

SAlign: A Graph Neural Attention Framework for Aligning Structurally Heterogeneous Networks

Shruti Saxena

*Indian Institute of Technology Patna
Patna, 801106, India*

SHRUTI_2021CS31@IITP.AC.IN

Joydeep Chandra

*Indian Institute of Technology Patna
Patna, 801106, India*

JOYDEEP@IITP.AC.IN

Abstract

Network alignment techniques that map the same entities across multiple networks assume that the mapping nodes in two different networks have similar attributes and neighborhood proximity. However, real-world networks often violate such assumptions, having diverse attributes and structural properties. Node mapping across such *structurally heterogeneous* networks remains a challenge. Although capturing the nodes' entire neighborhood (in low-dimensional embeddings) may help deal with these characteristic differences, the issue of over-smoothing in the representations that come from higher-order learning still remains a major problem. To address the above concerns, we propose *SAlign*: a supervised graph neural attention framework for aligning structurally heterogeneous networks that learns the correlation of structural properties of mapping nodes using a set of labeled (mapped) anchor nodes. *SAlign* incorporates nodes' graphlet information with a novel structure-aware cross-network attention mechanism that transfers the required higher-order structure information across networks. The information exchanged across networks helps in enhancing the expressivity of the graph neural network, thereby handling any potential over-smoothing problem. Extensive experiments on three real datasets demonstrate that *SAlign* consistently outperforms the state-of-the-art network alignment methods by at least 1.3-8% in terms of accuracy score. The code is available at <https://github.com/shruti400/SAlign> for reproducibility.

1. Introduction

In the Big-data era, dealing with and analyzing large-scale networks such as social networks, co-authorship networks, and protein interaction networks has gained considerable interest. Multiple networks of the same domain often exist, and jointly studying their co-related information solves the sparsity and data insufficiency issue of analyzing from a single network (Man et al., 2016). Network alignment, which aims to map the same entities across multiple networks, plays a crucial role in distilling entity information. The collective information from multiple networks has been crucial for downstream tasks like cross-site friend recommendation, product recommendation, fraud detection, and revealing new interactional patterns in protein networks (Bayati et al., 2013).

Finding pairwise node correspondences across 2 different networks follows two underlying assumptions: (1) the attributes of mapped nodes are similar, which we refer to as the attribute consistency constraint, and (2) the mapped nodes exhibit a similar neighborhood structure called the structural consistency constraint (Trung et al., 2020). Several

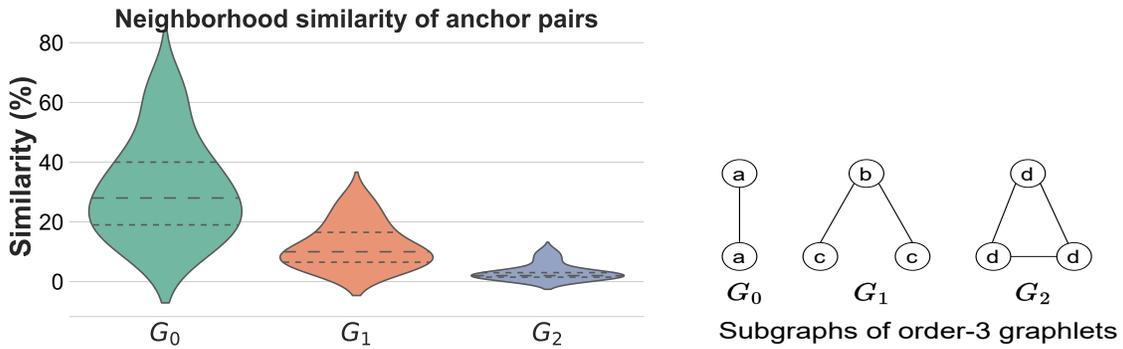


Figure 1: Neighborhood similarity of some anchor pairs across the Douban dataset. It illustrates the similarity percentage of anchors having similar graphlet structures.

existing methods often strictly adhere to these constraints by attempting to jointly learn the node structure similarities across the networks thereby preserving more comprehensive information about the node correspondences (Xu et al., 2019). However, such structural consistency assumption is often violated in real-world scenarios. From a global aspect, the networks can have varying degrees of sparsity, distribution and path lengths, and hence joint learning strongly based on such structural similarities/dissimilarities may not produce satisfactory results.

To reinforce the above statement, we study the structural co-relation of some mapped node pairs (anchor pairs) from the Chinese social networks - Douban Online and Douban Offline (Trung et al., 2020). We compare the subgraph interactional patterns of these anchors’ neighborhoods via graphlet information. We only examine graphlets of order three that capture up to three path lengths of the neighborhood. In Figure 1, G_0 , G_1 , and G_2 depict the subgraphs corresponding to order-3 graphlets. We then find the percent overlap in the count of these graphlet substructures, based on which we plot the anchors’ similarity distribution as shown in Figure 1. The area of the violin plot represents the relative number of anchors having the corresponding neighborhood similarity percentage constituted by each of the subgraphs, G_0 , G_1 , and G_2 , respectively. We observe that even though these are pre-defined anchor pairs, their structural similarity differs significantly. Most anchor pairs have less than 40% neighborhood similarity majorly contributed by the similarity in their single path length neighborhood i.e. G_0 . We see that the same node can have different interactional patterns in and across multiple networks, which we refer to as *structural heterogeneity*.

In this paper, we propose SAlign, a method to align networks in the presence of structural heterogeneity. We believe that jointly learning node embeddings of the two networks while being aware of their structural correlates may capture more comprehensive information on the node similarities. For example, anchor nodes in one network may be part of several closed triangles, while the corresponding neighborhood in the other network may be dominated by higher-length substructures (which leads to structural heterogeneity as inferred by Figure 1). While capturing these structural correlations would help to increase performance, existing strategies are not very effective at dealing with these higher-order substructures. State-of-the-art Graph Neural Network (GNN) based techniques suffer from

over-smoothing problem (Chen et al., 2020), whereby the node representations tend to be similar when such higher order structures are captured.

Specifically, we propose SAlign, a supervised graph neural network representation learning approach that jointly learns from the two networks hierarchically - one at the node level to capture the intra-network local interactions and one at the graph level to capture the inter-network interactions. We identify these interactions by the structural characteristics of the networks captured through graphlets, which represent rich structural features and play a decisive role in the connection of nodes. The novelty of SAlign is twofold. Along with capturing a node’s local neighborhood structure in its representation, it also proposes an attentive aggregation of the structural correlations between the subgraphs surrounding the node’s corresponding anchor. The attention mechanism focuses more on the discriminative subgraphs within and across networks, making it less influenced by other noisy neighborhoods that do not significantly contribute to finding the anchor pairs. Furthermore, attentively aggregating the hierarchical structural features enhances the node expressivity, thereby overcoming the over-smoothing problem. While some approaches have also used graphlet information (Almulhim et al., 2019) however, they treat them as node attributes, as opposed to our method, which uses the graphlets as structural features to determine their correlation across the networks. Several other existing methods that have also used higher-order structures (Zhang et al., 2019; Qiu et al., 2021; Xia et al., 2021), to generate the node representations neither make any attempt to learn the embeddings jointly nor consider capturing the structural variations across the network.

