Denoise Network Structure for User Alignment Across Networks via Graph Structure Learning

Li Liu, Chongyang Wang, Youmin Zhang(✉), Ye Wang, Qun Liu, and Guoyin Wang

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
ymzhang0103@hotmail.com

Abstract. User alignment aims to identify accounts of one natural person across networks. Nevertheless, different social purposes in multiple networks and randomness of following friends form the diverse local structures of the same person, leading to a high degree of non-isomorphism across networks. The edges resulting in non-isomorphism are harmful to learn consistent representations of one natural person across networks, i.e., the structural "noisy data" for user alignment. Furthermore, these edges increase the time complexity, compromising the model's efficiency. To this end, we propose a network structure denoising framework to learn an alignment driven structure heuristically. Specifically, under the guidance of alignment driven loss, parameter sharing encoder and graph neural network for structure denoising are learned using an iterative learning schema. Experiments on real-world datasets demonstrate the outperformance of the proposed framework in terms of efficiency and transferability.

Keywords: User alignment · Graph neural networks · Graph structure learning · Structure denoise

1 Introduction

Social network alignment aims to identify accounts of one natural person across multiple online social platforms. Aligning users across networks benefits the data transfer between standalone social networks and alleviates the "data isolation" issue in several data mining tasks, including information diffusion, recommendation, etc. Recently, graph representation learning (GRL) algorithms have demonstrated their superior performance on this task attributed to their ability to represent users without manual efforts.

Generally speaking, graph representation learning algorithms attempt to represent nodes by preserving the structural proximity. Specifically, according to the structure across networks, several studies [16,17,30–32,34] are proposed to learn the representations of users by aggregating and combing features of their

Y. Tan and Y. Shi (Eds.): DMBD 2022, CCIS 1744, pp. 105–119, 2022.
https://doi.org/10.1007/978-981-19-9297-1_9

neighbors. Based on the learned representations, nodes across networks can be aligned using the similarity of representations [28,35]. The aforementioned processes indicate that the network structure plays a critical role in user representation, and thus influences the precision of downstream alignment.

Nevertheless, people usually join multiple platforms for different social purposes. Therefore, the random nature of users' behavior in following friends is unavoidable across social networks. These factors result in the diverse local structure topologies across social networks for the same person, i.e., a high degree of non-isomorphism between multiple networks. Different from representing users within a single network, the non-isomorphism between the network structure brings structural "noisy data" for representing users across networks, compromising the effectiveness of the user alignment task. For the GRL algorithms, these "noisy data" will trigger a cascade of negative influences in their aggregation and combination process. Moreover, due to the presence of the "noisy data", high time and space complexities are unavoidable as many GRL algorithms are exponential growth with the increase in the scale of the network.

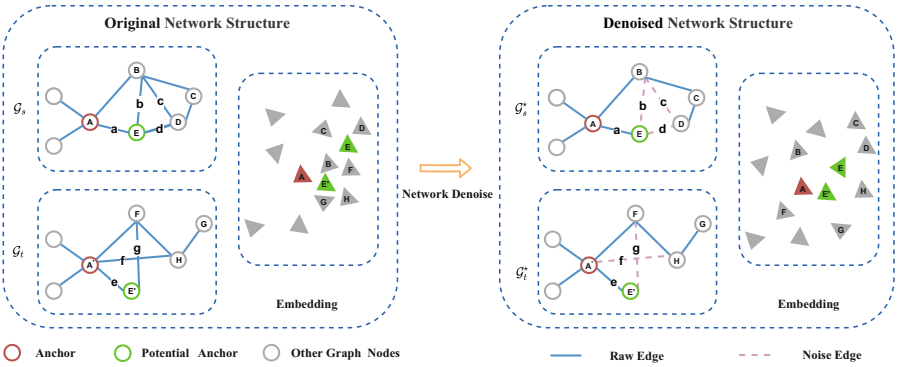


Fig. 1. A toy example for explaining why structural "noisy data" compromises the representation learning across networks.

Figure 1 illustrates an example to illustrate the key ideas of why the structural "noisy data" compromises the GRL across social networks. In Fig. 1, $\mathcal{G}_s$ and $\mathcal{G}_t$ are two networks. A and A' are labeled anchor (accounts belonging to the same person). E and E' are the potential anchors (nodes to be aligned). Due to the diverse social purposes in multiple networks, E' and E' may follow different users in separated networks such as B, D in $\mathcal{G}_s$ and F in $\mathcal{G}_t$. This phenomenon results in a really dense but non-isomorphism local structure around E and E', such as edges $\{a, b, c, d\}$ in $\mathcal{G}_s$ and edges $\{e, f, g\}$ in $\mathcal{G}_s$. According to the structure preserving objective of GRL [14,23,24,26], nodes with dense connectivity will be close in the embedding space. Therefore, B, E, and D will be close due to the existence of edges $\{b, c, d\}$ and F, E' and H will be close due to the existence of edges $\{f, g\}$. Nevertheless, the "overly-close" embedding space compromises the alignment task as potential anchors are hard to be aligned preciously, especially

