1^{\mathsf{T}}\mathbf{S}\mathbf{W}_2 \end{aligned}$$

Bregman divergence is used to depict the distance between S and $\hat{\mathrm{S}}$, and the neighborhood consistency regularization can be defined as

$$\begin{aligned} L_n &= \sum_{x,x',y,y'} \mathbf{L}_n(x, x', y, y') \\ &= \sum_{x,y} \left[\mathrm{S}(x, y) \log \frac{\mathrm{S}(x, y)}{\hat{\mathrm{S}}(x, y)} - \mathrm{S}(x, y) + \hat{\mathrm{S}}(x, y) \right] \\ &= -\langle \log \hat{\mathrm{S}}, \mathrm{S} \rangle + \langle \log \mathrm{S}, \mathrm{S} \rangle \end{aligned} \tag{9}$$

The last equation is due to $\sum_{x,y} \mathrm{S}(x, y) = 1$ and $\sum_{x,y} \hat{\mathrm{S}}(x, y) = 1$.

Furthermore, to exploit the supervision information $\mathcal{L}$, we introduce an *alignment preference* regularization term $\mathbf{L}_a$ to encode prior alignment preference $\mathbf{H}$. Here, $\mathbf{H} \in \mathbb{R}^{n_1 \times n_2}$ is a uniform distribution on anchor node pairs, i.e., $\mathbf{H}(x, y) = \frac{1}{|\mathcal{L}|}$ if and only if $(x, y) \in \mathcal{L}$. By using Bregman divergence to measure the distance between S and $\mathbf{H}$, the alignment preference regularization is defined as

$$\begin{aligned} L_a &= \sum_{x,y} \mathbf{L}_a(x, y) \\ &= \sum_{x,y} \left[\mathrm{S}(x, y) \log \frac{\mathrm{S}(x, y)}{\mathbf{H}(x, y)} - \mathrm{S}(x, y) + \mathbf{H}(x, y) \right] \\ &= -\langle \log \mathbf{H}, \mathrm{S} \rangle + \langle \log \mathrm{S}, \mathrm{S} \rangle \end{aligned} \tag{10}$$

The last equation is due to $\sum_{x,y} \mathrm{S}(x, y) = \sum_{x,y} \mathbf{H}(x, y) = 1$.

Algorithm 1 Position-Aware Transport Cost

```
Input: (1) networks $\mathcal{G}_1 = \{\mathcal{V}_1, \mathbf{A}_1, \mathbf{X}_1\}, \mathcal{G}_2 = \{\mathcal{V}_2, \mathbf{A}_2, \mathbf{X}_2\}$, (2) the set of anchor node pairs $\mathcal{L}$, (3) parameter $\alpha, \beta, \gamma$, (4) the maximum iteration number $T$.
Output: cross-network cost $\mathbf{C}_{\text{rwr}}$.
Compute RWR scores by running Eq. (2) iteratively;
Compute node-level cost $\mathbf{C}_{\text{node}}$ by Eq. (3);
Compute RWR-level cost $\mathbf{C}_{\text{rwr}}$ by running Eq. (4) iteratively;
return transport cost $\mathbf{C}_{\text{rwr}}$.
```

At last, by combining Eq. (5)-(10), the overall consistency-regularized OT problem is formulated as

$$\min_{\mathrm{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} f(\mathrm{S}) = L_{\text{node}} + \lambda_e L_e + \lambda_n L_n + \lambda_a L_a \tag{11}$$

where $\lambda_e, \lambda_n, \lambda_a$ are the hyperparameters that control the importance of each regularization term.

4 ALGORITHM AND ANALYSIS

In this section, we present and analyze our proposed algorithm PARROT. In Section 4.1, we first present the computation of the position-aware transport cost. Then we introduce several approximations to consistency regularization for convexity guarantee and the proximal operator for convergence guarantee. We further decompose the regularized OT into a series of convex subproblems that can be regarded as classic OT problems with modified transport costs and propose PARROT for fast computation. Convergence and complexity analyses of PARROT are carried out in Section 4.2.

4.1 PARROT: Optimization Algorithm

The overall framework of PARROT can be divided into two parts: (1) calculating the position-aware transport cost based on Eq. (2)-(4), and (2) solving the regularized OT problem in Eq. (11).

To compute the transport cost $\mathbf{C}_{\text{rwr}}$, we apply the fixed-point algorithm to iteratively update $\mathbf{r}_{x_{l_k}}, \mathbf{r}_{y_{l_k}}$ and $\mathbf{C}_{\text{rwr}}$ as in Eq. (2) and Eq. (4). Since the eigenvalues of the normalized adjacency matrices $\mathbf{W}_1, \mathbf{W}_2$ lie in $[-1, 1]$, the convergence can be guaranteed. The entire computation of $\mathbf{C}_{\text{rwr}}$ is summarized in Algorithm 1.

To solve Eq. (11), we propose PARROT based on the constrained proximal point method for fast solution with guaranteed convergence. Proximal point method is a widely adopted method to find the optimal solution to convex optimization problem. In general, Given a fixed convex objective function $f(\mathrm{S})$ defined on the feasible region $\mathcal{S}$, the proximal point method finds the global optimal solution $\mathrm{S}^* \in \mathcal{S}$ that minimizes $f(\mathrm{S})$ by generating a solution sequence $\{\mathrm{S}^{(t)}\}_{t=1,2,...}$ to the following subproblems.

$$\mathrm{S}^{(t+1)} = \arg\min_{\mathrm{S} \in \mathcal{S}} f(\mathrm{S}) + \lambda L_p^{(t)} \tag{12}$$

where $L_p^{(t)}$ is the proximal operator constraining the distance between two consecutive solutions for the sake of convergence. In this work, we define $L_p^{(t)}$ as the Bregman divergence between S and $\mathrm{S}^{(t)}$ as follows.

$$\begin{aligned} L_p^{(t)} &= \sum_{x,y} \left[\mathrm{S}(x,y) \log \frac{\mathrm{S}(x,y)}{\mathrm{S}^{(t)}(x,y)} - \mathrm{S}(x,y) + \mathrm{S}^{(t)}(x,y) \right] \\ &= -\langle \log \mathrm{S}^{(t)}, \mathrm{S} \rangle + \langle \log \mathrm{S}, \mathrm{S} \rangle \end{aligned} \tag{13}$$

The last equation is due to $\sum_{x,y} \mathrm{S}(x,y) = \sum_{x,y} \mathrm{S}^{(t)}(x,y) = 1$.