The main contributions of our work are summarized as follows:

1. To the best of our knowledge, we are the first to leverage the correlation of higher-order substructures for encoding joint relationships for aligning two structurally heterogeneous networks.
2. We facilitate intra-network and inter-network propagation by learning node-level and graph-level structural features through a hierarchical graphlet-based attention mechanism. The attentive aggregation of these structural features enhances node expressivity while addressing the over-smoothing issue.
3. We empirically validate the effectiveness of SAlign on three real-world datasets and compare it to eight state-of-the-art baselines in terms of $Acc@1$, $Acc@10$, MAP , and AUC . We further perform case studies to validate the effectiveness of the hierarchical attention mechanism of SAlign in dealing with structural heterogeneity and over-smoothing issues.

The organization of the paper is as follows. In Section 2, we briefly outline the related works. In Section 3, we discuss some preliminaries required to understand the problem. In Section 4, we describe the SAlign framework in detail, followed by the experimental setup in Section 5. We report and analyze the experimental results in Section 6. Finally, we conclude and present the future work in Section 7.

2. Related Works

This section elaborates on the different network embedding approaches followed by the state-of-the-art network alignment methods.

2.1 Network Embedding

In recent years, deep neural embedding methods to exploit node dependence and develop node representations have generated promising results, with GNN models receiving the most attention (Scarselli et al., 2008). Based on different aggregation functions to aggregate node neighborhood information, GCNs (Kipf & Welling, 2016) and GATs (Velickovic et al., 2018) are the most prevalent. There are several other proposed variants of GNNs; however, they are all confined to only capturing low-order graph structures around every node (Li et al., 2018). GNN models have recently incorporated graphlets (Tu et al., 2018; Feng & Chen, 2020), motifs (Zhao et al., 2018; Sankar et al., 2020; Subramonian, 2021), and anonymous walks (Long et al., 2020; Jin et al., 2020) to leverage higher-order graph structures. gl-DCNN (Tu et al., 2018) concatenates node graphlet information and node features for input to the diffusion-convolutional neural networks. GraphLSP (Jin et al., 2020) captures the complex structural patterns via random anonymous walks. GraphStone (Long et al., 2020) additionally builds topic models upon graphs to capture the distributional differences over the local structural patterns. HM-Modularity (Huang et al., 2018) and HCEMM (Huang et al., 2023) propose a concept of harmonic motif to capture a slightly higher order connection structure across various views for multi-view network community detection. However, the assumption of the same node set across every view does not hold true when aligning two networks. Motif-based GNN models use one (Zhao et al., 2018), or more (Sankar et al., 2017; Chen et al., 2023) motif-based adjacency matrices to perform message passing. Our work differs from the discussed approaches in several vital points. All of the above methods primarily focus on a single network, and directly adopting them in the network alignment task is insufficient. Although a recent approach introduces a GCN model for cross-network learning (Jiang, 2021), it fails to capture the higher-order complex structures. Hence, we are the first to propose a higher-order network embedding method that utilizes novel graphlet vector-based hierarchical attention to model multiple aligned networks. We design a novel aggregation scheme to transfer more useful information across networks and overcome network characteristics contradictions.

2.2 Network Alignment

Network alignment is an emerging active area. Using network structures, node attributes, and edge attributes to integrate complete network information has been beneficial for network alignment. With neural embedding methods gaining popularity, researchers have improved and applied network embedding methods in a multi-network environment. Methods like PALE (Man et al., 2016), and Deeplink (Zhou et al., 2018) learn the node embeddings separately and then learn a mapping function to map the embeddings based on the labeled anchor information. While PALE uses LINE (Tang et al., 2015) or DeepWalk (Perozzi et al., 2014), Deeplink uses random walk to produce the node embeddings. More recent approaches use an end-to-end learning framework to jointly train the embeddings of the

two networks by optimizing a loss function over the dataset (Nguyen et al., 2021; Liu et al., 2019b). These alignment approaches utilize deep learning-based node embedding approaches, like adversarial-based training (Hong et al., 2020; Chen et al., 2019), graph convolutional network (Saxena et al., 2022; Trung et al., 2020; Cheng et al., 2019) for building node representations of both networks using common trainable parameters. Although these alignment approaches have shown promising results, they lack to jointly learn and transfer more useful complementary information across the partially aligned networks. To overcome this, CENALP (Du et al., 2019), and BRIGHT (Yan et al., 2021) capture the structural properties from separate graphs using a cross-network embedding method employing random walks. CrossMNA (Chu et al., 2019) expresses network differences through layer vectors and uses a linear transformation between nodes across networks. CCALP (Lan et al., 2021), in addition, models community-level inter-network relationships. A recent adversarial learning-based approach, HackGAN (Yang et al., 2022), captures the local and global node features by solving the Wasserstein Procrustes problem. However, these methods cannot distinguish nodes with isomorphic low-order graph structures from different higher-order structures. Moreover, they do not deal with the network characteristic differences focused on by our proposed method.

3. Preliminaries

This section covers some fundamental concepts required for comprehending the paper. In line with the majority of literature, we focus on aligning two unweighted, undirected, and attributed networks.

3.1 Notations and Problem Formulation

Given two partially aligned networks, a source network $G_s = (V_s, E_s, X_s, \hat{A}_s)$ and a target network $G_t = (V_t, E_t, X_t, \hat{A}_t)$, network alignment is the task of finding node correspondences between them. Here $V = \{v_1, v_2, \dots, v_n\}$ is the set of n nodes, E is the set of edges between nodes, $X = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n]^T \in \mathbb{R}^{n \times m}$ is the node attribute matrix, and \hat{A} is the adjacency matrix with self loops. Moreover, $A = \{(u_i, v_j) \mid u_i \in V_s, v_j \in V_t\}$ is the set of pre-known anchor links that are one-to-one mapped node pairs between G_s and G_t . We formulate network alignment as the calculation of an alignment matrix P^* , where the $(u, v)^{th}$ element, $P^*[u, v]$, represents the degree of similarity between the nodes $u \in V_s$ and $v \in V_t$, and thus provides a measure of the probability of them being anchor nodes.

3.2 Graphlets and Orbits

Graphlets: Graphlets are small, connected, non-isomorphic induced subgraphs representing connected patterns in a network between k nodes in a graph (Ribeiro et al., 2021). For example, in Figure 1, G_0 , G_1 , and G_2 depicts the subgraphs corresponding to order-3 graphlets.

Orbits: The nodes of every graphlet are partitioned into a set of automorphism groups called orbits (Ribeiro et al., 2021). Two nodes belong to the same orbit if they map to each other in some isomorphic projection of the graphlet onto itself. In Figure 1, a , b, c and d are the node orbits of order-3 graphlets.

Graphlet degree vectors: The graphlet-degree vector (GDV) of a node is a feature vector specifying the number of times it occurs in each orbit (Ribeiro et al., 2021). Each position in the GDV corresponds to an orbit.

3.3 Graph Neural Networks

Graph Neural Networks (GNNs) compute a representation for node v by aggregating features from its neighborhood through a learnable aggregator function F^θ . Let $\mathcal{N}(i) = \{j : (i, j) \in E\}$ denote a set of neighbors of $i \in V$, general GNN message passing rule at layer ℓ for node i then consists of:

$$\vec{h}_i^{(\ell)} = F^\theta \left(\left\{ \vec{h}_j^{(\ell-1)}, v_j \in N(v_i) \cup \{v_i\} \right\} \right) \quad (1)$$

F^θ defines the message passing mechanism, and a variety of aggregator architectures like GCN (Kipf & Welling, 2016), GAT (Velickovic et al., 2018), and pooling can be generalised by Equation 1. $\vec{h}_j^{(\ell-1)}$ denotes the embedding vector of node v_j in the $(\ell - 1)$ -th layer of GNN.