with many non-anchor nodes (such as F and B) around them. Therefore, we name the edges $\{b, c, d, g, f\}$ as alignment task oriented "structural noisy data". To address this issue, we attempt to design a strategy that can denoise the structure heuristically. As shown in the right part of Fig. 1, we hope the structures $\mathcal{G}_t^\star$ and $\mathcal{G}_s^\star$ can be learned. Compared to the original network structure, E and E' can be close due to their connectivity to anchor A. Meanwhile, B and D can be far away from them due to the absence of edge $\{b, c, d, f, g\}$, facilitating the downstream user aligning module.

To this end, we first leverage a parameter sharing graph encoder to obtain the primary embedding of every node. To further denoise the original network structure, a graph neural network is adopted for determining which edges can be removed. Under the guidance of the designed alignment-oriented loss and structure regularization, we perform the aforementioned process iteratively for parameter learning. Finally, the denoised networks can be obtained via the learned graph neural networks. The denoised networks also can be transferred to other state-of-the-art (SOTA) alignment models for efficient learning.

Our main contributions are summarized as follows:

- We propose a network structure denoising framework for the user alignment task. With the guidance of the alignment-oriented loss, a parameterized graph neural network is learned to denoise the network structure.
- We investigate the transferability of the learned network structure. We provide evidence that, beyond the graph encoder adopted in the framework, the denoised structure can boost several SOTA network alignment algorithms.
- We evaluate the proposed framework by applying it to several state-of-the-art models. The experimental results on three real-world datasets demonstrate the effectiveness of the proposed framework.

2 Related Work

2.1 Network Alignment

Network alignment aims to identify different accounts of one natural person. Recently, graph representation learning algorithms demonstrate their superior performance. Compared to classification based [15] and matrix factorization based algorithm [22], it learns user representations via preserving structural proximity without manual efforts, and the Stochastic Gradient Descent and sampling strategy adopted in the learning process guarantees its effectiveness. Generally speaking, the related studies can be categorized into *supervised* and *unsupervised* according to the existence of labeled anchors.

For supervised algorithms, IONE [17] learns representations of users via preserving second order structure proximity and leverages cosine similarity to identify the potential anchors. PALE [20] learns node embedding in the separated network and further leverage a shallow neural network to conduct the user alignment. DeepLink [35] and DCIM [21] further adopt deep neural networks for constructing the mapping function. Besides, SNNA [16] leverages a generate adversarial network to train a mapping function. Different from the above studies that focus on

constructing mapping functions, some works investigate the structure consistency across networks. NextAlign [31] studies the correlation between graph convolutional networks and the assumption of network consistency adopted in the traditional alignment model. cM^{2}NE [28] proposes an end-to-end contrastive learning framework to model the inconsistency across social networks. Also, several studies attempt to learn better representation to facilitate the mapping functions. MGCN [2] uses convolution on both local and hypergraph network structure to learn network embedding. iMap [27] iteratively constructs sub-graphs and adopts graph neural network to learn the representations.

Under the condition of the absence of the labeled anchors, the unsupervised models are designed based on the consistency assumption across networks [3]. In general, structural and attributes representations are learned simultaneously to complement each other [10,34]. To align users from the distribution perspective, UAGA [1] and WAlign [7] try to perform the alignment according to the distributions of the entire embedding space. After learning the embeddings across networks, they adopt Wasserstein distance to measure the discrepancy of nodes' distributions to identify potential anchors.

The aforementioned methods demonstrate their outperformance in the network alignment task. However, these methods are learned using networks that may contain structural "noisy data" for the alignment task. As we introduced in Fig. 1, structural "noisy data" will compromise the learning efficiency and performance, which motivates our proposed denoising framework.

2.2 Graph Structure Learning

As the network grows in size, the graph representation learning algorithms faced several challenges, including noisy data, training efficiency, etc. To this end, graph structure learning algorithms are proposed to learn a proper network structure for representation learning. Primary studies attempt to learn network structure based on the similarities between nodes. Gidaris et al. [8] construct K nearest neighbor graph based on structure similarity via setting a threshold. Wang et al. [25] adopt graph neural network learn node representations. They further calculate the similarity between nodes to construct the network structure. Rather than the single similarity mentioned above, Jonathan et al. [9] leverages the multiple similarities to learn the network structure, where weak similarities between nodes are also incorporated into the network construction. Chen et al. [4] propose an iterative learning schema for learning graph networks. They learn the network structure using a parameterized adjacency matrix.