However, the edge consistency in Eq. (8) and neighborhood consistency regularization in Eq. (9) are non-convex terms. To tackle the non-convexity, our key idea is two-fold: (1) *subproblem decomposition* and (2) *convex approximation.* First (subproblem decomposition), the original problem in Eq. (12) is decomposed into a series of subproblems where $\mathbf{L}^{(t)}$ in $L_e^{(t)}$ and $\hat{\mathrm{S}}^{(t)}$ in $L_n^{(t)}$ are fixed.

$$\begin{aligned} \mathrm{S}^{(t+1)} &= \arg\min_{\mathrm{S} \in \mathcal{S}} f_t(\mathrm{S}) + \lambda_p L_p^{(t)} \\ f_t(\mathrm{S}) &= L_{\text{node}} + \lambda_e L_e^{(t)} + \lambda_n L_n^{(t)} + \lambda_a L_a \end{aligned} \tag{14}$$

where $f_t(\mathrm{S})$ changes along the proximal point iteration. Second (convex approximation), we introduce the following approximations to Eq. (8) and Eq. (9) to guarantee the convexity of $f_t(\mathrm{S})$.

$$\begin{aligned} L_e^{(t)} &= \langle \mathbf{L}^{(t)}, \mathrm{S} \rangle \approx \langle \mathbf{C}_1^2 \boldsymbol{\mu} \mathbf{1}_{n_2}^\top + \mathbf{1}_{n_1} \boldsymbol{\nu}^\top \mathbf{C}_2^2 - 2\mathbf{C}_1 \mathrm{S}^{(t)} \mathbf{C}_2^\top, \mathrm{S} \rangle \\ L_n^{(t)} &\approx -\langle \log \hat{\mathrm{S}}^{(t)}, \mathrm{S} \rangle + \langle \log \mathrm{S}, \mathrm{S} \rangle \end{aligned} \tag{15}$$

Combining Eq. (14) and (15), each subproblem is formulated as

$$\begin{aligned} &\min_{\mathrm{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} f_t(\mathrm{S}) + \lambda_p L_p(\mathrm{S}, \mathrm{S}^{(t)}) \\ &= \langle \underbrace{\mathbf{C}_{\text{rwr}}}_{\text{node}} + \underbrace{\lambda_e \mathbf{L}^{(t)}}_{\text{edge}} + \underbrace{\lambda_n \log \frac{\mathrm{S}}{\hat{\mathrm{S}}^{(t)}}}_{\text{neighborhood}} + \underbrace{\lambda_a \log \frac{\mathrm{S}}{\mathbf{H}}}_{\text{preference}} + \underbrace{\lambda_p \log \frac{\mathrm{S}}{\mathrm{S}^{(t)}}}_{\text{proximal}}, \mathrm{S} \rangle \\ &= \langle \mathbf{C}^{(t)}, \mathrm{S} \rangle + \lambda \langle \log \mathrm{S}, \mathrm{S} \rangle \end{aligned} \tag{16}$$

where $\mathbf{C}^{(t)} = \mathbf{C}_{\text{rwr}} + \lambda_e \mathbf{L}^{(t)} - \lambda_n \log \hat{\mathrm{S}}^{(t)} - \lambda_a \log \mathbf{H} - \lambda_p \log \mathrm{S}^{(t)}$ and $\lambda = \lambda_n + \lambda_a + \lambda_p$. Notice that $\mathbf{C}^{(t)}$ is a fixed matrix, so that each subproblem in Eq. (16) can be regarded as a classic OT problem with modified cost $\mathbf{C}^{(t)}$ and can be efficiently solve by the Sinkhorn algorithm [6]. Specifically, starting with $\mathbf{b}^{(0)} = \frac{\mathbf{1}_{n_2}}{n_2}$, the Sinkhorn algorithm computes the scaling vector by the following iteration

$$\mathbf{a}^{(l)} = \frac{\boldsymbol{\mu}}{e^{-\mathbf{C}^{(t)}/\lambda} \mathbf{b}^{(l-1)}}, \ \mathbf{b}^{(l)} = \frac{\boldsymbol{\nu}}{e^{-\mathbf{C}^{(t)\top}/\lambda} \mathbf{a}^{(l)}}, \forall l = 1, ..., L \tag{17}$$

and the optimal solution can be obtained by

$$\mathrm{S}^{(t)} = \text{diag}(\mathbf{a}^{(L)}) e^{-\mathbf{C}^{(t)}/\lambda} \text{diag}(\mathbf{b}^{(L)}) \tag{18}$$

Note that the exponential term in Eq. (17) amplifies the gaps between values in $\mathbf{C}^{(t)}$, hence providing a more deterministic, or noise-reduced, alignment result. In fact, the doubly-stochastic matrix $\mathrm{S}^{(t)}$ will approach a hard alignment when λ approaches 0 [16].

Although the proximal point method has guaranteed convergence for a fixed convex optimization problem based on the descent property, i.e. $f(\mathrm{S}^{(t+1)}) \leq f(\mathrm{S}^{(t)})$, this is not the case in our formulation as the objective $f_t(\mathrm{S})$ changes along the iteration. The major problem is that alternating the objective function may violate the descent property. To tackle this problem, we propose the following constrained proximal point method that only updates the objective function when the decrease property is satisfied.

$$\begin{aligned} \mathrm{S}^{(t+1)} &= \arg\min_{\mathrm{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} f_t(\mathrm{S}) + \lambda \langle \log \mathrm{S}, \mathrm{S} \rangle \\ \text{where } f_t(\mathrm{S}) &= \begin{cases} \langle \mathbf{C}^{(t)}, \mathrm{S} \rangle, \text{ if } \langle \mathbf{C}^{(t)}, \mathrm{S}^{(t)} \rangle \leq f_{t-1}(\mathrm{S}^{(t)}) \\ f_{t-1}(\mathrm{S}), \text{ otherwise} \end{cases} \end{aligned} \tag{19}$$

Under such update constraint, the objective function, as we theoretically prove, is non-increasing and converges along the optimization

Algorithm 2 PARROT

```
Input: (1) networks $\mathcal{G}_1 = \{\mathcal{V}_1, \mathbf{A}_1, \mathbf{X}_1\}, \mathcal{G}_2 = \{\mathcal{V}_2, \mathbf{A}_2, \mathbf{X}_2\}$, (2)
    prior alignment preference $\mathbf{H}$, (3) parameters $\alpha, \beta, \gamma, T, \lambda_{e/n/p/a}$.
Output: the alignment matrix $\mathbf{S}$.
Compute cross-network cost $\mathbf{C}_{\mathrm{rwr}}$ by Algorithm 1;
Compute intra-network cost $\mathbf{C}_1, \mathbf{C}_2$ by Eq. (6);
$\boldsymbol{\mu} = \frac{\mathbf{1}_{n_1}}{n_1}, \boldsymbol{\nu} = \frac{\mathbf{1}_{n_2}}{n_2}, \mathbf{S}^{(0)} = \boldsymbol{\mu} \otimes \boldsymbol{\nu}^{\mathrm{T}}, \lambda = \lambda_n + \lambda_a + \lambda_p$;
for $t = 0, 1, ..., T$ do
    Compute $\mathbf{L}^{(t)}$ by Eq. (8) and $\mathbf{C}^{(t)}$ by Eq. (16);
    if $\langle \mathbf{C}^{(t)}, \mathbf{S}^{(t)} \rangle \leq f_{t-1}(\mathbf{S}^{(t)})$ then
        Update objective function $f_t(\mathbf{S}) = \langle \mathbf{C}^{(t)}, \mathbf{S} \rangle$;
    else
        Objective function stays unchanged $f_t(\mathbf{S}) = f_{t-1}(\mathbf{S})$;
    end if
    Solve $\mathbf{S}^{(t+1)} = \underset{\mathbf{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})}{\arg\min}\ f_t(\mathbf{S}) + \lambda_p L_p^{(t)}$ by Eq. (17) and (18);
end for
return alignment matrix $\mathbf{S}$.
```

process. Therefore, our proposed PARROT summarized in Algorithm 2 provides a fast solution to the consistency-regularized OT problem with guaranteed convergence.

