Cross Network Representation Matching with Outliers

Mingliang Hou*, Jing Ren†, Falih Febrinanto†, Ahsan Shehzad*, and Feng Xia† (✉)
*School of Software, Dalian University of Technology, Dalian 116620, China
†School of Engineering, IT and Physical Sciences, Federation University Australia, Ballarat, VIC 3353, Australia
teemohold@outlook.com, ch.yum@outlook.com, falihgozifeb@gmail.com, ahsan.shehzad@outlook.com, f.xia@ieee.org

***Abstract*—Research has revealed the effectiveness of network representation techniques in handling diverse downstream machine learning tasks upon graph structured data. However, most network representation methods only seek to learn information in a single network, which fails to learn knowledge across different networks. Moreover, outliers in real-world networks pose great challenges to match distribution shift of learned embeddings. In this paper, we propose a novel joint learning framework, called CrossOSR, to learn network-invariant embeddings across different networks in the presence of outliers in the source network. To learn outlier-aware representations, a modified graph convolutional network (GCN) layer is designed to indicate the potential outliers. To learn more fine-grained information between different domains, a subdomain matching is adopted to align the shift distribution of learned vectors. To learn robust network representations, the learned indicator is utilized to smooth the noise effect from source domain to target domain. Extensive experimental results on three real-world datasets in the node classification task show that the proposed framework yields state-of-the-art cross network representation matching performance with outliers in the source network.**

***Index Terms*—network representation, subdomain matching, outlier, graph data, graph learning**

I. Introduction

Networks are ubiquitous in variety of real-world application scenarios. Data from many real-world domains can be represented as networks, including social networks [1], protein-protein interaction networks [2] and citation neworks [3], where nodes represent entities with attributes and links indicate the interactions or relationships between entities. More recently, there has been a surge of methods that seek to learn representations that preserve information about networks. The idea behind these representation learning methods is to learn low-dimensioal node vector representations with orginial network information being well preserved [4]. The learned representations can be used as feature inputs for downstream machine learnig tasks, e.g., classification. However, most existing network representation methods only work in a single network incapable of matching representations across different networks. Moreover, node embeddings in two networks may not fully align in a noise environment due to the existing of outliers, which affects the representation matching performance if the noise effects from outliers are not smoothed.

The goal of cross network representation matching is to learn transferable knowledge from source network to assist in solving the same task in target network by matching the learned embeddings. It is also called domain adaptive network representation learning. Matching distributions of representations learned from different networks determines the transfer learning performance for downstream tasks.

Recent studies (e.g. [3] [5] [6]) have attempted to design framework by combing network representation with domain adaptation techniques to learn transferable node representation across different networks. Though these works achieve impressive performance on learning transferable network representations, there are still three challenges as follows:

1) Outlier-aware network representation learning. Existing domain adaptive network representation learning methods are outlier-unaware framework. For example, GCN layers rely more on aggregating information from neighbors to smooth the noise effects from outliers. However, it is hard to distinguish the outliers from entangled embeddings.
2) Discriminative information mixing. Most existing domain adaptive network representation learning methods align the shift distributions in a global way. By combing with discriminator or GRL (Gradient Reversal Layer) [7], the learned target vectors can be matched with distributions of source vectors. More fine-grained information of each domain such as inter-category information may be mixed up in the global view. Moreover, the adversarial mechanism leads to the instability and slow convergence during training.
3) Robust network representation learning across different networks. In domain adaptive network representation, source domain data plays an important role in the transferable model training. Though the network representation learning aggregates information based on structural information, the abnormal attribute information from outliers still hinder the transfer learning performance. The idea behind it is that compared with structural information, attribute information is more network-invariant. Moreover, noise effects may be amplified in matching the shift distributions of two domains.

To address these challenges, in this paper, we propose the cross network representation matching (CrossOSR) framework, to jointly learned robust network-invariant embeddings across different networks. We first design a novel information aggregation mechanism in GCN (Graph Convolutional

Network) layer to learn structural and attribute information as well as indicating the unseen outliers in source network (i.e., outlier-aware). Then, a subdomain discrepancy metric is incorporated to match the inter-category fine-grained information between two domains (i.e., subdomain matching) [8]. Moreover, the learned outlier indicator is utilized to smooth the noise effects from source domain to target domain by combing with subdomain matching (i.e., robust representation learning).

The contributions of this paper are summarized as follows:

- We present a novel problem for domain adaptive network representation with outliers in source domain, and propose a joint learning framework to solve this problem.
- We propose a novel outlier-aware GCN layer which can indicate outliers as well as capturing network structural and attribute information. Furthermore, the leaned indicators can be utilized to smooth the noise effects from source domain to target domain during distribution alignment.
- We design a subdomain matching to align the embedding vectors between different networks. In contrast to adversarial and MMD (Maximum Mean Discrepancy) metric based discrepancy metrics, the cosine distance based metric is stable and fast-coverage during training.
- We evaluate our framework upon real-world datasets. The experimental results demonstrate the effectiveness of the proposed framework as compared against baseline methods.