4. Proposed Approach

This section introduces the SAlign framework in detail. Figure 2 illustrates the overall alignment architecture. We begin with a general overview of the model, then go over each component in detail.

4.1 Method Overview

We illustrate the overall alignment framework of SAlign in Figure 2. The key idea of SAlign is to train the source and target networks jointly to transmit complementary information among the known anchor links. The anchor node in a source network can show both similar as well as distinctive structural and attribute features as compared to the corresponding node in the target network. To precisely capture these variations, we exploit the deeper neighborhood structure of two nodes rather than solely depending on local structural information such as common neighbors. Hence the first step of SAlign is to extract the graphlet features of network nodes to capture the fine-grained features that also compose their global neighborhood structure. While OC-GAE (Feng & Chen, 2020) proposed concatenating these structural features with the node attributes, we argue that it may lead to information loss. Therefore we propose a novel representation generation strategy that allows for the selective gathering of information necessary to capture complex structures such as cliques or dense clusters that indicate strong connections. Furthermore, each neighbor (intra-network) does not contribute equally to generating a node’s representation. Similarly, the relevance of a node’s anchor from another network (inter-network) may vary for networks of diverse domains and topologies. Hence we propose a hierarchical attention mechanism for normalizing the intra-network and inter-network information propagation. The following sections discuss the graphlet feature extraction, SAlign’s aligned network embedding architecture, optimization, and alignment computation in detail.

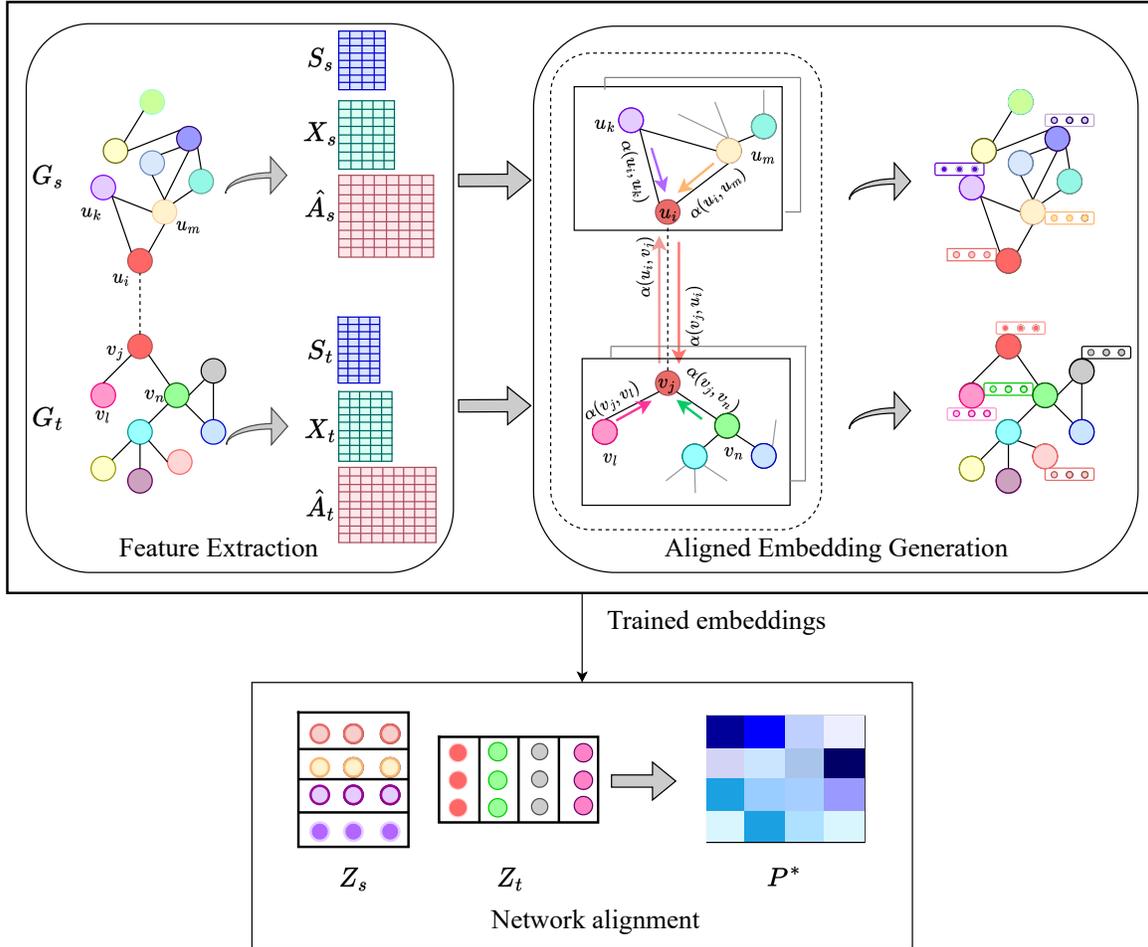
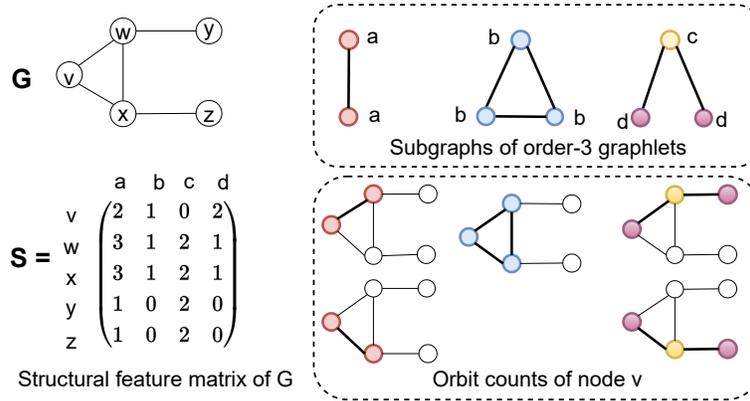


Figure 2: **SAlign model architecture.** The *Feature Extraction* module takes graphs G_s and G_t as input and extracts their corresponding feature matrices S . \hat{A} and X are input to the encoder initialization. The *Embedding Generation* module iteratively embeds the networks using the combined intra- and inter-network attention mechanism, optimizing over the loss \mathcal{L} . P^* is the final alignment matrix obtained from the trained embeddings.

4.2 Feature Extraction

Graphlets contain complex non-linear patterns that reflect the higher-order characteristics of graph nodes. In addition to the lower-order structural features (like node neighbors), capturing these higher-order features leads to a more informative and distinctive representation of network nodes. Hence, we build a structural feature matrix, $S = [\vec{s}_1, \vec{s}_2, \dots, \vec{s}_n]^T \in \mathbb{R}^{n \times o}$, to capture a network’s higher-order structures. Here \vec{s}_i is the GDV of node i , and $S[i, j]$ represents the number of times node i participates in the orbit j . Each position in the GDV corresponds to an orbit.

We take a sample graph and find its structural feature matrix for a better understanding. For example, in Figure 3, we only consider order-3 graphlet structures to calculate the structural feature matrix of graph G . a, b, c , and d represents the orbits of order-3

Figure 3: Structural feature matrix S obtained by considering graphlet orbits a, b, c , and d .Table 1: Graphlets of order- k

Graphlet order	Number of subgraphs	Number of orbits
2	1	1
3	3	4
4	9	15
5	30	73
6	142	480

subgraphs. In particular, if we see $S[v]$, each column entry represents the number of times node v participates in the orbits a, b, c , and d , respectively. We clearly illustrate that node v participates two times in the a orbit, one time in the b orbit, and two times in the d orbit. Similarly, we calculate for all the nodes in G .