4.2 Proof and Analysis

In this subsection, we provide theoretical analyses of our proposed PARROT. We first show the convergence of position-aware transport cost computation (Propositions 1 and 2). Then we show the strict convexity of subproblems in Eq. (16) (Lemma 1) and the convergence analysis of PARROT (Lemma 2 and Theorem 1). The complexity analysis (Theorem 2) is carried out thereafter. Without loss of generality, we assume that graphs share a comparable size (i.e., $O(n_1) \approx O(n_2) \approx O(n)$ nodes and $O(m_1) \approx O(m_2) \approx O(m)$ edges).

Proposition 1. Convergence of RWR on separated graphs. *With the error tolerance ϵ, $T = \log_{1-\beta} \epsilon$ iterations guarantee $\|\mathbf{R}_1^* - \mathbf{R}_1^{(T)}\|_\infty \leq \epsilon$ and $\|\mathbf{R}_2^* - \mathbf{R}_2^{(T)}\|_\infty \leq \epsilon$.*

Proposition 2. Convergence of RWR on the product graph. *With the error tolerance ϵ, $T = \log_{(1-\beta)\gamma}[(1-\gamma)\epsilon]$ iterations guarantee $\|\mathbf{C}_{rwr}^* - \mathbf{C}_{rwr}^{(T)}\|_\infty \leq \epsilon$.*

Lemma 1. **Strict convexity of Eq.** (16)**.** *The subproblem in Eq.* (16) *is strictly convex.*

Lemma 2. **Convergence of proximal point method [30].** *Given a convex objective function $f(\mathbf{S})$, the solution sequence $\{f(\mathbf{S}^{(t)})\}$ given by proximal point method is non-increasing and converges.*

Theorem 1. **Convergence of PARROT.** *The solution sequence $\{f_t(\mathbf{S}^{(t)})\}$ given by PARROT is non-increasing and converges.*

Theorem 2. **Complexity of PARROT.** *The overall time complexity of PARROT is $O(Tmn + TLn^2)$, where T is the number of outer iterations and L is the number of inner iterations in PARROT.*

All the proofs for the above propositions, lemmas, and theorems are provided in Appendix A.

5 EXPERIMENT

We apply the proposed algorithm PARROT to the network alignment task and evaluate it from the following aspects:

- Q1. How effective is the proposed PARROT in both plain and attributed network alignment tasks?
- Q2. How efficient and scalable is the proposed PARROT?
- Q3. How is the empirical convergence of PARROT?
- Q4. To what extent does PARROT benefit from different components of our method?

5.1 Experimental Setup

Datasets. Our method is evaluated on both plain and attributed networks. The dataset statistics are shown in Table 5. Detailed descriptions and experimental settings are in Appendix C[1].

Baseline methods. PARROT is compared with the following methods on both attributed and plain networks under the semi-supervised setting, including (1) Consistency-based methods: IsoRank [26] and FINAL [40], and (2) Embedding-based methods: IONE [12], CrossMNA [5], DANA [10], REGAL [9], NetTrans [44], BRIGHT [34] and NeXtAlign [41]. Since many existing OT-based methods [13, 14, 20, 32] can not scale to large networks, we evaluate two unsupervised OT-based methods S-GWL [31] and GOAT [25] on the small Phone-Email dataset with 1,000 nodes. For a fair comparison, we ablated the proposed PARROT with the known anchor links as well as RWR on separated graphs (denoted as "PARROT (ablated)").

Metrics. In our experiment, we evaluate the effectiveness in terms of Hits@K and Mean Reciprocal Rank (MRR). Given a test node $x \in \mathcal{G}_1$, if the corresponding node $x' \in \mathcal{G}_2$ is among the top-K most similar nodes in $\mathcal{G}_2$, it is regarded as a hit. For a test dataset with n node pairs, The Hits@K is computed by Hits@$K = \frac{\text{\# of hits}}{n}$. MRR is a widely-adopted metric computed by the average of the inverse of alignment ranking MRR $= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\mathrm{rank}((a_i,x_i))}$.

5.2 Effectiveness Results

Comparison with consistency and embedding-based methods. We first compare the alignment performance with consistency and embedding-based methods under the semi-supervised setting on both attributed and plain networks. Using 20% data as the prior knowledge, the results are shown in Tables 2 and 3. Compared with consistency-based methods, PARROT achieves up to 17% improvement in Hits@30 and 25% improvement in MRR on plain network tasks. On attributed network tasks, PARROT achieves up to 9% improvement in Hits@30 and 27% improvement in MRR compare with the state-of-the-art. These results indicate that the OT framework consistently provides a better distance metric than those Frobenius-like distances. Besides, the improvements on plain tasks are even more significant than those on attributed tasks, indicating the position-aware transport cost can depict the topology information precisely. Compared with embedding-based methods, PARROT outperforms all embedding-based methods on plain tasks and achieves up to 7% improvement in Hits@30 and 13% improvement in MRR. On attributed tasks, PARROT outperforms embedding-based methods achieving up to 1% improvement in Hits@30 and 3% improvement in MRR compared with the best competitor.

[1] Code and datasets are available at https://github.com/zhichenz98/PARROT-WWW23.

Table 2: Comparison with consistency and embedding-based methods on plain network alignment.

Dataset	Foursquare-Twitter				ACM-DBLP			
Metrics	Hits@1	Hits@10	Hits@30	MRR	Hits@1	Hits@10	Hits@30	MRR
IsoRank	0.023	0.117	0.223	0.055	0.151	0.632	0.796	0.294
IONE	0.042	0.168	0.281	0.085	0.312	0.748	0.843	0.467
FINAL	0.051	0.238	0.342	0.110	0.193	0.692	0.832	0.353
CrossMNA	0.000	0.036	0.119	0.015	0.082	0.646	0.795	0.238
DANA	<u>0.127</u>	<u>0.344</u>	<u>0.486</u>	0.197	0.234	0.432	0.467	0.290
BRIGHT	0.064	0.252	0.335	0.130	<u>0.405</u>	0.813	0.841	0.539
NextAlign	0.102	0.277	0.371	<u>0.224</u>	0.403	<u>0.816</u>	<u>0.870</u>	<u>0.585</u>
PARROT	**0.245**	**0.409**	**0.508**	**0.304**	**0.619**	**0.912**	**0.940**	**0.719**

Table 3: Comparison with consistency and embedding-based methods on attributed network alignment.