The rest of this paper is organized as follows. A brief survey of related work is given in Section II. In Section III, we introduce the definitions and problem formulation. We present the technical details of CrossOSR in Section IV. In Section V, we evaluate the performance of CrossOSR on three real-world datasets from both quantitative and qualitative perspectives. We conclude our work in Section VI.

II. Related Work

This work is closely to network representation, domain adaptation and outlier detection. In this section, we make a brief review of related works in these areas.

A. Network Representation

The goal of network representation is to extract effective features of a given network to support the downstream network analysis tasks such as node classification and link prediction. A mapping function is utilized to encode the network into a low-dimensional vector space with original network topology and attribute information being well preserved. For topology information preserving, the proximity information such as the first-order, second-order and high-order proximity can be learned during mapping process. Deepwalk [2] samples paths of network by random walk and regards the co-occurrence nodes in a path as topology proximity context of a given node. LINE [9] defines the first-order and second-order proximity based on a node's neighbors. Node2vec [10] designs generalized random walk strategy to capture the structural equivalence between two k-hop($k \geq 3$) nodes. For attribute information, TAWD [11] combines the text description information and topology information in representing process by matrix factorization to preserve more useful attribute information into the learned vectors. ANRL [12] designs an autoencoder-enhancement framework to learning topology and attribute information in an unsupervised way. GCN [13] extends the convolutional operation on graph-structured data to extract node features by integrating neighboring attribute information and graph topology information in convolutional layers. However, most network representation models only learn node representations on a single network without generalizing the learned vectors across different networks [14].

B. Domain adaptation

Domain adaptation is a subcategory of transfer learning, which enables the learned models to perform on different but relevant domain with the shared same label space [15]. Most existing domain adaptation works are applied in the fields of Computer Vision and Natural Language Processing.

Recent studies have shown that deep neural networks can be utilized to extract more transferable features for domain adaptation [16]. The idea behind these methods is to find a matching strategy to learn more domain-invariant representations. Two categories of matching mechanisms can be identified in these works. First is the statistic moment matching based mechanism such as MMD [17] [18], CMD (Central Moment Discrepancy) [19] and second-order statistic matching [20]. Second is the adversarial matching mechanism which enforces sample representations from different domain to be nondiscriminative by minimizing the adversarial loss. Moreover, the latest advances have been achieved by subdomain adaptation [21], which is centered on learning the local domain shift to learn the correlations of inter-domain relations.

Some works have been proposed to learn generalized representations between different networks. CDNE [5] contains two staked autoencoders to learn network representations separately by matching embedding vector with conditional MMD. DANE [3] designs an unsupervised model to learn cross domain network representations by using GCN as the feature extractor. UDA-GCN [6] solves the node classification problem for unlabeled target network. Global and local relationships are learned into representations.

C. Outlier Detection

Outlier detection is one of most vital problems among networks downstream tasks due to its widely applied scenarios including cyber attack detection in computer networks, fraud detection in finance and spammers discovery in social media, etc [22] [23]. Conventional methods for network outlier detection mostly focused on the topological features of the graph to detect anomalous patterns, such as subgraph frequency, community structure [24], etc. In recent years, lots of methods attempted to discover network outliers utilizing network representation techniques. These network representation based methods, which aim to incorporate structural and attribute information of networks to find outliers or detect the anomalies. SEANO [25] designs an outlier-aware network representation framework to discover outliers in a semi-supervised way. DOMINANT [26] utilizes GCN and autoencoder as the basic feature extractor to detect anomalies by the reconstruction error. However, most existing network representation based outlier detection methods can not consider the noise effect from outliers in domain adaptive network representation [27]. More specifically, domain adaptive network representation model may suffer from the noise effects from outliers in source domain, which results in limiting the model capacity.

III. Definitions and Problem Formulation

In this section, several definitions are given, and then we formulate the problem.

A. Definitions

Definition 1: **Network representation:** Given a network, $\mathcal{N} = \{V, E, X, Y\}$, where: $V = \{v_i\}_{i=1,2,...,N}$ is the set of vertices; $e_{i,j} = (v_i, v_j) \in E$ is the set of edges; $X \in \mathbb{R}^{m\times N}$ indicates attribute information (e.g., text description or metadata) associated with vertices; $Y \in \mathbb{R}^{N\times C}$ is the set of labels and C is the number of vertex categories, the network representation aims to learn a mapping: $\Phi(\mathcal{N}) = Z, Z \in \mathbb{R}^{N\times d}$, $d \ll N$.

Definition 2: **Domain adaptive network representation:** Given a source network, $\mathcal{N}_s = \{V_s, E_s, X_s, Y_s\}$ with fully labeled set Y_s and a target network, $\mathcal{N}_t = \{V_t, E_t, X_t, Y_t\}$ with unseen labeled set Y_t, the domain adaptive network representation aims to learn a mapping $\Phi_{st}(\mathcal{N}_s) = Z_s$ and $\Phi_{st}(\mathcal{N}_t) = Z_t$, where the distributions of Z_s and Z_d should be aligned.