We leverage the orca algorithm (Hočevar & Demšar, 2014) that constructs a system of equations to count the node orbits on graphs. In our experiments, we evaluate the performance by capturing structures up to 5-order graphlets with 73 orbits. Table 1 gives the subgraph counts and the orbits count for graphlet size of order- k . For example, $S \in \mathbb{R}^{n \times 73}$ when we consider graphlet of order-3.

4.3 Aligned Network Embedding Framework

We propose a hierarchical graph attention mechanism at the node and network levels that pay attention to different neighbor nodes and networks based on their higher-order patterns while building node representations. We introduce a more sophisticated aggregation of higher-order structural features along with the node-level attributes, rather than just concatenating them. The proposed attentive aggregation of information enhances the expressivity of the learned representations while avoiding the over-smoothing issue.

To fulfil the above discussed impacts, we devise the following aggregation function

$$\vec{h}_i^{(\ell)} = \sigma(\text{intra}(u_i) + \text{inter}(u_i)) + \vec{b}^{(\ell)} \quad (2)$$

The input to our first layer is the node feature matrix, $X \in \mathbb{R}^{n \times m}$. L layers are stacked to produce new set of node features, $Z^\ell = \{\vec{h}_1^\ell, \vec{h}_2^\ell, \dots, \vec{h}_n^\ell\} \in \mathbb{R}^{n \times m'}$. We now detail the two aggregation terms, one for generating intra-network embedding and other for inter-network embedding with their employed attention mechanisms.

Intra-network embedding: We derive the intra-network embedding vector of a node u_i by aggregating its neighbor embeddings. These intra-network neighbors can have varying importance, shown by their structural patterns. Moreover, we integrate the higher-order structural features with each neighbor’s node embeddings given as:

$$intra(u_i) = \sum_{u_k \in \mathcal{N}(u_i)} w(u_i, u_k) W_{(1)}^{(\ell)} \left(\vec{h}_k^{(\ell-1)} \odot g_k \right) \quad (3)$$

To model the selective aggregation of node structural features in the neighborhood and avoid the over-smoothing of representations, we have

$$g_i = \sigma \left(Q^{(\ell)} \vec{s}_i + r^{(\ell)} \right) \quad (4)$$

where $W_{(1)}^{(\ell)}$, $Q^{(\ell)}$ and $r^{(\ell)}$ are trainable parameters, \odot denotes element-wise multiplication and $w(u_i, u_k)$ is the attention exerted by a neighbor node u_k on node u_i , we discuss later below.

Inter-network embedding: We derive the inter-network embedding vector of node u_i by taking features from its corresponding anchor node v_j from the other network as

$$inter(u_i) = w(u_i, v_j) W_{(2)}^{(\ell)} \left(\vec{h}_j^{(\ell-1)} \odot g_j \right) \quad (5)$$

where $w(u_i, v_j)$ balances the information transfer across the two networks as discussed below.

Combined Attention mechanism: The intra-network attention co-efficient given to u_i from $u_k \in \mathcal{N}_{u_i}$ is given as

$$\alpha(u_i, u_k) = \sigma \left(a_{(1)}^T \left[W_{(1)} \vec{h}_i \| W_{(1)} \vec{h}_k \right] \right) \quad (6)$$

and similarly the inter-network attention co-efficient given to u_i from its anchor pair v_j in the other network is given by

$$\alpha(u_i, v_j) = \sigma \left(a_{(2)}^T \left[W_{(1)} \vec{h}_i \| W_{(2)} \vec{h}_j \right] \right) \quad (7)$$

where $.^T$ represents transposition and $\|$ is the concatenation operation. σ denotes the sigmoid function and $a_{(1)}^T, a_{(2)}^T$ are trainable weight vectors. Besides, to make these coefficients easily comparable across different nodes, we apply the softmax function as

$$w(u_i, u_k/v_j) = \text{softmax}_{u_k/v_j} (\alpha(u_i, u_k/v_j)) \quad (8)$$

$$= \frac{\exp(\alpha(u_i, u_k/v_j))}{\sum_{u_{k'} \in \mathcal{N}(u_i)} \exp(\alpha(u_i, u_{k'})) + \exp(\alpha(u_i, v_j))} \quad (9)$$

We further discovered that adopting multi-head attention, as inspired by (Velickovic et al., 2018), helped stabilize the learning process of node embeddings. Hence we employ K independent attention mechanisms to carry out this transformation and subsequently concatenate their embeddings to provide final nodes' representations given as

$$\vec{h}_i^{(\ell)} = \parallel_{k=1}^K \sigma(\text{intra}(u_i) + \text{inter}(u_i)) + \vec{b}^{(\ell)} \quad (10)$$

4.4 Model Learning and Alignment

For a node's embedding to be consistent with respect to its intra-network structure as well as its inter-network anchor connection, we optimize SAlign by minimizing the following loss functions jointly over both networks in a unified framework:

Consistency loss: The consistency loss determines if the resulting node embeddings are consistent with the intra-network topology given as:

$$\mathcal{L}_{\{s,t\}}^{cons} = \sum_{\ell \in [1 \dots L]} \left\| \hat{A} - \sigma \left(Z^{(\ell)} Z^{(\ell)T} \right) \right\|_F \quad (11)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The loss function is computed from the embeddings at all layers to capture a node's different neighborhood orders.

Alignment loss: We use the alignment loss to enhance the similarity between the true anchor links. We comprehensively leverage information transfer across the networks to learn the embeddings by aligning anchor nodes using a negative sampling strategy. We consider the true anchors, A , as positive anchor links and randomly sample negative anchor links, U , from the unmapped links across the networks. The aim is to increase the similarity between the true anchor links and dissimilarity between the sampled negative links. We define the node alignment loss function as:

$$\mathcal{L}^{align} = \frac{1}{|A|} \sum_{(u_i, v_j) \in A} \left(p(u_i, v_j) + \frac{1}{|U|} \sum_{(u_m, v_n) \in U} (1 - p(u_m, v_n)) \right) \quad (12)$$

where $p(u_i, v_j)$ gives a measure of similarity between nodes u_i and u_j as

$$p(u_i, v_j) = \sigma \left(\vec{h}_i^T \cdot \vec{h}_j \right) \quad (13)$$

The final loss function of the SAlign model is derived as:

$$\mathcal{L} = \mathcal{L}_s^{cons} + \mathcal{L}_t^{cons} + \beta \mathcal{L}^{align} \quad (14)$$

where β is a balancing factor that controls the information flow across networks.

After SAlign converges, the final step is to derive the alignment matrix P^* to infer all pairs of anchor nodes across G_s and G_t . P^* is calculated with the generated embeddings as :

$$P^* = \sigma \left(Z_s^T \cdot Z_t \right) \quad (15)$$

The $(u, v)^{th}$ entry in P^* signifies the similarity score between the corresponding nodes. Hence the highest value corresponding to a node u in the row $P^*(u)$ most likely represents the same node in the other network.

5. Experimental Details

In this section, we discuss the details of datasets and the baseline methods used for comparing the effectiveness of SALign. We subsequently discuss the metrics used for performance evaluation and the experimental setup used for implementation.

Datasets: We evaluate the performance of SALign on three public datasets used in recent state-of-the-art works, as summarized in Table 1.