Dataset	Cora1-Cora2				ACM(A)-DBLP(A)			
Metrics	Hits@1	Hits@10	Hits@30	MRR	Hits@1	Hits@10	Hits@30	MRR
FINAL	0.710	0.881	0.907	0.773	0.397	0.833	0.925	0.541
REGAL	0.453	0.611	0.689	0.510	0.361	0.629	0.696	0.449
NetTrans	<u>0.993</u>	<u>0.999</u>	**1.000**	<u>0.995</u>	<u>0.692</u>	<u>0.931</u>	<u>0.960</u>	<u>0.774</u>
BRIGHT	0.839	0.991	0.997	0.904	0.453	0.878	0.922	0.599
NeXtAlign	0.492	0.729	0.786	0.577	0.487	0.859	0.915	0.633
PARROT	**0.996**	**1.000**	**1.000**	**0.998**	**0.721**	**0.960**	**0.971**	**0.806**

Table 4: Comparison with OT-based methods on plain network alignment under unsupervised setting. PARROT (ablated) is the unsupervised version and PARROT is the semi-supervised version with 20% data as prior knowledge.

Dataset	Phone-Email			
Metrics	Hits@1	Hits@10	Hits@30	MRR
S-GWL	0.003	0.021	0.058	0.014
GOAT	0.000	0.009	0.026	0.005
PARROT (ablated)	<u>0.043</u>	<u>0.314</u>	<u>0.649</u>	<u>0.131</u>
PARROT	**0.323**	**0.749**	**0.931**	**0.469**

Comparison with OT-based methods. We also compare the alignment performance with OT-based methods. We use the unsupervised PARROT (ablated) for a fair comparison and also report the performance of PARROT using 20% data as prior knowledge in Table 3. It is shown that without the anchor links/supervision, none of the OT-based methods in Table 4 performs well. Nonetheless, the proposed PARROT (ablated) still consistently outperforms both S-GWL [31] and GOAT [25], thanks to the RWR-level cross-network transport cost and consistency regularization. With the additional supervision of anchor links and more importantly equipped with our RWR on separated graphs, the proposed PARROT (i.e., the last row of Table 4) leads to significant alignment performance improvement over both existing OT-based methods. In Section 5.4, we present further ablation studies to quantify the relative contributions of different components in the proposed PARROT.

5.3 Scalability Results

We compare the running time of the proposed PARROT with that of S-GWL [31] and GOAT [25], and results are shown in Figure 5. Note that the numbers above the blue/green curves are the ratios of MRR of S-GWL/GOAT to that of PARROT. For example, for networks with 20,000 edges, PARROT runs 100 times faster than S-GWL while S-GWL's MRR is only 37% of that of PARROT, and PARROT runs 5 times faster than GOAT while GOAT's MRR is about 56% of PARROT. Under 600-second running time limit, PARROT can process networks 6 times the size of S-GWL and twice the size of GOAT. Combining with the effectiveness comparison in Table 4, we conclude that the proposed PARROT improves the existing OT-based methods in both effectiveness and scalability (up to 100×+ speed-up).

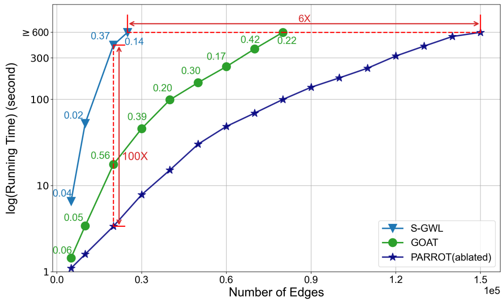
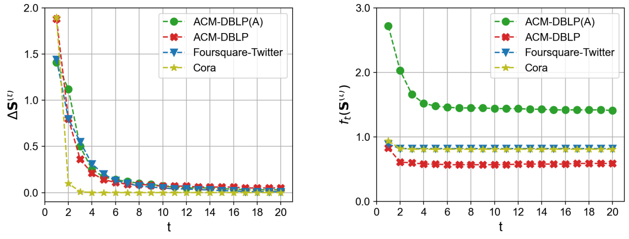


Figure 5: Scalability analysis: numbers over curves indicate the ratio of MRR of baseline methods and that of PARROT.

5.4 Convergence Results

We evaluate the convergence of the proposed PARROT in terms of the difference between two consecutive solutions $\Delta \mathbf{S}^{(t)} = \|\mathbf{S}^{(t)} - \mathbf{S}^{(t-1)}\|_1$ and the values of the objective function $f_t(\mathbf{S}^{(t)})$. Results given in Figure 6 show that the solution and the objective function generated by PARROT converges along the optimization process.



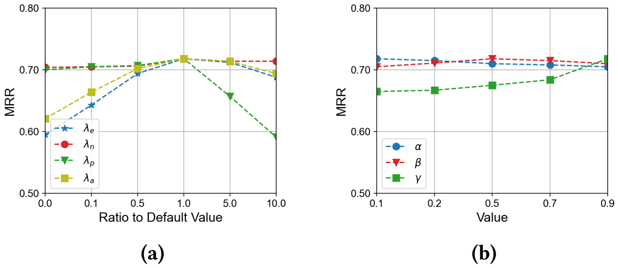
(a) (b)

Figure 6: Convergence analysis. (a) Difference between consecutive solutions; (b) Values of the objective function.

5.5 Analysis of the Method

Hyperparameter sensitivity study. We analyze how the alignment performance varies with different hyperparameters. We first analyze regularization parameters $(\lambda_e, \lambda_n, \lambda_p, \lambda_a)$ by varying between 0.1 to 10 times of the default value. Then we analyze ratio parameters (α, β, γ) with values from $\{0.1, 0.2, 0.5, 0.7, 0.9\}$. Results are shown in Figure 7. It shown that our method is robust to hyperparameters in a relatively wide range.

Ablation study on regularization. To evaluate the effectiveness of our proposed consistency regularization, we use PARROT without any regularization as baseline and compare the performance of each regularization (edge/neighborhood/preference) on different datasets. Results are shown in Figure 8. Compared with the baseline, all regularization terms can boost the alignment performance. The edge consistency leads to the largest improvement. As stated before, node consistency and edge consistency are mutually



(a) **(b)**

Figure 7: Hyperparameter study on ACM-DBLP: (a) Study on regularization parameters; (b) Study on ratio parameters.