Rather than learning network structure only, some studies are proposed to learn network structure and model of the target task simultaneously. AneesKazi et al. [13] learn node representations using a graph neural network. They further construct a graph generator to learn a proper network structure under the guidance of the predictions of a downstream GNN. Zheng et al. [33] propose a graph structure learning algorithm to sparse the network structure, where a deep neural network is adopted to model the network sparsing process. Luca et al. [6] propose a probabilistic graph generator where edges are learnable parameters. The generator is optimized with the task driven graph neural networks simultaneously.

Jiang et al. [11] proposed a framework consisting of two components, the first one to learn graph structure, and the second one is a graph convolutional network. Jin et al. [12] learn network structure under the principle that the learned network has certain characteristics, including sparsity and low rank.

Different from the aforementioned studies which may add and remove edges for optimizing graph structure, we focus on denoising (only removing) the network structure for efficient graph representation learning across networks. Furthermore, rather than learning structure within the single network, we specifically design an alignment-oriented loss for network denoising.

3 Preliminaries

In this section, we provide brief descriptions of the notions and definitions.

Network: We use $\mathcal{G} = (V, A)$ to denote the network, where V is the set of the nodes. Every node $v_i \in V$ represents one user. A is the adjacency matrix of $\mathcal{G}$. $a_{i,j} = 1$ when there is a relationship between v_i and v_j, $a_{i,j} = 0$ otherwise.

Anchors: The anchors denote the identities across networks of one natural person. Given two networks $\mathcal{G}_s$ and $\mathcal{G}_t$. $v_s \in V_s, v_t \in V_t$ are one anchor if they belong to one person.

Network Alignment: Given two networks $\mathcal{G}_s$ and $\mathcal{G}_t$, the network alignment aims to learn a function $f(\boldsymbol{u}_s, \boldsymbol{u}_t) \in \{0, 1\}$ to determine whether v_s and v_t are the anchor pair, where $\boldsymbol{u}_s$ and $\boldsymbol{u}_t$ are the embeddings/features that can be learned by graph representation learning algorithms.

Network Denoise: Given one network $\mathcal{G} = (V, A)$, network denoise aims to learn a mask generator $Mask = g(V, A)$ that determines which edges can be removed for the user alignment. Based on the $Mask$, we can obtain the denoised network $\mathcal{G}^\star = (V, A^\star)$. Rather than removing nodes, we only remove edges as removing nodes may exclude potential anchors in the network, which is inadvisable for the network alignment task.

4 Model Framework

To design a structure denoising framework for the network alignment task, we propose to learn a parameterized mask generator based on the network representations across networks. As shown in Fig. 2, our proposed framework consists of three components. The first is a graph encoder across networks that learns the initial representations according to the network structure. The second is the parameterized graph neural network that acts as the mask generator. The third is the network regularizer and the alignment loss calculated by the graph encoder.

4.1 Graph Encoder Across Networks

The graph encoder in this paper serves two purposes. The first is to learn initial embeddings for nodes across networks. The second is to learn embeddings

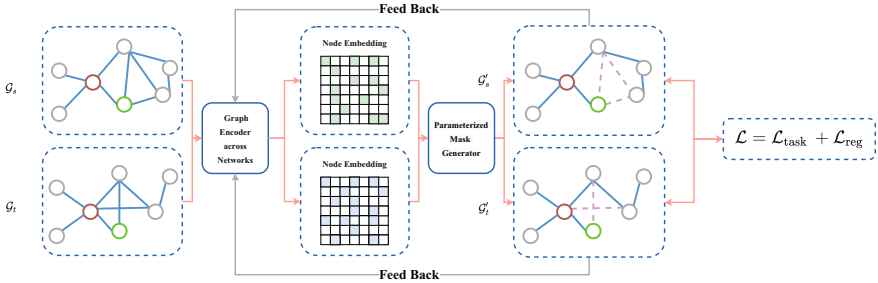


Fig. 2. Model framework

of denoised networks for calculating the alignment loss. Rather than designing a graph representation learning algorithm, our goal is to design a denoising framework for networks. To this end, we adopt IONE [17] as our graph encoder attributed to its learning efficiency in the iterative learning process. First, we denote the learned initial embedding as $X_{s/t}^{Init} \in \mathbb{R}^{n_{s/t} \times d}$

$$X_s^{Init}, X_t^{Init} = Encoder(\mathcal{G}_s, \mathcal{G}_t) \tag{1}$$

where $n_{s/t}$ are the number of nodes in $\mathcal{G}_{s/t}$, d is the dimension of learned embeddings. We believe that the learned embeddings contain structural features of nodes across networks, and properly utilizing them can benefit the learning of the mask generator.