Definition 3: **Source domain outlier:** Given a source network, $\mathcal{N}_s = \{V_s, E_s, X_s, Y_s\}$, outlier is defined as a vertex v_{si} with attribute information that significantly deviate from v_i's neighbor's attribute information.

B. Problem Formulation

Cross network matching with outliers: Given a source network $\mathcal{N}_s$ with outliers and a target network $\mathcal{N}_t$, we aim to learn transferable robust embeddings for supporting specific tasks in target network, e.g., node classification.

In this paper, we design a unified task-specific domain adaptive network representation framework for supporting the node classification. A fully labeled source network $\mathcal{N}_s$ and an unlabeled target network $\mathcal{N}_t = \{V_t, E_t, X_t\}$ are trained by neural network. The learned target network node representations can be utilized to accurately classify nodes.

IV. Methodology

In this section, we first present the overall framework of CrossOSR. Then, detailed discussions for each component are given.

A. Overall Framework

To learn useful target network representations for supporting node classification task with outliers in the source network, we propose CrossOSR, an outlier-aware subdomain adaptive framework to learn robust network representations , as shown in Fig.1.

The first component of CrossOSR is the outlier-aware network representation. In order to learn structural and attribute information, we use GCN as the basic network feature extractor. To discover the outliers in source network, we modify the original information aggregation form from the perspective of Laplacian smoothing to discover the outliers in source network.

The second component of CrossOSR is the label information based MMD discrepancy to match node representations within the same class category. Cosine distance is utilized to measure the distribution shift of vectors in the latent embedding space. Then, the indicator learned in the first component is utilized to smooth the noise effects from source network by incorporating it with subdomain matching.

Moreover, a source domain classifier and a fuzzy target domain classifier are designed in CrossOSR to utilize source domain and target domain information for learning the task-dependent knowledge.

B. Outlier-aware Network Representation

Network representation aims to learn low-dimensional vectors with original network structural and attribute information being well preserved. GCN [13], as an effective semi-supervised network representation method, is tailored for our problem. Therefore, in this paper, we utilize GCN as the basic network representation component [28]. In GCN, each layer can be expressed as follows:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W_l), l = 1, 2, ..., L \quad (1)$$

where $\hat{A} = A+I_N$, $\hat{D}_{i,i} = \sum_j \hat{A}_{i,j}$, A is the adjacency matrix of network $\mathcal{N}$, H^l is the output of the l-th layer, $H^{(0)} = X$, σ is the activation function and W_l are the learnable parameters of the l-th layer. The final network node representation $Z = H^{(L)} = Z = \{z_i\}_{i=1,2,...,N}$.

In general, GCN can be seen as a special form of Laplacian smoothing [29]. For each vertex, the Laplacian smoothing of the input features is defined as:

$$\hat{y}_i = (1-\alpha)x_i + \alpha\sum_j \frac{\hat{a}_{i,j}}{d_j}x_j, \hat{a}_{i,j} \in \hat{A} \quad (2)$$

where $0 < \alpha < 1$ indicates the feature aggregation between current vertex v_i and its neighbors. Eq.(2) can be written as:

$$\hat{y}_i = x_i - \alpha\sum_j \hat{d}_j^{-1}(\hat{d}_j - \hat{a}_{i,j})x_j \quad (3)$$

The matrix form of Eq.(3) is:

$$\hat{Y} = X - \alpha\hat{D}^{-1}(\hat{D} - \hat{A})X = X - \alpha\hat{D}^{-1}\hat{L}X \quad (4)$$

Eq.(4) is the Laplacian smoothing. The normalized Laplacian $\hat{D}^{-1}\hat{L}$ can be replaced by the symmetrically normalized