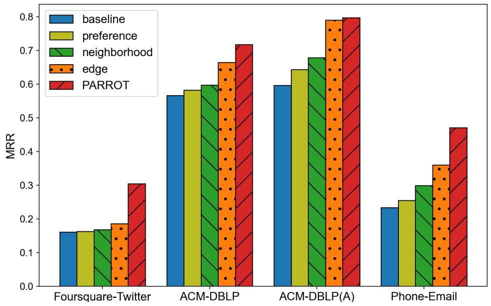


Figure 8: Study on consistency regularization.

complementary which forces the OT framework to align nodes and edges simultaneously. Besides, the neighborhood consistency and alignment preference also improve the performance to some extent. Therefore, this ablation study validates the necessities of introducing consistency regularization to the OT framework.

6 RELATED WORK

In this section, we review the related work, which can be categorized into two groups: network alignment and optimal transport.

6.1 Network Alignment

Most existing network alignment methods can be categorized into consistency-based and embedding-based methods. Consistency-based methods are built upon the linearity and/or consistency assumptions. The linearity assumption assumes a noisy permutation **P** between two networks forming a Frobenius-like objective [11, 38]. Some variants address the network alignment problem from the view of alignment consistency. IsoRank [26] propagates node similarities on the product graph to achieve topology consistency. FINAL [40] further integrates node, edge, and topology consistency to deal with attributed networks. Although consistency-based methods address neighborhood topology and attribute consistency, the global structure is often under-exploited. Besides, the consistency assumption may be violated due to network disparity [41].

Another line of work is based on node embeddings. These methods aim to find informative low-dimensional node embeddings that (1) preserve the topology information of each network and (2) make anchor node pairs as close as possible [33, 34]. To achieve these two goals, IONE [12] adopts the follower/followee-ship to generate embeddings preserving node proximities. REGAL [9] learns effective and low-variance node embeddings via cross-network matrix factorization. CrossMNA [5] utilizes diverse embeddings to address global/intra/inter-network features. NetTrans [44] handles network alignment from the view of network transformation. BRIGHT [34] and FITO [35] generate position embeddings via the random walk with restart to tackle the space disparity issue. DANA [10] and RNA [47] study the network alignment problem via adversarial learning to obtain robust alignments. Several recent works focus on augmenting dataset for model training. Attent [46] involves human in loop for data labeling via active learning. NeXtAlign [41] studies the negative sampling strategy to achieve a balance between alignment consistency and disparity. CPUGA [21] introduces a non-sampling schema that progressively selects and utilizes trusty node pairs for model training. The advantage of embedding-based methods comes from the power of nonlinear functions, at the cost of the computationally expensive training processes whose convergence guarantee is intractable. Besides, embedding-based methods may introduce the space disparity issue [41, 44].

6.2 Optimal Transport

OT is recently introduced to cope with network alignment and comparison tasks. The idea is to represent graphs as distributions and optimize for distribution matching via minimizing the cost of transporting one distribution to another [2]. EMD [19] performs OT on eigenvector-based node embeddings for alignment. GOT$_1$ [14] and fGOT [13] use Gaussian distribution w.r.t. different graph kernels for graph representation, and apply stochastic gradient descent to find the solution. GOT$_2$ [2] utilizes WD and GWD to address both node and edge correspondence. S-GWL [31] aligns two networks based on the GWD discrepancy. More recently, GraphOTC [20] performs OT on stationary Markov chains for network alignment and comparison. GOAT [25] attempts to handle the scalability issue by taking advantage of the Sinkhorn algorithm [6]. Though great progress has been made, most existing OT-based methods demand repeated computationally expensive loss calculations and bear relatively poor scalability. Besides, existing OT-based methods do not fully make use of topology information and hardly address the alignment consistency principle. These limitations collectively lead to relatively poor effectiveness and scalability.

7 CONCLUSION

In this paper, we study the semi-supervised network alignment problem from the view of optimal transport (OT). The OT framework provides a better distance measure (WD) capturing the underlying structure of graph data, compared with the Frobenius-like distance or ranking-based loss behind the existing consistency and embedding-based methods. To overcome the effectiveness and scalability issues of existing OT-based methods, we introduce a position-aware transport cost to capture topology information and consistency regularization to address the alignment consistency principle. We further decompose the regularized OT problem into a series of convex subproblems that can be efficiently solved by a scalable algorithm named PARROT based on the constrained proximal point method, with guaranteed convergence. Extensive experiments show that PARROT significantly outperforms the state-of-the-art. Specifically, the proposed PARROT is up to 3% - 13% better than the best competitor among consistency and embedding-based methods and runs 100×+ faster than existing OT-based methods.

ACKNOWLEDGEMENT

The ZZ and HT are partially supported by NSF (1947135, 2134079 and 1939725), DARPA (HR001121C0165), NIFA (2020-67021-32799) and ARO (W911NF2110088).

REFERENCES

[1] Xuezhi Cao and Yong Yu. 2017. Joint User Modeling Across Aligned Heterogeneous Sites Using Neural Networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 799–815.

[2] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. 2020. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*. PMLR, 1542–1553.

[3] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2016. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954* (2016).

[4] Xiyuan Chen, Mark Heimann, Fatemeh Vahedian, and Danai Koutra. 2020. Cone-align: Consistent network alignment with proximity-preserving node embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1985–1988.

[5] Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. 2019. Cross-network embedding for multi-network alignment. In *The world wide web conference*. 273–284.

[6] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013), 2292–2300.

[7] Boxin Du and Hanghang Tong. 2018. Fasten: Fast sylvester equation solver for graph mining. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1339–1347.

[8] Boxin Du, Si Zhang, Yuchen Yan, and Hanghang Tong. 2021. New frontiers of multi-network mining: Recent developments and future trend. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4038–4039.

[9] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. 2018. Regal: Representation learning-based graph alignment. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 117–126.

[10] Huiting Hong, Xin Li, Yuangang Pan, and Ivor Tsang. 2020. Domain-adversarial Network Alignment. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[11] Danai Koutra, Hanghang Tong, and David Lubensky. 2013. Big-align: Fast bipartite graph alignment. In *2013 IEEE 13th international conference on data mining*. IEEE, 389–398.

[12] Li Liu, William K Cheung, Xin Li, and Lejian Liao. 2016. Aligning Users across Social Networks Using Network Embedding.. In *Ijcai*. 1774–1780.

[13] Hermina Petric Maretic, Mireille El Gheche, Giovanni Chierchia, and Pascal Frossard. 2022. FGOT: Graph distances based on filters and optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7710–7718.

[14] Hermina Petric Maretic, Mireille EL Gheche, Giovanni Chierchia, and Pascal Frossard. 2019. GOT: An optimal transport framework for graph comparison. *arXiv preprint arXiv:1906.02085* (2019).