4.2 Parameterized Mask Generator

Given the initial embeddings learned by the graph encoder, we further leverage a graph neural network to determine which edges to remove for the alignment task. Here we use GAT [23] for this task attributed to its powerful ability in modeling the structure and features simultaneously, shown as Eq. (2).

$$Z = GAT(X, A; \Theta) \tag{2}$$

where A is the adjacency matrix for one network and X is the corresponding learned initial embedding. Z is the latent representation learned by GAT and Θ is the learnable parameter.

After obtaining the latent representation Z, we try to norm the representation Z_{norm} to avoid the influence of the value scale, shown in Eq. (3).

$$Z_{norm} = \frac{v_i}{\max(||v_i||_2, \epsilon)}, [v_0, v_1, v_2, ..., v_i] \in Z^T \tag{3}$$

where $||\ ||_2$ is the L_2 norm. And we set $\epsilon = 1e - 12$ to avoid the zero values in the denominator.

Algorithm 1. The Learning Algorithm of the Proposed Framework

Input: A training set of anchors V_a, the network alignment encoder *Encoder*, The original networks $\mathcal{G}_s = (V_s, A_s)$ and $\mathcal{G}_t = (V_t, A_t)$. The number of negative samples N_{neg}. The number of iterations $Iter$

Output: Learned parameters for mask generator Θ. Denoised network $\mathcal{G}_s^\star$, $\mathcal{G}_t^\star$

```
Sample N_neg negative nodes from all nodes as S_neg
Calculate node embedding using encoder X_s^Init, X_t^Init = Encoder(G_s, G_t)
for i = 0; i < Iter; i = i + 1 do
    if i == 0 then X_s = X_s^Init, X_t = X_t^Init
    end if
    Calculate Z_norm according to Eqs. (2) and (3).
    Calculate A_s' and A_t' according to Eqs. (4) and (5). Then get the structure G_s', G_t'
    Calculate node embedding matrix using encoder X_s, X_t = Encoder(G_s', G_t')
    Calculate L_reg and L_task according to Eqs. (6) and (7).
    Update Θ using the Adam optimizer
end for
G_s^⋆, G_t^⋆ = G_s', G_t'
```

Based on the normalized representation, we try to learn the mask of the adjacency matrix, shown in Eq. (4).

$$Mask = \sigma(Z_{norm} \times Z_{norm}^T) \odot A \tag{4}$$

where σ is the sigmoid activation function, and $\odot$ is the Hadamard product. Then the values in $\sigma(Z_{norm} \times Z_{norm}^T)$ denote the importance scores of all node pairs. We further use the adjacency matrix A to filter the $m_{i,j} \in Mask$ to ensure the corresponding users v_i and v_j have an edge. After that, we can select denoised network A' based on a hyper-parameter R, given as Eq. (5).

$$A^{'} = top(Mask; R) \tag{5}$$

where $top(Mask; R)$ means ranking all elements in the $Mask$ matrix and then retaining the top R largest elements. We performs the above processes on $\mathcal{G}_s = (V_s, A_s), \mathcal{G}_t = (V_t, A_t)$ separately to obtain $A_s^{'}$ and $A_t^{'}$. Then we feed them to the graph encoder across networks to calculate the loss. We repeat it for the parameter learning of the mask generator until a stable performance is achieved.

4.3 Design of the Loss Function

To guide the learning of the parameter Θ in the mask generator, we design an alignment-oriented loss function. Specifically, we try to achieve two objectives.

Objective 1: Given the denoised networks, we hope the embedding of anchors should be as close as possible. Meanwhile, the anchor node should be apart from other nodes as far as possible. We define the loss function $\mathcal{L}_{task}$ as Eq. (6).

$$\mathcal{L}_{task}(v_s, v_t) = |cos(X_s, X_t)| - |cos(X_{s/t}, X_{t/s}^{neg})| \tag{6}$$

where $X_{s/t}$ is the embedding learned by the graph encoder when feeding the denoised network to it in each iteration. The $X_{t/s}^{neg}$ is the embeddings of the nodes in the opposite network sampled according to the degree distribution.

Objective 2: In addition to objective 1, to ensure the robustness of the denoised network, we hope the denoised network retains some characteristics of the original network. We use line-wise cosine similarities of embeddings learned from the denoised and the original networks, shown in Eq. (7).

$$\mathcal{L}_{reg}(X^{\star}, X^{Init}) = cos(X^{\star}, X^{Init}) \tag{7}$$

Finally, we combine the above objective functions $\mathcal{L} = \mathcal{L}_{task} + L_{reg}$ to guide the optimization of the parameters in the proposed framework. Algorithm 1 provides a detailed description of the learning process.