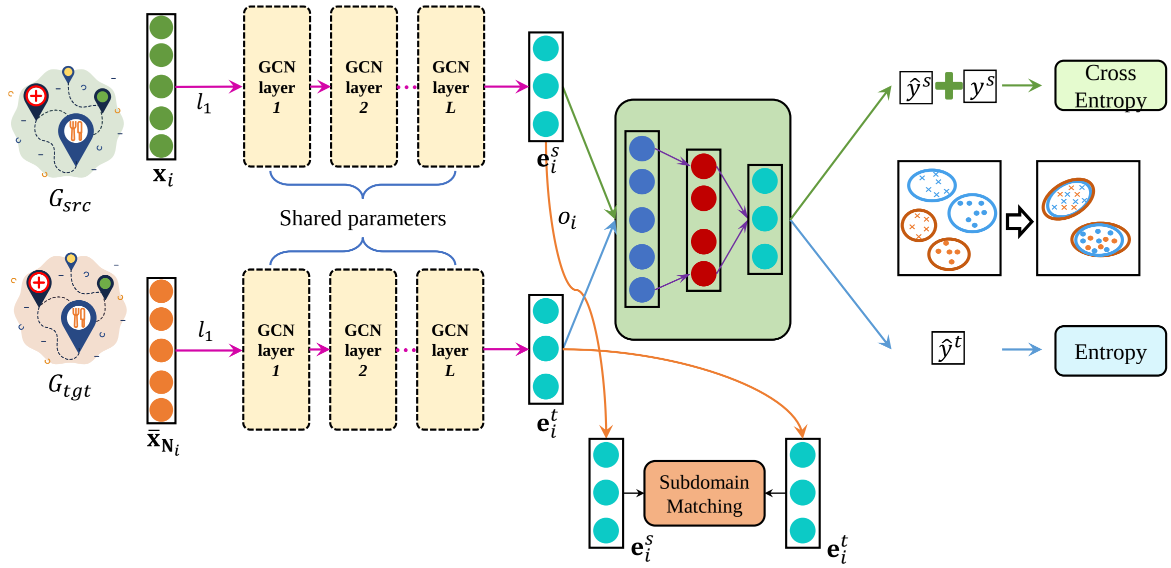


Fig. 1: The framework of CrossOSR.

Laplacian $\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$ and α can set to 1, then the Eq.(4) is equal to the convolution operation. Therefore, GCN is a special form of Laplacian smoothing [30].

From the Eq.(4), we can find that the α determines the smoothing degree of information aggregation on the network. Intuitively, if a vertex is an outlier with highly deviated attribute information compared to its neighbors, then the classification results should rely more on the attribute information of its neighbors. Therefore, in this paper, we replace the convolution operation in the first layer as:

$$Y = X - O\hat{D}^{-1}\hat{L}X = (I - O)X + O\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}X \quad (5)$$

where $O = \{o_i\}_{i=1,2,...,N}$ is trainable weight for each source vertex that determines information smoothing degree between itself and neighbors. As a consequence, if a vertex is an outlier, our modification will enforce the neural network to learn a large value (rely more on attribute information of its neighbors). Therefore, in our modification of GCN, the learned weight O can be used as an outlier indicator for smoothing the noise effects for target node representation. For the target network without outliers, we set o_i as 1.

C. Subdomain Network Representation Matching

After representing source network and target network into a common latent feature space by shared GCN weights, the distribution shift between z_s and z_t limits the knowledge transferring from source network to target network. In this paper, inspired by the previous subdomain methods [21] [31], we split the source domain and target domain into different subdomains based on the label-information.

If we assume the source and target data are S and T, the loss of domain adaptation can be defined as the distance between these :

$$\mathcal{L}_{ST} = D(S, T) \quad (6)$$

where $D(\cdot)$ refers to the distance metric to measure the discrepancy of S and T. The commonly-used metrics are adversarial loss and MMD loss. The adversarial loss use discriminator or gradient reversal layer (GRL) to finish the minimax training as:

$$D(\mathbb{E}_{P_s}, \mathbb{E}_{P_t}) \triangleq \mathbb{E}_{x\sim P_s}[log\mathcal{A}(x)] + \mathbb{E}_{x\sim P_t}[log(1 - \mathcal{A}(x))] \quad (7)$$

where P_s and P_t are distributions of S and T. The $\mathcal{A}$ is the adversarial function to maximize the discrepancy between source domain and target domain. $\mathcal{A}$ is implemented by a discriminator or reversal gradient. The MMD loss usually calculates the discrepancy based on the statistics of source data and target data. Formally, MMD defines the following difference as:

$$D(\mathbb{E}_{P_s}, \mathbb{E}_{P_q}) \triangleq ||\mathbb{E}_{x\sim P_s}[\mu(x)] - \mathbb{E}_{x\sim P_t}[\mu(x)]||^2_{\mathcal{H}} \quad (8)$$

where $\mu(\cdot)$ is the mapping function to transform the original samples to $\mathcal{H}$, the Reproducing Kernel Hillbert Space (RKHS) [31].

The domain adaptive network representation match distributions between learned vectors from different networks in the latent embedding space, $\mathcal{Z}$. In this paper, we focus on matching this shift by accurately aligning distribution of the relevant subdomains with the same label category in feature latent space, namely, conditional distribution matching [5]. Therefore, the $D(z_s, z_t)$ is defined as:

$$D_{\mathcal{Z}}(S, T) \triangleq D_{\mathcal{Z}}(z_s^c, z_t^c), z_s^c \sim P_s^c, z_t^c \sim P_t^c \quad (9)$$

To enable a stable and fast-coverage training, we choose MMD as the distance metric:

$$D_{\mathcal{H}}(S, T) \triangleq \mathbb{E}_c||\mathbb{E}_{x\sim P_s^c}[\mu(x)] - \mathbb{E}_{x\sim P_t^c}[\mu(x)]||^2_{\mathcal{H}} \quad (10)$$