[15] Hermina Petric Maretic, Mireille El Gheche, Matthias Minder, Giovanni Chierchia, and Pascal Frossard. 2020. Wasserstein-based graph alignment. *arXiv preprint arXiv:2003.06048* (2020).

[16] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. 2018. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665* (2018).

[17] Gaspard Monge. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris* (1781).

[18] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. 2014. Inside the atoms: ranking on a network of networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1356–1365.

[19] Giannis Nikolentzos, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. Matching node embeddings for graph similarity. In *Thirty-first AAAI conference on artificial intelligence*.

[20] Kevin O'Connor, Bongsoo Yi, Kevin McGoff, and Andrew B Nobel. 2021. Graph Optimal Transport with Transition Couplings of Random Walks. *arXiv preprint arXiv:2106.07106* (2021).

[21] Shichao Pei, Lu Yu, Guoxian Yu, and Xiangliang Zhang. 2022. Graph Alignment with Noisy Supervision. In *Proceedings of the ACM Web Conference 2022*. 1104–1114.

[22] Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* 11, 5-6 (2019), 355–607.

[23] Gabriel Peyré, Marco Cuturi, and Justin Solomon. 2016. Gromov-Wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*. PMLR, 2664–2672.

[24] Yousef Saad. 2003. *Iterative methods for sparse linear systems*. SIAM.

[25] Ali Saad-Eldin, Benjamin D Pedigo, Carey E Priebe, and Joshua T Vogelstein. 2021. Graph Matching via Optimal Transport. *arXiv preprint arXiv:2111.05366* (2021).

[26] Rohit Singh, Jinbo Xu, and Bonnie Berger. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences* 105, 35 (2008), 12763–12768.

[27] Derek Tam, Nicholas Monath, Ari Kobren, Aaron Traylor, Rajarshi Das, and Andrew McCallum. 2019. Optimal transport-based alignment of learned character representations for string similarity. *arXiv preprint arXiv:1907.10165* (2019).

[28] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

[29] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*. IEEE, 613–622.

[30] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. 2020. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in Artificial Intelligence*. PMLR, 433–453.

[31] Hongteng Xu, Dixin Luo, and Lawrence Carin. 2019. Scalable Gromov-Wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems* 32 (2019), 3052–3062.

[32] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. 2019. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*. PMLR, 6932–6941.

[33] Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. 2021. Dynamic knowledge graph alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4564–4572.

[34] Yuchen Yan, Si Zhang, and Hanghang Tong. 2021. BRIGHT: A Bridging Algorithm for Network Alignment. In *Proceedings of the Web Conference 2021*. 3907–3917.

[35] Yuchen Yan, Qinghai Zhou, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. 2022. Dissecting cross-layer dependency inference on multi-layered interdependent networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2341–2351.

[36] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*. PMLR, 40–48.

[37] Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware graph neural networks. In *International Conference on Machine Learning*. PMLR, 7134–7143.

[38] Jiawei Zhang and S Yu Philip. 2015. Integrated anchor and social link predictions across social networks. In *Twenty-fourth international joint conference on artificial intelligence*.

[39] Jiawei Zhang and S Yu Philip. 2015. Multiple anonymized social networks alignment. In *2015 IEEE International Conference on Data Mining*. IEEE, 599–608.

[40] Si Zhang and Hanghang Tong. 2016. Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1345–1354.

[41] Si Zhang, Hanghang Tong, Long Jin, Yinglong Xia, and Yunsong Guo. 2021. Balancing Consistency and Disparity in Network Alignment. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2212–2222.

[42] Si Zhang, Hanghang Tong, Ross Maciejewski, and Tina Eliassi-Rad. 2019. Multilevel network alignment. In *The World Wide Web Conference*. 2344–2354.

[43] Si Zhang, Hanghang Tong, Jie Tang, Jiejun Xu, and Wei Fan. 2020. Incomplete network alignment: Problem definitions and fast solutions. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 4 (2020), 1–26.

[44] Si Zhang, Hanghang Tong, Yinglong Xia, Liang Xiong, and Jiejun Xu. 2020. Nettrans: Neural cross-network transformation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 986–996.

[45] Si Zhang, Hanghang Tong, Jiejun Xu, Yifan Hu, and Ross Maciejewski. 2019. Origin: Non-rigid network alignment. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 998–1007.

[46] Qinghai Zhou, Liangyue Li, Xintao Wu, Nan Cao, Lei Ying, and Hanghang Tong. 2021. Attent: Active attributed network alignment. In *Proceedings of the Web Conference 2021*. 3896–3906.

[47] Yang Zhou, Zeru Zhang, Sixing Wu, Victor Sheng, Xiaoying Han, Zijie Zhang, and Ruoming Jin. 2021. Robust network alignment via attack signal scaling and adversarial perturbation elimination. In *Proceedings of the Web Conference 2021*. 3884–3895.

A PROOF

A.1 Proof of Proposition 1

Proposition. **Convergence of RWR on separated graphs.** *With the error tolerance ϵ, $T = \log_{1-\beta}\epsilon$ iterations guarantee $\|\mathbf{R}_1^* - \mathbf{R}_1^{(T)}\|_\infty \leq \epsilon$ and $\|\mathbf{R}_2^* - \mathbf{R}_2^{(T)}\|_\infty \leq \epsilon$.*

Proof. RWR on separated graphs aims to find the fixed point solution to Eq. (2), or in the matrix form, $\mathbf{R}_1 = (1-\beta)\mathbf{W}_1\mathbf{R}_1 + \beta\mathbf{e}_1$. By denoting the initial solution as $\mathbf{R}_1^{(0)}$, fixed point solution as $\mathbf{R}_1^*$, and solution at the t-th iteration as $\mathbf{R}_1^{(t)}$, we have:

$$\|\mathbf{R}_1^* - \mathbf{R}_1^{(t)}\|_\infty = \left\|(1-\beta)\mathbf{W}_1\mathbf{R}_1^* - (1-\beta)\mathbf{W}_1\mathbf{R}_1^{(t-1)}\right\|_\infty$$

Since $\mathbf{W}_1$ is the transpose of the row normalized adjacency matrix with $\|\mathbf{W}_1\|_\infty \leq 1$, we have:

$$\begin{aligned}\|\mathbf{R}_1^* - \mathbf{R}_1^{(t)}\|_\infty &\leq (1-\beta)\|\mathbf{R}_1^* - \mathbf{R}_1^{(t-1)}\|_\infty \\ &\leq (1-\beta)^t\|\mathbf{R}_1^* - \mathbf{R}_1^{(0)}\|_\infty\end{aligned}$$

Due to $\|\mathbf{R}_1^{(0)}\|_\infty \leq 1$ and $\|\mathbf{R}_1^*\|_\infty \leq 1$, we have

$$\|\mathbf{R}_1^* - \mathbf{R}_1^{(t)}\|_\infty \leq (1-\beta)^t$$

Therefore, $T = \log_{1-\beta}\epsilon$ iterations guarantee $\|\mathbf{R}_1^* - \mathbf{R}_1^{(T)}\| \leq \epsilon$. Similar proof can be derived for $\mathbf{R}_2$. We omitted here for brevity. □