5 Experiment and Analysis

5.1 Datasets and Evaluation Metrics

To evaluate the performance of the proposed framework, we conduct extensive experiments on three public datasets. The first Foursquare-Twitter [17,29,35], the second is ACM-DBLP [31,35] and the third is DBLP [18]. Foursquare and Twitter are two social networks. The labeled anchors are obtained by finding users who provide their Twitter accounts in Foursquare profiles. ACM and DBLP are two academic social networks, where identical authors in both ACM and DBLP are the anchors. DBLP are two academic social networks, where authors are split into different co-author networks by filtering publication venues of their papers.Table 1 lists out the statistics of the datasets.Table 1 lists out the statistics of the datasets.

Table 1. Statistics of datasets.

Networks	#Nodes	#Edges	#Anchors
Foursquare	5313	54233	1609
Twitter	5120	130575	
ACM	9872	39561	6325
DBLP	9916	44808	
DBLP_DM	11526	47326	1295
DBLP_ML	12311	43948	

We use a widely adopted metric $Precison@N$ [5,17,19,35–37] to evaluate the performances of the above three datasets, shown in Eq. (8).

$$Precision@N = \frac{|CorrUser@N|^X + |CorrUser@N|^Y}{|UnMappedAnchors| \times 2} \tag{8}$$

where $|CorrUser@N|$ is the number of anchors that can be identified among the top-N candidate list defined by the cosine similarity. $|UnMappedAnchors|$ is the number of all testing anchors. We report the averaged $Precison@N$ by considering one network as the source and the target network respectively. For the configurations of the baseline models, the default settings of the open source codes provided by the authors are utilized.

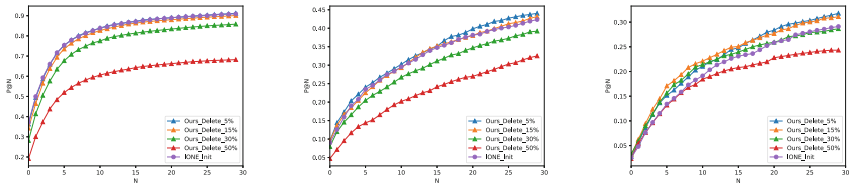
5.2 Baseline Methods

We evaluate our framework based on three state-of-the-art models.

- **IONE** [17] is an embedding sharing based algorithm. It learns a unified latent space for the alignment via preserving the second-order proximity.
- **DEEPLINK** [35] is an embedding mapping based algorithm. It learns representation learned using random walk and skip-gram algorithms, deep neural networks and dual learning are leveraged to align users.
- **NeXtAlign** [31] proposes a RelGCN-U model and a scoring function to learn user embeddings. This model achieves a good trade-off between alignment consistency and alignment disparity,

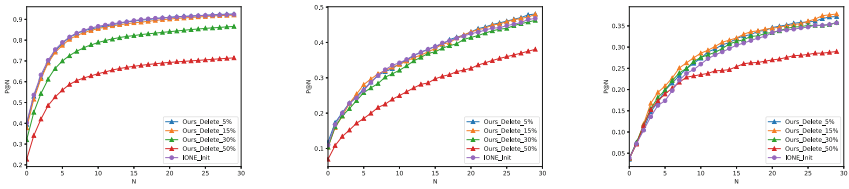
5.3 Performance Comparison

We first investigate the performance when deleting different ratios of edges. Figure 3 and Fig. 4 illustrate the $Precision@1-30$ performance under training ratio 50% and 60%, where the network encoder is IONE [17] attributed to its training efficiency. And the ratios (the parameter R in Eq. (5) of deleted edges are set as [5%, 15%, 30%, 50%]. We observed that even if we delete 15% edges of the original network, the performance of the proposed framework does not decrease on all datasets, indicating the effectiveness of the proposed framework. Under the training ratio of 60% of the ACM-DBLP dataset, the model on the denoised network still shows comparable performance when we deleted 30% edges.



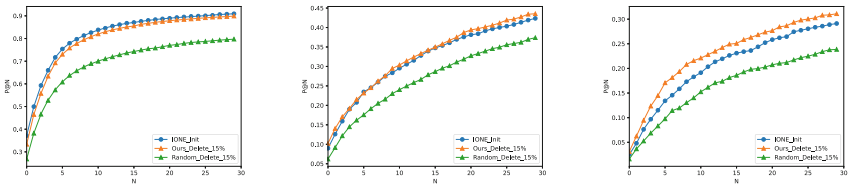
(a) ACM-DBLP (b) Foursquare-Twitter (c) DBLP

Fig. 3. Precision@1-30 performance under training ratio of 50%.



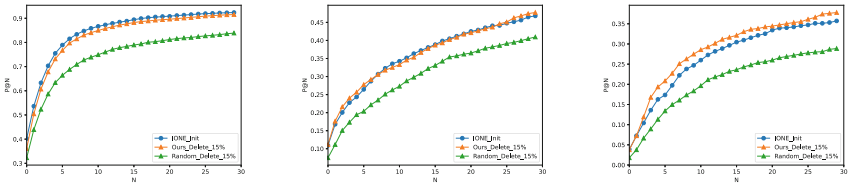
(a) ACM-DBLP (b) Foursquare-Twitter (c) DBLP

Fig. 4. P@1-30 performance under training ratio of 60%.