Therefore, the distance of relevant subdomains can be defined as [31]:

$$\hat{D}_{\mathcal{H}}(S,T) = \frac{1}{C}\sum_{c=1}^{C} || \sum_{z_i^s \in Z_s} w_i^{sc}\mu(z_i^s) - \sum_{z_j^s \in Z_t} w_j^{tc}\mu(z_j^t)||_{\mathcal{H}}^2 \tag{11}$$

where w_i^c denote the weight of vector z belonging to label c. The label set $Y_s \in \mathbb{R}^{C\times 1}$ is represented by a vector to indicate the category distribution of x_i. In source network, y_i is one-hot vector. Therefore, the w_i^c is computed as:

$$w_i^c = \frac{y_{ic}}{\sum_{(x_j,y_j)\in\mathcal{N}} y_{jc}} \tag{12}$$

where y_c is the cth entry of y_i. In the target network, the y_i can not be seen during training. Therefore, the fuzzy label, namely, the predicted value $\hat{y}_i$ is utilized to compute w_j^{tc}.

In computing Eq.(11), the kernel function is utilized to measure the discrepancy of two embedding vectors in RKHS. However, for graph-structured data with large samples, kernel function such as Gaussian kernel [32] requires a high computational cost. In CrossOSR, we utilize cosine distance to measure the discrepancy between two embeddings.

D. Robust Representation Learning

Though the GCN layer smooths the noise effect from source data. The learned vectors of outliers still diverge other normal vectors in latent space. This divergence will be amplified in matching the distributions from source to target, which hinders the overall performance on the specific downstream task. Therefore, to alleviate this effects from outliers, the learned indicator is combined with the subdomain matching to learn a more robust embeddings for the target network. Therefore, the Eq.(11) is modified as:

$$\begin{aligned}\hat{D}_{\mathcal{Z}}(S,T) = \frac{1}{C}\sum_{c=1}^{C}[&\sum_{i=1}^{N_s}\sum_{j=1}^{N_s} w_i^{sc}w_j^{sc}(1-o_i) < z_i^s, z_j^s > \\ &+ \sum_{i=1}^{N_t}\sum_{j=1}^{N_t} w_i^{tc}w_j^{tc} < z_i^t, z_j^t > \\ &- 2\sum_{i=1}^{N_s}\sum_{j=1}^{N_t} w_i^{sc}w_j^{tc}(1-o_i) < z_i^s, z_j^t >]\end{aligned} \tag{13}$$

Intuitively, o_i can indicate the possibility of an outlier. In Eq.(13), the potential negative effects of outliers will be alleviated by o_i.

V. Experiments

In this section, we first introduce the experiment settings. Then, we demonstrate the effectiveness of the proposed framework CrossOSR.

A. Experiment Settings

1) Datasets: We conduct experiments on three real-world citation networks. These three citation networks are constructed from ArnetMiner [33]. Citationv1, DBLPv7 and ACMv9 contains the paper citation information from different sources. In these three citation networks, node represents paper and edge indicates the citation relations among papers. Title information is utilized to construct attribute information. The detailed information of datasets can founded in Table I.

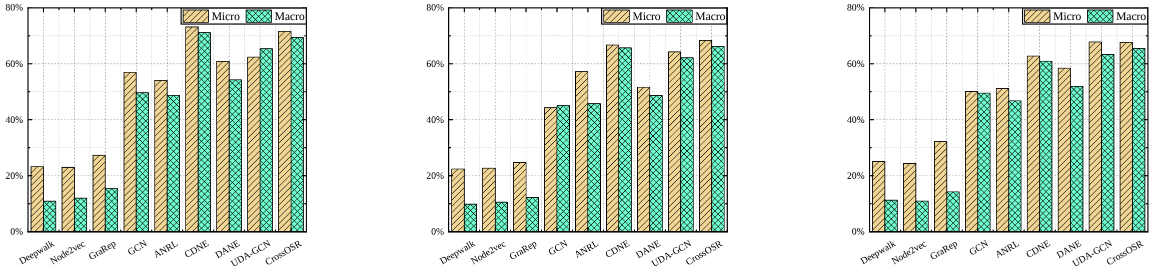
TABLE I: Statistics of the datasets used in CrossOSR.

Datasets	$\|V\|$	$\|E\|$	$\|X\|$	$\|C\|$
Citationv1	8935	15113	5379	5
DBLPv7	5484	8130	4412	5
ACMv9	9360	15602	5571	5

2) Baselines: To obtain a fair comparison and demonstrate the effectiveness of our proposed framework, in this paper, we use the following three classes methods as baselines:

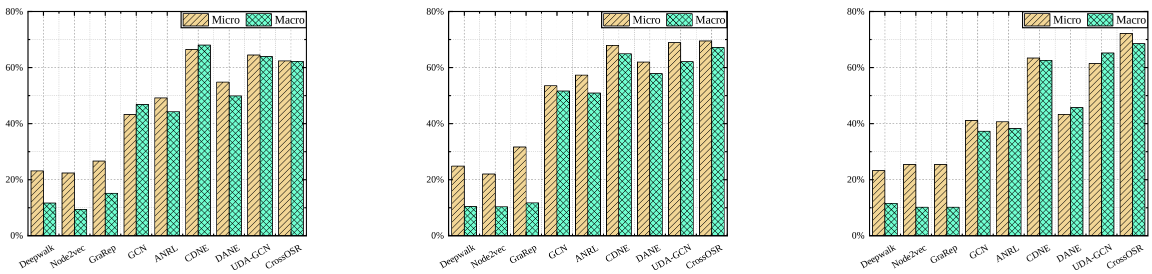
- **Structures based network representation methods for single network**: Deepwalk [2] employs random walk sampling and skip-gram model to learn node vectors based on the local structural information. Node2vec [10] extends the random walk sampling with bias to explore more generalized structural information for each node. Deepwalk [2] and Node2vec [10] are two random walk based methods, which learns the only structural information on graphs. GraRep [34] constructs a PPMI (Positive Pointwise Mutual Information) matrix to learn the global structural information of graph.
- **Structures and attributes representation methods for single network**: GCN [13] aggregates graph structural and attributed information though graph convolutional operation. ANRL [12] designs a neighbor enhancement autoencoder that combined with attribute-aware skip-gram to learn robust node representation.
- **Domain adaptive network representation methods** CDNE [5], DANE [3] and UDA-GCN [6] are three transfer learning frameworks on graph-structured data.

3) Implementation details: In this paper, all methods are implemented in Pytorch [35]. For all methods, the embedding size is set to 128. The label set is available in source network and unavailable in target network. For each method, the parameters is tuned to be optimal. For Deepwalk, Node2vec and GraRep, we do not inject any outliers since these three methods only capture the structural information. The random walk length and window size are 100 and 10. The p and q are set to 0.5 and 2 in Node2vec. The PPMI order in GraRep is 4. For GCN and ANRL, the source network representation and target network representation share with the same neural networks, and the source network is given with fully labeled data. For CDNE, we remove the 5% labeled data in target network. DANE is an unsupervised method, in this paper, we add the source classifier and target classifier for a fair comparison. For CrossOSR, the adaptation parameter is set



(a) ACMv9→Citationv1 (b) Citationv1→ACMv9 (c) ACMv9→DBLPv7

Fig. 2: Node classification results on the tasks of ACMv9→Citationv1,Citationv1→ACMv9 and ACMv9→DBLPv7.



(a) DBLPv7→ACMv9 (b) Citationv1→DBLPv7 (c) DBLPv7→Citationv1

Fig. 3: Node classification results on the tasks of DBLPv7→ACMv9, Citationv1→DBLPv7 and DBLPv7→Citationv1.

as [6] [31]. The representation neural network consist of two GCN layers and the λ_1 is set to 0.2 and the λ_2 is set to 1.

TABLE II: Node classification results on six cross-domain tasks with 10% outliers in source network.

$\mathcal{N}_s \rightarrow \mathcal{N}_t$	F1(%)	GCN	ANRL	CDNE	DANE	UDA-GCN	CrossOSR
ACMv9→Citationv1	Micro	53.34	52.15	67.09	56.06	57.23	**69.29**
	Macro	43.47	42.22	64.07	46.40	49.54	**66.95**
Citationv1→ACMv9	Micro	40.29	39.37	60.63	50.96	58.78	**66.01**
	Macro	41.99	43.72	59.84	49.60	47.79	**64.75**
ACMv9→DBLPv7	Micro	47.14	48.98	61.08	55.94	64.32	**67.69**
	Macro	45.75	44.33	59.19	51.71	54.20	**64.67**
DBLPv7→ACMv9	Micro	42.12	47.74	60.69	44.74	61.60	**62.68**
	Macro	38.44	37.65	60.57	40.68	44.32	**62.40**
Citationv1→DBLPv7	Micro	48.33	54.31	61.34	62.67	55.90	**69.17**
	Macro	47.56	48.65	60.54	59.40	63.20	**66.64**
DBLPv7→Citationv1	Micro	38.99	39.11	61.76	57.10	57.32	**71.20**
	Macro	36.28	36.77	59.11	49.76	54.38	**68.25**

4) Evaluation metrics: In this paper, we aim to learn the robust representations. Therefore, to make a fair comparison, after having obtained the node representations, we use source network representations to retrain a SVM (Support Vector Machines) classifier and the target network representations are taken to test the performance. We repeat this process 10 times, and show the average performance to demonstrate the robustness of learned representations. Macro-F1 and Micro-F1 are two metrics to evaluate the classification results.