A.2 Proof of Proposition 2

Proposition. **Convergence of RWR on the product graph.** *With the error tolerance ϵ, $T = \log_{(1-\beta)\gamma}[(1-\gamma)\epsilon]$ iterations guarantee $\|\mathbf{C}^*_{cross} - \mathbf{C}^{(T)}_{cross}\|_\infty \leq \epsilon$.*

Proof. Similar to the proof of Proposition 1, with the initial solution $\mathbf{C}^{(0)}_{\text{cross}}$, the fixed point solution $\mathbf{C}^*_{\text{cross}}$ and the solution at the t-th iteration as $\mathbf{C}^{(t)}_{\text{cross}}$, we have

$$\|\mathbf{C}^*_{\text{cross}} - \mathbf{C}^{(t)}_{\text{cross}}\|_\infty \leq [(1-\beta)\gamma]^t\|\mathbf{C}^*_{\text{cross}} - \mathbf{C}^{(0)}_{\text{cross}}\|_\infty$$

Since we adopt the exponential of the negative cosine similarity to calculate $\mathbf{C}_{\text{node}}$ with its infinite norm less than or equal to 1, we have

$$\mathbf{C}^*_{\text{cross}} = \mathbb{E}_\mathbf{W}\sum_{i=1}^{\infty}\gamma^{i-1}\mathbf{C}_{\text{node}}(s_i) \leq \mathbb{E}_\mathbf{W}\sum_{i=1}^{\infty}\gamma^{i-1}\|\mathbf{C}_{\text{node}}\|_\infty \leq \frac{1}{1-\gamma}.$$

By combining the above two inequalities, we have

$$\|\mathbf{C}^*_{\text{cross}} - \mathbf{C}^{(t)}_{\text{cross}}\|_\infty \leq \frac{(1-\beta)^t\gamma^t}{1-\gamma}$$

Therefore, $T = \log_{(1-\beta)\gamma}[(1-\gamma)\epsilon]$ iterations guarantee $\|\mathbf{C}^*_{\text{cross}} - \mathbf{C}^{(T)}_{\text{cross}}\|_\infty \leq \epsilon$. □

A.3 Proof of Lemma 1

Lemma. **Strict convexity of Eq.** (16). *The subproblem in Eq.* (16) *is strictly convex.*

Proof. Since the $\mathrm{C}^{(t)}$ is a fixed matrix, the second derivative of the inner product $\langle \mathrm{C}^{(t)}, \mathrm{S}\rangle$ w.r.t. S equals to zero. Therefore, we only need to focus on the convexity of $\langle \log \mathrm{S}, \mathrm{S}\rangle$.

For $\langle \log \mathrm{S}, \mathrm{S}\rangle$, its second derivative w.r.t. S can be computed as follows.

$$\nabla^2_\mathrm{S}\langle \log \mathrm{S}, \mathrm{S}\rangle = \operatorname{diag}\left(\frac{1}{\operatorname{vec}(\mathrm{S})}\right)$$

Since all entries of the alignment score S are positive, the second derivative $\nabla^2_\mathrm{S}\langle \log \mathrm{S}, \mathrm{S}\rangle$ is a positive definite matrix and $\langle \log \mathrm{S}, \mathrm{S}\rangle$ is strictly convex. Therefore, the consistency-regularized OT problem is strictly convex. □

A.4 Proof of Lemma 2

Lemma. **Convergence of proximal point method [30].** *Given a convex objective function $f(\mathrm{S})$, the solution sequence $\{f(\mathrm{S}^{(t)})\}$ generated by the proximal point method is non-increasing.*

Proof. The first-order optimality condition of Eq. (12) can be written as:

$$\left(\mathrm{S} - \mathrm{S}^{(t+1)}\right)^\top \left[\nabla f(\mathrm{S}^{(t+1)}) + \lambda\nabla L_p^{(t)}\right] \geq \mathbf{0}, \forall \mathrm{S} \in \mathcal{S}$$

The convexity of f can be written as:

$$\left(\mathrm{S} - \mathrm{S}^{(t+1)}\right)^\top \nabla f(\mathrm{S}^{(t+1)}) \leq f(\mathrm{S}) - f(\mathrm{S}^{(t+1)}), \forall \mathrm{S} \in \mathcal{S}$$

Combine the above two inequality and set $\mathrm{S} = \mathrm{S}^{(t)}$, we have:

$$\begin{aligned}f(\mathrm{S}^{(t+1)}) &\leq f(\mathrm{S}^{(t)}) + \lambda\left(\mathrm{S}^{(t)} - \mathrm{S}^{(t+1)}\right)^\top\left(\mathbf{1}_{n_1\times n_2} + \log\frac{\mathrm{S}^{(\mathrm{t})}}{\mathrm{S}^{(t)}}\right) \\ &= f(\mathrm{S}^{(t)})\end{aligned}$$

The last equation is based on the fact that $\mathrm{S}^{(t)}, \mathrm{S}^{(t+1)} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$, so that $\left(\mathrm{S}^{(t)} - \mathrm{S}^{(t+1)}\right)^\top \mathbf{1}_{n_1\times n_2} = \mathbf{0}_{n_1\times n_2}$. Therefore, the solution sequence $\{f(\mathrm{S}^{(t)})\}$ is non-increasing, i.e. $f(\mathrm{S}^{(t+1)}) \leq f(\mathrm{S}^{(t)})$, and converges as $\lim_{T\to\infty} f(\mathrm{S}^{(T)}) \leq f(\mathrm{S}), \forall \mathrm{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$. □

A.5 Proof of Theorem 1

Theorem. **Convergence of PARROT.** *The solution sequence $\{f_t(\mathrm{S}^{(t)})\}$ generated by PARROT is non-increasing and converges.*

Proof. The constrained proximal point method can be divided into two cases: (1) $f_t(\mathrm{S})$ stays unchanged when $\langle \mathrm{C}^{(t)}, \mathrm{S}^{(t)}\rangle > f_{t-1}(\mathrm{S}^{(t)})$, and (2) $f_t(\mathrm{S}) = \langle \mathrm{C}^{(t)}, \mathrm{S}\rangle$ when $\langle \mathrm{C}^{(t)}, \mathrm{S}^{(t)}\rangle \leq f_{t-1}(\mathrm{S}^{(t)})$.