(a) ACM-DBLP (b) Foursquare-Twitter (c) DBLP

Fig. 5. Comparison with randomly deleting edges under training ratio of 50%.

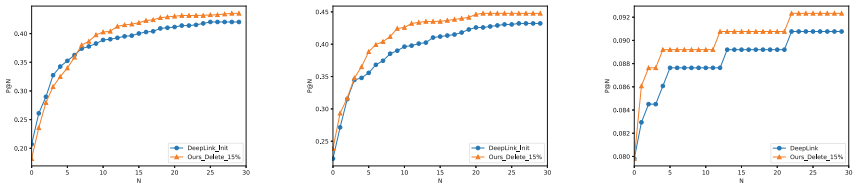


(a) ACM-DBLP (b) Foursquare-Twitter (c) DBLP

Fig. 6. Comparison with randomly deleting edges under training ratio of 60%.

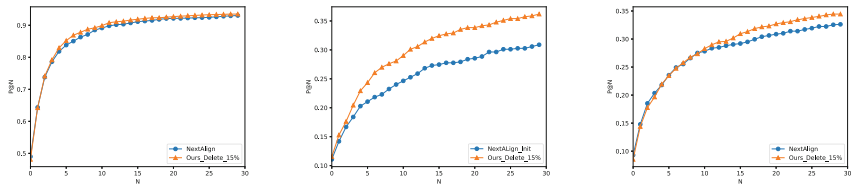
To prove the effectiveness of the mask generator for the alignment task, we compare our framework with a method that randomly deletes 15% edges, illustrated in Fig. 5 and Fig. 6. Compared with randomly deleting 15% edges, our framework shows a significant increment under the $Precision@1-30$ metric. This phenomenon means that heuristically selecting edges for the alignment task is crucial, randomly removing edges may break the characteristics of the original network, compromising the performance of alignment. The increment indicates that our proposed framework can heuristically remove the edges for the network alignment task.

Further, to demonstrate that the mask generator can perform masking operations using all the commonly used graph convolution networks, we replace the GAT [23] in the mask generator with GCN [14], which also achieves the similar results. Figure 9 shows that using any GNN as the mask generator can be obtained similar results.



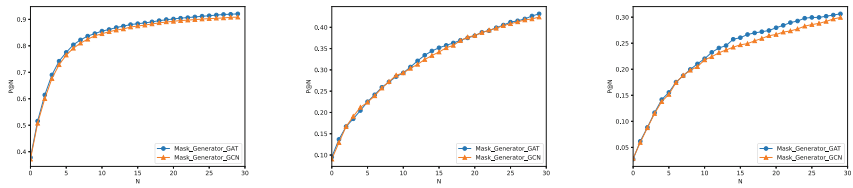
(a) ACM-DBLP (b) Foursquare-Twitter (c) DBLP

Fig. 7. Performance comparison when feeding the denosied network to DeepLink.



(a) ACM-DBLP (b) Foursquare-Twitter (c) DBLP

Fig. 8. Performance comparison when feeding the denosied network to NeXtAlign.



(a) ACM-DBLP (b) Foursquare-Twitter (c) DBLP

Fig. 9. Performance comparison when feeding the denosied network to NeXtAlign.

To further investigate the transferability of the denoised network, we feed the learned networks with 15% edges removed to two STOA models, including the DeepLink [35] and the NeXtAlign [31]. The performance are shown in Fig. 7 and Fig. 8. We observe that, compared with using the original network structure, there is an increment when we feed the denoised network learned by IONE to the DeepLink model. One possible reason for this is the powerful ability of the deep neural networks in the DeepLink model. For the NeXtAlign model, we observe a similar performance on the ACM-DBLP dataset compared with the original network, while there is a significant increment on the Foursquare-Twitter dataset. We notice that the ratio of anchors of ACM-DBLP is higher than it of Foursquare-Twitter. It indicates that the denoising structure will benefit the alignment model more when there are few anchors for supervision.

As illustrated in Table 2, we investigate the time consumption when feeding the denoised network with 15% edges removed to the IONE model. We implement our model using PyTorch and run it on GeForce RTX 3090 GPU and

Table 2. Time consumption on different datasets when deleted 15% edges

Dataset	The original network	The denosied network	Time saving (%)
ACM-DBLP	698 s	561 s	19.6%
Foursquare-Twitter	569 s	462 s	18.8%
DBLP	675 s	807 s	16.3%

Intel(R) Xeon(R) Silver 4210R CPU. We observe that the denoised network can same time more than 15% on both the networks. The percentages of the time saving are 19.6%,18.8% and 16.3% for ACM-DBLP, Foursquare-Twitter and DBLP, respectively. It provides evidence that denoising the network structure can benefit the alignment model for efficient learning.