B. Results and Analysis

1) Classification performance comparison: Fig.2 and Fig.3 show the classification results on six cross-domain tasks without outliers and we can achieve the following observations:

- The Deepwalk, Node2vec and GraRep achieve the worst performance among these baselines. The reason behind it is that these three methods only preserve the structural information. Though GraRep captures higher order proximity than Deepwalk and Node2vec, the matrix factorization technique utilized in GraRep hinders its performance when the network is sparse.
- GCN and ANRL performs better than three structures based methods. Both of them learn the structural and attribute information in an end-to-end framework.
- Three domain adaptive network representation methods outperform all the single network representation methods. CDNE beats other two methods on six cross-domain tasks. The reason is that CDNE add penalty on the reconstruction error of autoencoder by smoothing the sparsity of the attribute information. In this paper, attribute information is extracted from title, which is more sparser than the abstract information used in DANE and UDAGCN.
- CrossOSR performs better than most of baseline methods. Compared to single network representation learning methods, our domain adaptive network representation framework improve the performance significantly. Compared with DANE and UDAGCN, the subdomain matching strategy learns more fine-grained inner-domain correlations. Compared with CNDE, CrossOSR can achieve state-of-art performance without any outliers in source network.

Table II lists the classification results on six cross-domain tasks with 10% outliers in source network. By injecting the random outliers in the source network, we can conclude that:

- In single network representation, noise effects from outliers hinders the performance of models in classification tasks. The ANRL have worse performance than GCN since ANRL is a semi-supervised learning framework. Abnormal attribute information learned in the vectors disturbs the performance on the classification task.

- The performance of three domain adaptive network representation methods is effected by outliers in the source domain. CDNE achieve better performance compared to DANE and UDA-GCN due to the adversarial mechanism utilized in DANE and UDA-GCN may hinder the shift alignment. More specifically, abnormal attribute information learned source vectors may be classified as the target network embeddings in the adversarial process.
- CrossOSR beats all the baselines on six cross-domain tasks. It demonstrates that our proposed method can really smooth the noise effects from outliers in the source domain and learn more robust representations across different networks.

2) Variants analysis: CrossOSR contains three key components: (1) Outlier-aware GCN network representation; (2) Subdomain network representation matching; (3) Robust representation learning. In this subsection, we compare CrossOSR to the following variants to demonstrate the effectiveness of each component:

- **Cross-S**: Outlier-aware GCN layer and Robust representation learning work together to indicate the outliers and then smooth the noise effect from source domain to target domain. A variant of CrossOSR is to remove Outlier-aware GCN by setting the o_i to 1 and replace the $\mathcal{L}_{st}$ with Eq.(13).
- **Cross-O**: A variant of CrossOSR with the Subdomain matching is replaced by a global matching strategy with MMD as previous methods [17].

The ablation test results are listed in the Table III. We can observe that CrossOSR outperforms Cross-S and Cross-O on six cross-domain tasks with 10% outliers in the source network.

Compared to Cross-S, CrossOSR have better performance which confirms the usage of Outlier-aware and Robust representation learning components can learn more robust learning representations.

Compared to Cross-O, CrossOSR have better performance which confirms the usage of Subdomain matching can learn more fine-grained information.

3) Visualization: To make a qualitative analysis of our proposed framework, we visualize the learned embeddings from source network to target network. As shown in Fig.4, we choose the Citationv1→DBLPv7 to validate the effectiveness of CrossOSR. For each method, we map the learned embeddings to 2-D space with t-SNE [36].

Compared with Fig.4 (a) and Fig.4 (d), we can observe that the subdomain matching can really align the distribution shift across different networks in the latent space. DANE and UDA-GCN achieves worse alignment results compared to CrossOSR. The cluster boundaries in DANE and UDA-GCN are unclear which demonstrates the global domain alignment may not learn the more fine-grained information among subdomains. However, in CrossOSR, some nodes (black dots) are fixed up since class imbalance between two networks. In summary, CrossOSR can generate more meaningful layout across different networks than other domain adaptive network representation methods.

TABLE III: Node classification comparisons between CrossOSR variants on six cross-domain tasks with 10% outliers in source network.

$\mathcal{N}_s \rightarrow \mathcal{N}_t$	F1(%)	Cross-S	Cross-O	CrossOSR
ACMv9→Citationv1	Micro	67.73	60.32	**69.29**
	Macro	64.88	61.21	**66.95**
Citationv1→ACMv9	Micro	64.89	61.57	**66.01**
	Macro	60.79	59.18	**64.75**
ACMv9→DBLPv7	Micro	66.99	65.21	**67.69**
	Macro	62.21	59.15	**64.67**
DBLPv7→ACMv9	Micro	61.59	59.71	**62.68**
	Macro	62.17	57.30	**62.40**
Citationv1→DBLPv7	Micro	68.15	63.22	**69.17**
	Macro	64.55	63.21	**66.64**
DBLPv7→Citationv1	Micro	67.22	69.77	**71.20**
	Macro	62.17	63.70	**68.65**

VI. Conclusion

In this paper, we study the cross network representation matching with outliers. The most existing network representation learning methods only learn node embeddings in a single network without considering the transfer learning across different networks. We presented a novel framework, CrossOSR, to learn robust representations across different networks with outliers in the source network. An Outlier-aware GCN layer learned indicators to measure the possibility of outliers. A subdomain matching strategy is employed to capture more fine-grained correlations of relevant subdomains. Finally, combined with the learned indicator and subdomain matching, noise effects from outliers in source domain can be smoothed to learn more robust network representations. Experimental results show on three real-world citation networks demonstrate the effectiveness of our proposed framework.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China under Grant No. 61872054.