- **Douban Online-Offline:** These are Chinese social networks (Trung et al., 2020; Yang et al., 2022), one connecting user friends online and the other connecting users based on offline co-occurrences in social events. It considers user locations as node attributes.
- **Allmovie-IMDB:** These are online movie guide service networks that connect two films if they have at least one actor in common (Trung et al., 2020; Saxena et al., 2022). It treats details like movie genre and cast as node attributes.
- **DBLP:** It is a network of co-authors of the computer science conferences (Zhang & Tong, 2016; Du et al., 2019). It treats the number of publications in each conference as node attributes. We randomly delete 10% of the edges and shuffle the attributes to generate the target dataset (Zhang & Tong, 2016).

We choose these datasets for their public availability and better reproducibility. Moreover, all these datasets have high structural heterogeneity, with the network pairs having a significant difference in the number of nodes, edges, and higher-order structures, especially the Douban dataset.

Baseline methods: We compare SALign with several state-of-the-art baselines. FINAL (Zhang & Tong, 2016) performs alignment based on first-order neighborhood consistency as well as node and edge feature similarity. IONE-D (Liu et al., 2019a) considers second-order node similarity along with community structure for cross-network learning. DANA (Hong et al., 2020) and GALign (Trung et al., 2020) are embedding-based approaches employing GCN. CENALP (Du et al., 2019), and BRIGHT (Yan et al., 2021) learn from multiple networks using a cross-network embedding method employing random walks. CrossMNA (Chu et al., 2019) learns two types of node vectors for multiple network alignment using structural information only. HackGAN (Yang et al., 2022) captures the local and global network features using an adversarial learning approach.

Evaluation metrics: Similar to several high-quality recent works (Yang et al., 2022; Yan et al., 2021; Trung et al., 2020), we compare the approaches using $Acc@q$, which indicates

Table 2: Statistics of the datasets

Networks	Nodes	Edges	Attributes	Anchors
Allmovie & IMDB	6011 5713	124709 119073	14	5176
Douban-Online & Douban-Offline	3906 1118	8164 1511	538	1118
DBLP & distributed copy	2151 2151	6306 5699	8	2151

Table 3: Comparison of network alignment algorithms on real-world datasets

Methods	Allmovie-IMDB				Douban Online-Offline				DBLP			
	Acc@1	Acc@10	MAP	AUC	Acc@1	Acc@10	MAP	AUC	Acc@1	Acc@10	MAP	AUC
FINAL	0.765	0.950	0.846	0.989	0.438	0.771	0.554	0.987	0.806	0.913	0.824	0.979
IONE-D	0.418	0.598	0.459	0.879	0.210	0.446	0.281	0.820	0.362	0.537	0.478	0.811
CrossMNA	0.546	0.632	0.589	0.968	0.239	0.437	0.341	0.762	0.395	0.569	0.521	0.870
CENALP	0.486	0.832	0.569	0.958	0.256	0.458	0.303	0.718	0.815	0.934	0.866	0.991
DANA	0.811	0.866	0.794	0.984	0.457	0.692	0.562	0.991	0.781	0.897	0.749	0.978
GAlign	0.775	0.874	0.812	0.993	0.441	0.780	0.554	0.990	0.841	0.926	0.875	0.987
BRIGHT	0.820	0.875	0.786	0.987	0.442	0.753	0.524	0.989	0.858	0.914	0.880	0.989
HackGAN	0.728	0.853	0.788	0.973	0.167	0.432	0.253	0.811	0.765	0.842	0.781	0.952
SAlign	0.831	0.920	0.867	0.996	0.497	0.821	0.587	0.994	0.877	0.958	0.889	0.993

if a node’s true anchor match is present in a list of top- q potential anchors. It is given as

$$Acc@q = \frac{\sum_{u_s^* \in V_s} \mathbb{1}_{P^*[u_s^*, u_t^*] \in R(u_s^*)}}{\#\{ \text{ground truth anchor links} \}} \quad (16)$$

where $(u_s^*, u_t^*) \in A$ and $R(u_s)$ is a list of highest q values in the row $P^*(u_s)$. Apart from this we also observe the *Mean Average Precision (MAP)* and *AUC* scores (Trung et al., 2020; Yang et al., 2022).

Experimental setup: We tune the hyper-parameters using the grid search algorithm implemented with Hyperopt. We consider order-4 graphlets and obtain the nodes’ structural feature vectors of size $o = 15$. We take the number of attention layers $L = 2$, with $K = 4$ attention heads in the first layer and $K = 1$ for the second. The embedding dimension for all three datasets is 100. The negative sampling size for each anchor in the training set is $|U| = 150$. The balancing factor $\beta = 2$. We train for 300 epochs using Adam as a gradient optimizer. The learning rate for the IMDB and DBLP datasets is 0.01 and for Douban is 0.005. We use the default values of hyper-parameters for the baselines except for the embedding dimensions for a fair comparison. We report all results averaged over 30 trials to minimize randomness. We use Pytorch libraries for implementation on a system with an 11GB NVIDIA GeForce GTX 1080 Ti GPU.

6. Results and Model Analysis

In this section, we provide the experimental results and their detailed analysis. Section 6.1 compares the performance of SAlign with the baselines on the evaluation metrics. Section 6.2 performs an ablation study on the model design to verify the importance of each of its components. Additionally, Section 6.3 investigates the adaptivity of SAlign to various adversarial conditions like structural noise, attribute noise, and graph size imbalance. We discuss the parameter sensitivity in Section 6.4, followed by scalability analysis in Section 6.5 and model convergence in Section 6.6. Finally, we perform case studies in Section 6.7 to understand and validate our claims.

6.1 Comparison with Baselines

The effectiveness of SALign in comparison to the baselines is shown in Table 3. We report our findings using 30% training data and observe the $Acc@1$ values for the supervised methods in Figure 4 by varying the training ratio from 10 – 80%. We clearly observe that SALign performs consistently better than the baselines over all the four considered metrics, $Acc@1$, $Acc@10$, MAP , and AUC . SALign improves the alignment accuracy, $Acc@1$, by 1.4 – 98% for the IMDB dataset, 8.8 – 197% for the Douban dataset, and 2.2 – 142% for the DBLP dataset. IONE-D performs poorly on the Douban and IMDB datasets. It is because of their diverse structures and varying network sizes. While CrossMNA performs 12.3% better than CENALP on the IMDB dataset, CENALP outperforms it by 7.1% on the Douban and 106.3% on DBLP networks. CrossMNA entirely depends on structural similarity and fails to capture the rich attribute information of the Douban networks. Also, CENALP is proven to perform better on networks with similar distributions like DBLP. SALign shows an 8.6% improvement over FINAL. It justifies that although FINAL captures nodes’ close relationships, it is necessary to capture higher-order structures for better results. GALign and DANA consistently outperform the other studied baselines. GALign uses different GCN layers to capture a node’s deeper neighborhood. DANA primarily focuses on capturing the domain invariant features for the alignment of diverse networks. BRIGHT outperforms GALign and DANA by additionally addressing the space disparity issue. SALign shows further improvement of 1.3%, 11%, and 2.2% on the IMDB, Douban, and DBLP datasets, respectively.