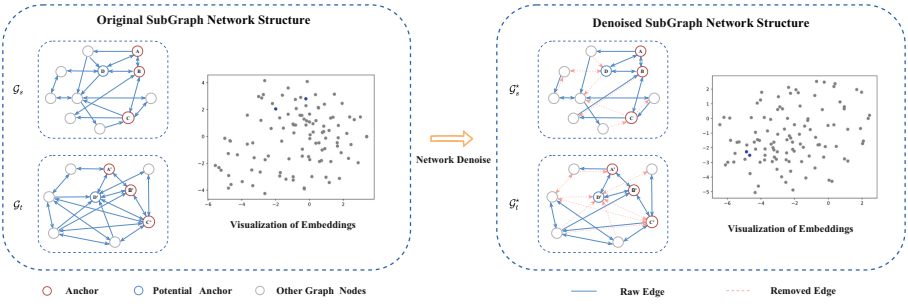


Fig. 10. Case study of the real-world dataset

5.4 Case Study

Figure 10 illustrates the structure of two subgraphs and the corresponding visualization of embeddings in the Foursqure-Twitter dataset. We notice that the potential anchor $((D, D^{'})$ has different local connectivities in the original network, such as D' connects to the anchor C' in $\mathcal{G}_t$ while D is not in $\mathcal{G}_s$. Meanwhile, D and D' connect several other nodes in separate networks, which are the structural "noisy data" for the alignment. Thus we observe that the potential anchors are far from each other in the embedding space, compromising the alignment task. After deleting 15% edges of the original network structure by our proposed framework, illustrated in the right part of Fig. 10, we obtain a more isomorphic local structure around the D and D', i.e., the structural "noisy data" are denoised. It results in a more properly distributed embedding space, where the D and D' are close in this space, making them easy to align.

6 Conclusion

In this paper, we study the problem of denoising network structure for the user alignment task. We proposed a framework based on graph structure learning. Specifically, under the guidance of the specially designed alignment loss and structure regularization, a graph encoder across networks and a parameterized mask generator are learned in an iterative learning schema. Then whether one certain edge can be removed is determined by the mask generator. We conduct experiments from several perspectives, including performance, transferability, and running time, and the visualization of learned space. Results demonstrate the effectiveness of the proposed model. We hope this framework can provide a way to deploy an alignment model in practice attributed to its ability to denoise the structural "noisy data" and reduce the training complexity. Further studies will include denoising network structure in dynamic environments and ingenious GNNs for the mask generator.

Acknowledgements. The work is partially supported by National Natural Science Foundation of China (61936001, 61806031), and in part by the Natural Science Foundation of Chongqing (cstc2019jcyj-cxttX0002, cstc2020jcyj-msxmX0943), and in part by the key cooperation project of Chongqing Municipal Education Commission (HZ2021008), and in part by Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN202100629, KJQN202001901, KJQN201901901), and in part by Doctoral Innovation Talent Program of Chongqing University of Posts and Telecommunications (BYJS202118). This work is partially done when Li Liu works at Hong Kong Baptist University supported by the Hong Kong Scholars program (XJ2020054).

References

1. Chen, C., et al.: Unsupervised adversarial graph alignment with graph embedding. arXiv preprint arXiv:1907.00544 (2019)
2. Chen, H., Yin, H., Sun, X., Chen, T., Gabrys, B., Musial, K.: Multi-level graph convolutional networks for cross-platform anchor link prediction. In: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, CA, USA, 23–27 August 2020, pp. 1503–1511 (2020)
3. Chen, X., Heimann, M., Vahedian, F., Koutra, D.: Cone-align: consistent network alignment with proximity-preserving node embedding. In: The 29th ACM International Conference on Information and Knowledge Management, Ireland, 19–23 October 2020, pp. 1985–1988 (2020)
4. Chen, Y., Wu, L., Zaki, M.J.: Deep iterative and adaptive learning for graph neural networks. arXiv preprint arXiv:1912.07832 (2019)
5. Chu, X., Fan, X., Zhu, Z., Bi, J.: Variational cross-network embedding for anonymized user identity linkage. In: The 30th ACM International Conference on Information and Knowledge Management, Queensland, Australia, 1–5 November 2021, pp. 2955–2959 (2021)
6. Franceschi, L., Niepert, M., Pontil, M., He, X.: Learning discrete structures for graph neural networks. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 1972–1982. PMLR (2019)