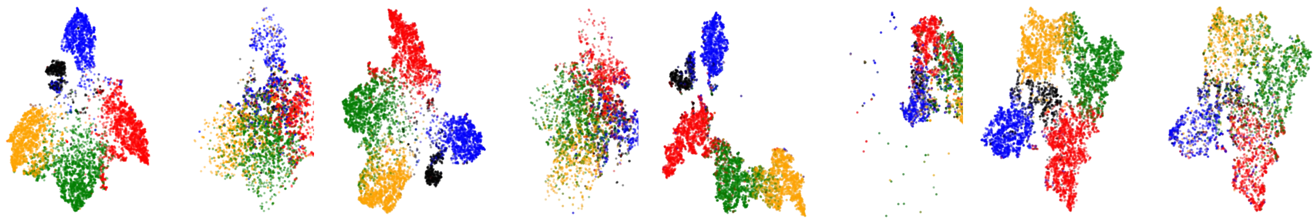
References

[1] J. Zhang, W. Wang, F. Xia, Y.-R. Lin, and H. Tong, “Data-driven computational social science: A survey,” *Big Data Research*, vol. 21, p. 100145, 2020.

[2] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[3] Y. Zhang, G. Song, L. Du, S. Yang, and Y. Jin, “Dane: Domain adaptive network embedding,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2019, pp. 4362–4368.

[4] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, “Graph learning: A survey,” *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 02, pp. 109–127, 2021.



(a) CrossOSR without subdomain matching (b) DANE (c) UDA-GCN (d) CrossOSR

Fig. 4: Visualization results on the task Citationv1→DBLPv7.

[5] X. Shen, Q. Dai, S. Mao, F.-L. Chung, and K.-S. Choi, "Network together: Node classification via cross-network deep network embedding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 1935–1948, 2021.

[6] M. Wu, S. Pan, C. Zhou, X. Chang, and X. Zhu, "Unsupervised domain adaptive graph convolutional networks," in *Proceedings of The Web Conference 2020*, 2020, pp. 1457–1467.

[7] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[8] J. Ren, F. Xia, X. Chen, J. Liu, M. Hou, A. Shehzad, N. Sultanova, and X. Kong, "Matching algorithms: Fundamentals, applications and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 3, pp. 332–350, 2021.

[9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[10] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[11] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," 2015.

[12] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, "Anrl: Attributed network representation learning via deep neural networks," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, p. 31553161.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[14] J. Liu, F. Xia, L. Wang, B. Xu, X. Kong, H. Tong, and I. King, "Shifu2: A network representation learning based model for advisor-advisee relationship mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1763–1777, 2021.

[15] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, 2020.

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3320–3328.

[17] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5989–5996.

[18] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*, 2017, pp. 2208–2217.

[19] K. Xu, S. Li, R. Li, J. Lu, and M. Zeng, "Deep domain adversarial method with central moment discrepancy for intelligent transfer fault diagnosis," *Measurement Science and Technology*, vol. 32, no. 12, p. 124005, 2021.

[20] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Computer Vision – ECCV 2016 Workshops*, 2016, pp. 443–450.

[21] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 1640–1650.

[22] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proceedings of the 2019 SIAM International Conference on Data Mining*, 2019, pp. 594–602.

[23] L. Wan, F. Xia, X. Kong, C.-H. Hsu, R. Huang, and J. Ma, "Deep matrix factorization for trust-aware recommendation in social networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 511–528, 2020.

[24] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han, "On community outliers and their efficient detection in information networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 813–822.

[25] J. Liang, P. Jacobs, J. Sun, and S. Parthasarathy, "Semi-supervised embedding in attributed networks with outliers," in *Proceedings of the 2018 SIAM International Conference on Data Mining*, 2018, pp. 153–161.

[26] Z. Peng, M. Luo, J. Li, H. Liu, and Q. Zheng, "Anomalous: A joint modeling approach for anomaly detection on attributed networks." in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3513–3519.

[27] S. Yu, F. Xia, Y. Sun, T. Tang, X. Yan, and I. Lee, "Detecting outlier patterns with query-based artificially generated searching conditions," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 134–147, 2020.

[28] H. D. Bedru, S. Yu, X. Xiao, D. Zhang, L. Wan, H. Guo, and F. Xia, "Big networks: A survey," *Computer Science Review*, vol. 37, p. 100247, 2020.

[29] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 351–358.

[30] M. Hou, J. Ren, D. Zhang, X. Kong, D. Zhang, and F. Xia, "Network embedding: Taxonomies, frameworks and applications," *Computer Science Review*, vol. 38, p. 100296, 2020.

[31] Y. Zhu, F. Zhuang, J. Wang, G. Ke, J. Chen, J. Bian, H. Xiong, and Q. He, "Deep subdomain adaptation network for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, pp. 1713–1722, 2021.

[32] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 5423–5432.

[33] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 990–998.

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

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.

[36] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.