One important finding is that the $Acc@1$ disparity between SALign and the baselines, especially HackGAN, is significantly wider for Douban networks, having drastically different topologies. SALign performs around 6% better on the Douban dataset in comparison to the other two datasets. It shows that SALign, using a hierarchical attention mechanism, can capture the similarities even in very distinct networks where the alignment consistency constraints do not necessarily hold. Also, all the networks perform better on the DBLP networks, most likely owing to their linear node degree distributions and well-formatted user attributes. From Figure 4, we observe that all the methods’ performance improves with the increase in the training ratio. However, even with small training ratios, SALign maintains a relatively high performance as compared to the baselines. BRIGHT outperforms SALign on the IMDB dataset for higher training ratios by a small margin of 1.7 – 2%.

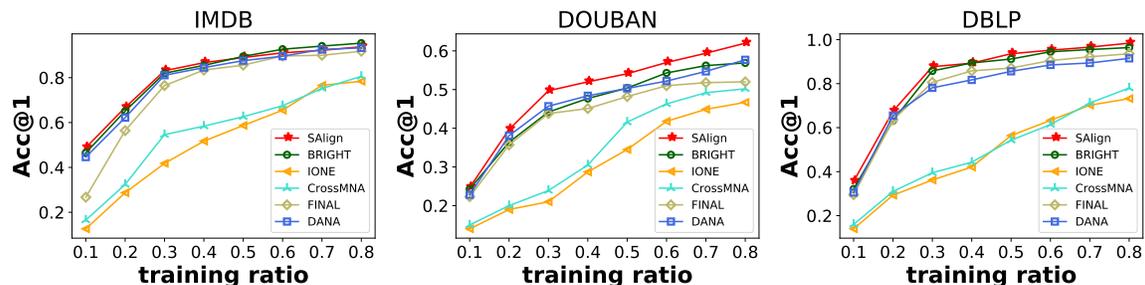


Figure 4: $Acc@1$ with different sizes of training ratios

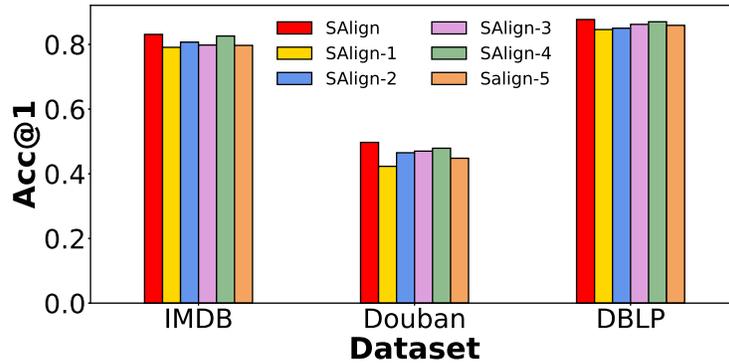


Figure 5: Ablation study

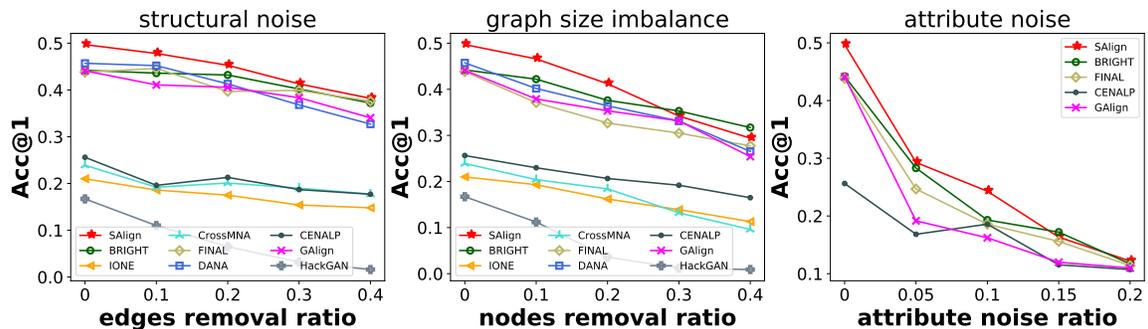


Figure 6: Model adaptivity analysis

6.2 Ablation Study on Model Design

We compare SAlign with its several variants to evaluate the effectiveness of each model component. The model variants based on using higher-order structural features and allowing inter-network information transfer are as follows: SAlign-1 employs only intra-network embeddings with no structural features. SAlign-2 employs only intra-network embeddings and concatenates the structural features with the node attributes, which are input to the first layer. SAlign-3 employs intra-network and inter-network embeddings without using structural features. SAlign-4, in addition, concatenates structural features. SAlign-5 follows the same pipeline as SAlign but does not employ the proposed attention mechanism, i.e., the model is trained without using the $w(u_i, u_k)$ and $w(u_i, v_j)$ terms in Equations 3 and 5 respectively. The results are shown in Figure 5. SAlign performs the best among its variants, showing $Acc@1$ improvement of 0.6 – 5%, 3.7 – 17.4%, and 0.8 – 3.6% on the IMDB, Douban, and DBLP dataset respectively. Hence it validates the importance of selectively incorporating structural features along with cross-network learning.

6.3 Model Adaptivity Analysis

Network alignment approaches rely heavily on the network structures and node attributes for their performance. Hence we study the robustness of SAlign and the baselines in the presence of different forms of noise, like structural noise, attribute noise, and graph size

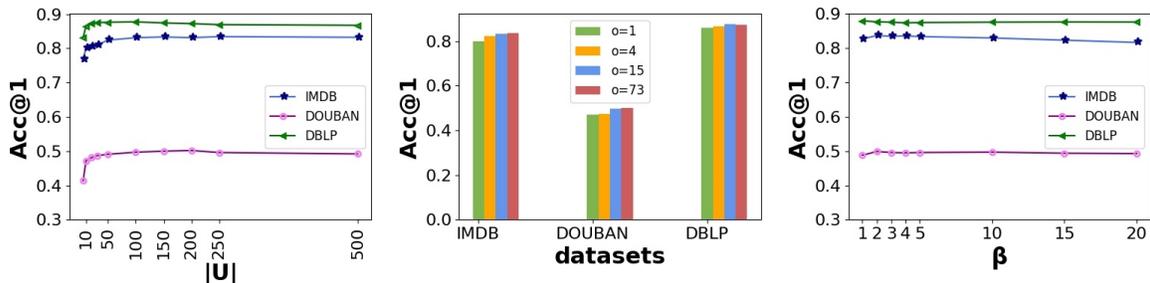


Figure 7: Parameter sensitivity analysis

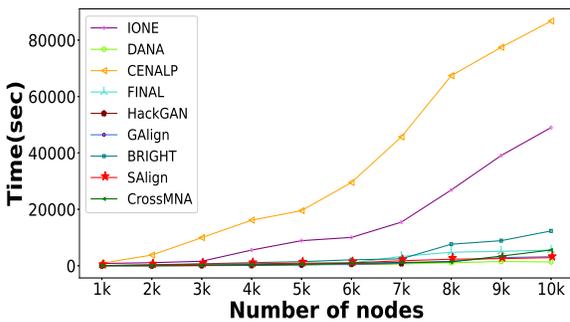


Figure 8: Computation analysis

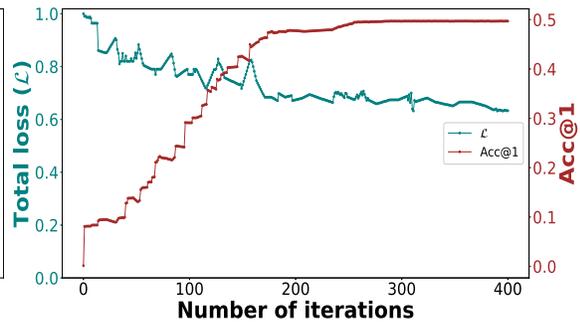


Figure 9: SAlign convergence on Douban dataset

imbalance. We randomly remove 10 – 40% of edges to induce structural noise, remove 10 – 40% of the total nodes of one network while keeping the other unchanged to create a size imbalanced network, and alter 5–20% of the total attribute values to introduce attribute noise (Trung et al., 2020; Nguyen et al., 2021). We show results for the Douban dataset in Figure 6. We observe that although the performance of all the methods degrades with the increase in noise ratio, SAlign still performs better than the baselines. The performance of BRIGHT is comparable to that of SAlign in the presence of high noise ratios. The $Acc@1$ score of SAlign drops by around 3.8 – 23.1% in the presence of structural noise, 6.2 – 40.8% in graph size imbalance, and 41.6 – 75.4% in the presence of attribute noise showing that SAlign is more sensitive to attribute variations than structural variations.