7. Gao, J., Huang, X., Li, J.: Unsupervised graph alignment with wasserstein distance discriminator. In: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Singapore, 14–18 August 2021, pp. 426–435 (2021)
8. Gidaris, S., Komodakis, N.: Generating classification weights with gnn denoising autoencoders for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21–30 (2019)
9. Halcrow, J., Mosoi, A., Ruth, S., Perozzi, B.: Grale: designing networks for graph learning. In: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, CA, USA, 23–27 August 2020, pp. 2523–2532. ACM (2020)
10. Heimann, M., Shen, H., Safavi, T., Koutra, D.: Regal: representation learning-based graph alignment. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 117–126. ACM (2018)
11. Jiang, B., Zhang, Z., Lin, D., Tang, J., Luo, B.: Semi-supervised learning with graph learning-convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019, pp. 11313–11320. Computer Vision Foundation/IEEE (2019)
12. Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., Tang, J.: Graph structure learning for robust graph neural networks. In: KDD 2020: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, 23–27 August 2020, pp. 66–74. ACM (2020)
13. Kazi, A., Cosmo, L., Ahmadi, S.A., Navab, N., Bronstein, M.: Differentiable graph module (DGM) for graph convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. (Early Access) (2022)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017. OpenReview.net (2017)
15. Kong, X., Zhang, J., Yu, P.S.: Inferring anchor links across multiple heterogeneous social networks. In: 22nd ACM International Conference on Information and Knowledge Management, CIKM 2013, San Francisco, CA, USA, 27 October–1 November 2013, pp. 179–188. ACM (2013)
16. Li, C., et al.: Adversarial learning for weakly-supervised social network alignment. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 996–1003 (2019)
17. Liu, L., Cheung, W.K., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 1774–1780. IJCAI/AAAI Press (2016)
18. Liu, L., Li, X., Cheung, W.K., Liao, L.: Structural representation learning for user alignment across social networks. IEEE Trans. Knowl. Data Eng. **32**(9), 1824–1837 (2020)
19. Liu, L., Zhang, Y., Fu, S., Zhong, F., Hu, J., Zhang, P.: ABNE: an attention-based network embedding for user alignment across social networks. IEEE Access **7**, 23595–23605 (2019)
20. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 1823–1829. IJCAI/AAAI Press (2016)
21. Nie, Y., Jia, Y., Li, S., Zhu, X., Li, A., Zhou, B.: Identifying users across social networks based on dynamic core interests. Neurocomputing **210**, 107–115 (2016)

22. Tang, R., Jiang, S., Chen, X., Wang, H., Wang, W., Wang, W.: Interlayer link prediction in multiplex social networks: an iterative degree penalty algorithm. Knowl. Based Syst. **194**, 105598 (2020)
23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
24. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, CA, USA, 13–17 August 2016, pp. 1225–1234. ACM (2016)
25. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. **38**(5), 1–12 (2019)
26. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**, 4–24 (2019)
27. Xia, Y., Gao, J., Cui, B.: iMap: incremental node mapping between large graphs using GNN. In: The 30th ACM International Conference on Information and Knowledge Management, Australia, 1–5 November 2021, pp. 2191–2200 (2021)
28. Xiong, H., Yan, J., Pan, L.: Contrastive multi-view multiplex network embedding with applications to robust network alignment. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1913–1923 (2021)
29. Zhang, J., Yu, P.S.: Integrated anchor and social link predictions across social networks. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, 25–31 July 2015, pp. 2125–2132. AAAI Press (2015)
30. Zhang, J., et al.: Mego2vec: embedding matched ego networks for user alignment across social networks. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 327–336 (2018)
31. Zhang, S., Tong, H., Jin, L., Xia, Y., Guo, Y.: Balancing consistency and disparity in network alignment. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2212–2222 (2021)
32. Zhang, S., Tong, H., Xia, Y., Xiong, L., Xu, J.: NetTrans: neural cross-network transformation. In: KDD 2020: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, CA, USA, 23–27 August 2020, pp. 986–996 (2020)
33. Zheng, C., et al.: Robust graph representation learning via neural sparsification. In: International Conference on Machine Learning, pp. 11458–11468. PMLR (2020)
34. Zhong, Z., Cao, Y., Guo, M., Nie, Z.: Colink: an unsupervised framework for user identity linkage. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018), pp. 5714–5721. AAAI Press (2018)
35. Zhou, F., Liu, L., Zhang, K., Trajcevski, G., Wu, J., Zhong, T.: Deeplink: a deep learning approach for user identity linkage. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 1313–1321. IEEE (2018)
36. Zhou, F., Wen, Z., Trajcevski, G., Zhang, K., Zhong, T., Liu, F.: Disentangled network alignment with matching explainability. IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, pp. 1360–1368 (2019)
37. Zhou, F., Wen, Z., Zhong, T., Trajcevski, G., Xu, X., Liu, L.: Unsupervised user identity linkage via graph neural networks. In: IEEE Global Communications Conference, GLOBECOM 2020, 7–11 December 2020, pp. 1–6. IEEE (2020)