For case (1), consecutive iterations t and $t+1$ are optimizing the same objective function based on proximal point method. According to Lemma 1 and 2, we have:

$$f_t(\mathrm{S}^{(t)}) = f_{t-1}(\mathrm{S}^{(t)}) \leq f_{t-1}(\mathrm{S}^{(t-1)})$$

For case (2), according to the constraint, we have:

$$f_t(\mathrm{S}^{(t)}) = \langle \mathrm{C}^{(t)}, \mathrm{S}^{(t)}\rangle \leq f_{t-1}(\mathrm{S}^{(t)})$$

Further based on Lemma 1 and 2 that $f_{t-1}(\mathrm{S}^{(t)}) \leq f_{t-1}(\mathrm{S}^{(t-1)})$, we conclude that $f_t(\mathrm{S}^{(t)}) \leq f_{t-1}(\mathrm{S}^{(t-1)})$ in case (2).

Therefore, the solution sequence $\{f_t(\mathrm{S}^{(t)})\}$ given by the constraint proximal point method is non-increasing and converges as $\lim_{T\to\infty} f_T(\mathrm{S}^{(T)}) \leq f_t(\mathrm{S}^{(t)}), \forall t = 0, 1, 2, \ldots.$ □

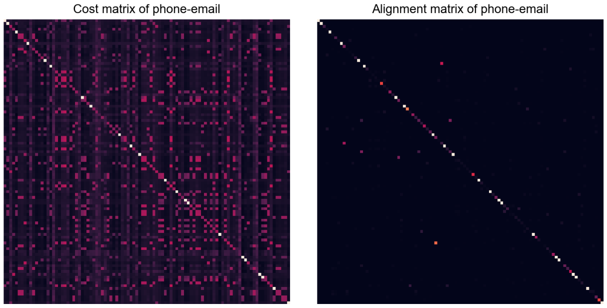


Figure 9: Visualization of the OT mapping on the Phone-Email dataset. Darker the pixel, lower the value.

A.6 Proof of Theorem 2

Theorem. **Complexity of PARROT.** *The overall time complexity of PARROT is $O(Tmn + TLn^2)$, where T is the number of outer iterations and L is the number of inner iterations in PARROT.*

Proof. Since $\mathbf{W}_1$ and $\mathbf{W}_2$ are sparse matrices with $O(m)$ non-zero entries, the complexity of RWR calculation in Eq. (2) and Eq. (4) are $O(mn)$.[2] For the constrained proximal point method in Algorithm 2, since $\mathbf{C}_1, \mathbf{C}_2$ and $\mathbf{W}_1, \mathbf{W}_2$ are sparse matrices with $O(m)$ non-zero entries the complexity for calculating $\mathbf{L}^{(t)}$ and $\mathbf{C}^{(t)}$ are $O(mn)$ per outer iteration. Solving each subproblem by Eq. (17) is $O(n^2)$ per inner iteration. Therefore, with T outer iterations and L inner iterations, the overall time complexity of PARROT is $O(Tmn + TLn^2)$. □

B VISUALIZATION OF OT MAPPINGS

We provide the visualization of the learned OT mapping between the first 100 nodes in the Phone-Email dataset in Figure 9. The groundtruth alignment is the diagonal matrix (i.e., the i-th node in $\mathcal{G}_1$ is aligned to the i-th node in $\mathcal{G}_2$). We reverse the values in the cost matrix so that higher the value (brighter the pixel), the more likely two nodes are aligned. It is shown that the cost matrix is a noisy matrix of the groundtruth alignment, while PARROT produces a denoised alignment matrix that is consistent with the groundtruth alignment.

C REPRODUCIBILITY

Dataset Descriptions. The datasets used in our experiments include:

- Foursquare-Twitter [38]: Two online social networks with nodes as users and edges as friendships. Foursquare network includes 5,313 nodes and 54,233 edges. Twitter network includes 5,120 nodes and 130,575 edges. Both networks are plain networks. There are 1,609 common users across two networks.
- ACM-DBLP [28]: Two co-authorship networks of the ACM Digital library and DBLP bibliography. Nodes represent authors and an edge exists between two authors if they are co-author for at least one publication. ACM co-author network includes 9,916 nodes and 44,808 edges. DBLP co-author network includes 9,872 nodes and 39,561 edges. Both networks are attributed networks where attributes indicate the number of papers that are published in different venues by that author. There are 6,325 common authors across two networks. In our experiment, we use ACM(A)/DBLP(A) to denote the dataset with node attributes and ACM/DBLP to denote the dataset without node attributes.

Table 5: Dataset Summary.

Scenarios	Networks	# nodes	# edges	# attributes
Plain	Foursquare	5,313	54,233	0
	Twitter	5,120	130,575	0
	ACM	9,872	39,561	0
	DBLP	9,916	44,808	0
	Phone	1,000	41,191	0
	Email	1,003	4,627	0
Attributed	ACM(A)	9,872	39,561	17
	DBLP(A)	9,916	44,808	17
	Cora1	2,708	6,334	1,433
	Cora2	2,708	4,542	1,433

- Phone-Email [43]: Two communication networks among people via phone or email. Nodes represent people and an edge exists between two people if they communicate via phone or email at least once. Phone network includes 1,000 nodes and 41,191 edges. Email network includes 1,003 nodes and 4,627 edges. Both networks are plain networks. There are 1,000 common people across two networks.
- Cora1-Cora2 [36]: A citation network with nodes as publications and edges as citations among publications. Cora1 and Cora2 are two permuted networks with noise added. More specifically, 10% edges are first added to Cora1 and 15% edges are removed from Cora2 thereafter. Cora1 includes 2,708 nodes and 6,334 edges. Cora2 includes 2,708 nodes and 4,542 edges. Both networks are attributed networks where node attributes are binary feature vector represented by bag-of-words. There are 2,708 common publications across two networks.

Dataset statistics are shown in Table 5. In our experiments, we use 20% ground-truth as the prior knowledge/training data and test on the rest of the ground-truth.

Machine configuration and code. The proposed method is implemented in MATLAB. We use an Apple M1 chip with 16 GB RAM to run PARROT, IsoRank, FINAL, and OT-based methods. We use NVIDIA Tesla V100 SXM2 as GPU for embedding-based methods.

Hyperparameters settings. An overview of hyperparameters settings for our experiments is shown in Table 6.

Table 6: Hyperparameters settings

Dataset	λ_e	λ_n	λ_a	λ_p	α	β	γ
Foursquare-Twitter	3e-6	5e-3	5e-4	1e-3	0.5	0.15	0.8
ACM-DBLP	1e-5	4e-4	1e-2	1e-3	0.1	0.5	0.9
Phone-Email	2e-5	5e-3	5e-4	5e-4	0.5	0.15	0.7
ACM(A)-DBLP(A)	5e-5	1e-2	1e-2	1e-1	0.1	0.15	0.2
Cora1-Cora2	1e-6	1e-2	2e-3	2e-3	0.5	0.3	0.2

[2] Using the recent advancement of faster Sylvester Equation solver, such as implicit Krylov subspace methods [7], it is possible to reduce this cost to be linear in m.