6.4 Parameter Sensitivity Analysis

We study the sensitivity of SAlign to some critical parameters. We analyze the model performance on the size of o , i.e., the orbit counts considered for capturing the higher-order structural features. We observe that the model performance stabilizes after 15 orbits, showing only slight improvement with 73 orbits. We next analyze the impact of the negative sampling size, $|U|$, used in Equation 12 on the alignment performance. We can see that even a small number of negative samples (i.e., $k = [10, 30]$) yields a good overall alignment performance. We also study the impact of the weight of the information transfer objective β . $\beta \in [1, 3]$ gives stable performance across all the datasets. Alignment performance on the IMDB dataset is more affected as β increases.

6.5 Execution Time and Scalability Analysis

We study the execution time and scalability of SAlign and the baselines by computing the time required to process the input networks as their size increases. We follow (Nguyen et al., 2021) to use a network-generative model to increase the network size gradually from 1000 to 10000 nodes. According to our observations, as shown in Figure 8, the execution time of all methods increases with network size, with CENALP and IONE exhibiting exponential increases, BRIGHT and FINAL showing significant increases, and the others showing gradual increases. CENALP has the highest execution time, primarily the cost of sampling random walks. HackGAN is not scalable to dense networks with more than 7000 nodes. SAlign, similar to the best-performing baseline, DANA, is computationally comparable to all other methods.

6.6 Model Convergence

We discuss the convergence of SAlign by observing its performance in terms of $Acc@1$ and total loss \mathcal{L} while increasing the number of iterations from 0 – 400. We record our observations for the Douban dataset as shown in Figure 9. We see that $mathcal{L}$ gradually decreases with the number of epochs, and similarly, $Acc@1$ increases and reaches a maximum in the first 280 iterations, then stabilizes as the number of epochs increases. This result clearly indicates the model convergence towards a similar alignment matrix after a fixed number of iterations. We find similar results in other datasets as well.

6.7 Case Studies

6.7.1 TACKLING STRUCTURAL HETEROGENEITY WITH HIERARCHICAL ATTENTION MECHANISM

We demonstrate the effect of taking higher-order structures in the attention mechanism of SAlign with a toy example. We take two arbitrary graphs and sample two node pairs as anchors, as shown in Figure 10. One anchor pair (K-8) has a similar neighborhood structure while the other (A-1) does not, emphasizing the presence of structural heterogeneity across the networks. In particular, we intuitively understand the embedding generation process of SAlign:

- Intra-network embedding depends on the node neighbors and the graphlet information of these neighbors. We see that A neighbors are densely connected, forming 3,4, and 5-length cycles. It means that each neighbor has a higher probability of occurring more times in each orbit, which the SAlign’s structurally aware attention mechanism captures. C, B, and D have decreasing importance in the intra-embedding of A. We observe a similar trend for the other nodes.
- Inter-network embedding depends on the corresponding anchor node and mainly contributes to capturing the mutual structural differences across the two networks. Information transfer across an anchor link depends on the structural similarity of the constituent nodes. As a result, attention is higher in A-1 and lower in K-8. Here, the networks’ higher order differences captured through graphlets prevent nodes’ embedding

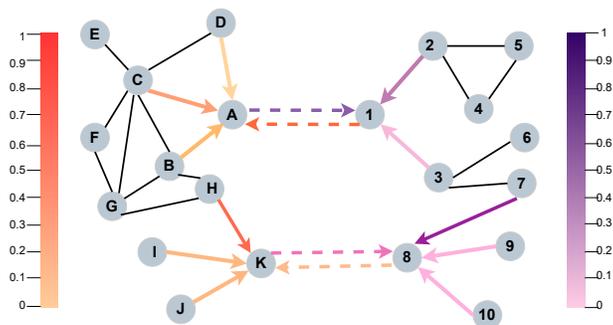


Figure 10: Toy example to understand the attention mechanism. Higher intensity of colors shows higher attention.

from losing their distinctive characteristics, thereby preventing the over-smoothing of representations, as discussed earlier.

- The combined embedding captures nodes’ immediate neighbors’ information, higher-order information through graphlets, and global information through the anchors, all mutually adding to their expressivity.

6.7.2 TACKLING THE OVER-SMOOTHING PROBLEM

We experimentally demonstrate how SALign’s hierarchical higher-order based attention mechanism tackles the over-smoothing problem. We compute the pairwise similarity scores of the node representations derived both using SALign, SALign-4 in which the graphlet information is considered as node features (both models use two encoder layers and follow the same pipeline) and DANA, which is the best-performing baseline. We subsequently plot the distribution of these pairwise similarity scores for these models in Figure 11. From the result, we find that for SALign-4 and DANA, the mass of the distribution is concentrated in a narrow region on the higher end of the x-axis indicating high similarity values of the representations of most node pairs (over-smoothing problem). On the other hand, for SALign, the distribution is more uniform, indicating that a large fraction of node representations are quite distinct from each other. This shows that our method successfully tackles the over-smoothing problem.

7. Conclusion and Future Work

This paper proposes SALign, an end-to-end framework for aligning structurally heterogeneous networks. SALign learns characteristic information from both networks and balances the amount of information transferred among them using a higher-order structure-aware attention mechanism, thereby increasing the expressivity of the learned representations. SALign effectively handles the over-smoothing issue of the node representations that prevail when capturing higher-order structures in the node representations. Although we evaluate SALign on limited datasets, we believe its applicability can be extended to other real-world networks, including transactions and biological networks. Rigorous empirical evaluations

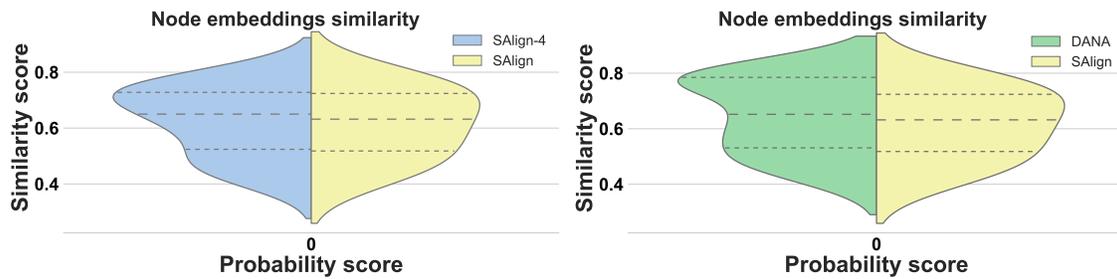


Figure 11: Probability distribution of pairwise node similarities using SAlign, its variant SAlign-4 (left), and the best-performing baseline DANA (right) for the Douban dataset. The vertical axis represents the similarity score, and the horizontal axes the probability.

reveal that SAlign consistently outperforms state-of-the-art baselines by at least 1.3 – 8% in terms of $Acc@1$ score.

As a next step to be worked on, we intend to extend the SAlign framework for aligning multiple networks. Learning from several networks may enhance the performance by generating more discriminate node representations. In addition, we would like to embed intra-link prediction in the framework and study if it enhances anchor link prediction.

References

- Almulhim, A., Dave, V. S., & Hasan, M. A. (2019). Network alignment using graphlet signature and high order proximity. In *International Conference on Machine Learning, Optimization, and Data Science*, pp. 130–142. Springer.
- Bayati, M., Gleich, D. F., Saberi, A., & Wang, Y. (2013). Message-passing algorithms for sparse network alignment. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(1), 1–31.
- Chen, C., Xie, W., Xu, T., Rong, Y., Huang, W., Ding, X., Huang, Y., & Huang, J. (2019). Unsupervised adversarial graph alignment with graph embedding. In *arXiv preprint, arXiv:1907.00544*.
- Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., & Sun, X. (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 3438–3445.
- Chen, X., Cai, R., Fang, Y., Wu, M., Li, Z., & Hao, Z. (2023). Motif graph neural network. *IEEE Transactions on Neural Networks and Learning Systems, Early Access Article*, 1–15.
- Cheng, A., Zhou, C., Yang, H., Wu, J., Li, L., Tan, J., & Guo, L. (2019). Deep active learning for anchor user prediction. In *28th International Joint Conference on Artificial Intelligence, IJCAI 2019*, pp. 2151–2157. International Joint Conferences on Artificial Intelligence.
- Chu, X., Fan, X., Yao, D., Zhu, Z., Huang, J., & Bi, J. (2019). Cross-network embedding for multi-network alignment. In *The world wide web conference*, pp. 273–284.

- Du, X., Yan, J., & Zha, H. (2019). Joint link prediction and network alignment via cross-graph embedding. In *IJCAI*, pp. 2251–2257.
- Feng, J., & Chen, S. (2020). Link prediction based on orbit counting and graph auto-encoder. *IEEE Access*, 8, 226773–226783.
- Hočevár, T., & Demšar, J. (2014). A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4), 559–565.
- Hong, H., Li, X., Pan, Y., & Tsang, I. W. (2020). Domain-adversarial network alignment. *IEEE Transactions on Knowledge and Data Engineering*, 34(7), 3211–3224.
- Huang, L., Wang, C., & Yu, P. (2023). Higher order connection enhanced community detection in adversarial multiview networks. *IEEE Transactions on Cybernetics*, 53(5), 3060–3074.
- Huang, L., Wang, C.-D., & Chao, H.-Y. (2018). A harmonic motif modularity approach for multi-layer network community detection. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1043–1048. IEEE.
- Jiang, M. (2021). Cross-network learning with partially aligned graph convolutional networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 746–755.
- Jin, Y., Song, G., & Shi, C. (2020). Gralsp: Graph neural networks with local structural patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 4361–4368.
- Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. In *arXiv preprint, arXiv:1609.02907*.
- Lan, L., Peng, H., Tong, C., Bai, X., & Dai, Q. (2021). Cross-network community sensing for anchor link prediction. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE.
- Li, Q., Han, Z., & Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*.
- Liu, L., Li, X., Cheung, W. K., & Liao, L. (2019a). Structural representation learning for user alignment across social networks. *IEEE Transactions on Knowledge and Data Engineering*, 32(9), 1824–1837.
- Liu, L., Zhang, Y., Fu, S., Zhong, F., Hu, J., & Zhang, P. (2019b). Abne: an attention-based network embedding for user alignment across social networks. *IEEE Access*, 7, 23595–23605.
- Long, Q., Jin, Y., Song, G., Li, Y., & Lin, W. (2020). Graph structural-topic neural network. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1065–1073.
- Man, T., Shen, H., Liu, S., Jin, X., & Cheng, X. (2016). Predict anchor links across social networks via an embedding approach. In *Ijcai*, Vol. 16, pp. 1823–1829.
- Nguyen, T. T., Pham, M. T., Nguyen, T. T., Huynh, T. T., Nguyen, Q. V. H., Quan, T. T., et al. (2021). Structural representation learning for network alignment with self-supervised anchor links. *Expert Systems with Applications*, 165, 113857.

- Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710.
- Qiu, Z., Shaydulin, R., Liu, X., Alexeev, Y., Henry, C. S., & Safro, I. (2021). Elruna: elimination rule-based network alignment. *Journal of Experimental Algorithmics (JEA)*, 26, 1–32.
- Ribeiro, P., Paredes, P., Silva, M. E., Aparicio, D., & Silva, F. (2021). A survey on subgraph counting: concepts, algorithms, and applications to network motifs and graphlets. *ACM Computing Surveys (CSUR)*, 54(2), 1–36.
- Sankar, A., Wang, J., Krishnan, A., & Sundaram, H. (2020). Beyond localized graph neural networks: An attributed motif regularization framework. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 472–481. IEEE.
- Sankar, A., Zhang, X., & Chen-Chuan Chang, K. (2017). Motif-based convolutional neural network on graphs. In *arXiv preprint, arXiv:1711.05697*.
- Saxena, S., Chakraborty, R., & Chandra, J. (2022). Hcna: Hyperbolic contrastive learning framework for self-supervised network alignment. *Information Processing & Management*, 59(5), 103021.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1), 61–80.
- Subramonian, A. (2021). Motif-driven contrastive learning of graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, pp. 15980–15981.
- Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pp. 1067–1077.
- Trung, H. T., Van Vinh, T., Tam, N. T., Yin, H., Weidlich, M., & Hung, N. Q. V. (2020). Adaptive network alignment with unsupervised and multi-order convolutional networks. In *2020 ICDE*, pp. 85–96. IEEE.
- Tu, K., Li, J., Towsley, D., Braines, D., & Turner, L. (2018). Learning features of network structures using graphlets. In *arXiv preprint, arXiv:1812.05473*.
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. *stat*, 1050, 4.
- Xia, Y., Gao, J., & Cui, B. (2021). imap: Incremental node mapping between large graphs using gnn. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2191–2200.
- Xu, H., Luo, D., Zha, H., & Duke, L. C. (2019). Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pp. 6932–6941. PMLR.
- Yan, Y., Zhang, S., & Tong, H. (2021). Bright: A bridging algorithm for network alignment. In *Proceedings of the Web Conference 2021*, pp. 3907–3917.

- Yang, L., Wang, X., Zhang, J., Yang, J., Xu, Y., Hou, J., Xin, K., & Wang, F.-Y. (2022). Hackgan: Harmonious cross-network mapping using cyclegan with wasserstein-procrustes learning for unsupervised network alignment. *IEEE Transactions on Computational Social Systems*, 10(2), 746–759.
- Zhang, S., & Tong, H. (2016). Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1345–1354.
- Zhang, S., Tong, H., Maciejewski, R., & Eliassi-Rad, T. (2019). Multilevel network alignment. In *The World Wide Web Conference*, pp. 2344–2354.
- Zhao, H., Xu, X., Song, Y., Lee, D. L., Chen, Z., & Gao, H. (2018). Ranking users in social networks with higher-order structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
- Zhou, F., Liu, L., Zhang, K., Trajcevski, G., Wu, J., & Zhong, T. (2018). Deeplink: A deep learning approach for user identity linkage. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1313–1321. IEEE.