Cross-Domain Graph Anomaly Detection

Kaize Ding, Kai Shu, Xuan Shan, Jundong Li, *Member, IEEE*, and Huan Liu, *Fellow, IEEE*

***Abstract*— Anomaly detection on attributed graphs has received increasing research attention lately due to the broad applications in various high-impact domains, such as cybersecurity, finance, and healthcare. Heretofore, most of the existing efforts are predominately performed in an unsupervised manner due to the expensive cost of acquiring anomaly labels, especially for newly formed domains. How to leverage the invaluable auxiliary information from a labeled attributed graph to facilitate the anomaly detection in the unlabeled attributed graph is seldom investigated. In this study, we aim to tackle the problem of cross-domain graph anomaly detection with domain adaptation. However, this task remains nontrivial mainly due to: 1) the data heterogeneity including both the topological structure and nodal attributes in an attributed graph and 2) the complexity of capturing both invariant and specific anomalies on the target domain graph. To tackle these challenges, we propose a novel framework COMMANDER for cross-domain anomaly detection on attributed graphs. Specifically, COMMANDER first compresses the two attributed graphs from different domains to low-dimensional space via a graph attentive encoder. In addition, we utilize a domain discriminator and an anomaly classifier to detect anomalies that appear across networks from different domains. In order to further detect the anomalies that merely appear in the target network, we develop an attribute decoder to provide additional signals for assessing node abnormality. Extensive experiments on various real-world cross-domain graph datasets demonstrate the efficacy of our approach.**

***Index Terms*— Anomaly detection, attributed graphs, domain adaptation, graph neural networks (GNNs).**

I. INTRODUCTION

ATTRIBUTED graphs are a type of graphs that not only model the attributes of each data instance but also encode the inherent dependencies among them. They have been widely used to model complex systems, such as social media networks [1], academic graphs [2], and financial transaction networks [3]. However, anomalous nodes—whose patterns significantly deviate from the majority—can be rampant in attributed graphs and cause real-world societal effects. For example, spammers in social networks can coordinate among themselves to launch various attacks, such as spreading ads to generate sales, disseminating pornography, viruses, and phishing [4]; fraud behaviors in financial networks may lead to huge financial loss for both customers and merchants [5]. Therefore, it is critical to detect anomalies on attributed graphs.

For a real-world anomaly detection system, it is often unrealistic to obtain abundant labeled data for every domain (e.g., *Hotels* and *Restaurants* are two different domains in Yelp) due to the expensive labeling cost [6], [7]. As such, graph anomaly detection is commonly performed in the single-domain setting, and unsupervised methods are proposed to handle those unlabeled domains [3]. However, the performances of unsupervised approaches may be limited without any supervision information. Thus, when the target graph is from an unlabeled domain, it is natural and important to explore the auxiliary knowledge from other related domains that come from the same data platform. Specifically, we would like to investigate whether the anomaly detection performance on an unlabeled attributed graph (target graph) can be improved by leveraging another labeled attributed graph (source graph). Recent advancements on domain adaptation have shown promising results in learning domain-invariant features across domains in various research disciplines, including computer vision [8]–[10] to natural language processing [11], [12]. In light of this, we propose to tackle the novel problem of cross-domain graph anomaly detection by adapting domain discrepancies between two attributed graphs.

Despite the unprecedented success of deep domain adaptation, directly grafting it for detecting anomalies on attributed graphs is infeasible due to the following challenges. First, compared to conventional text or image data, attributed graphs are notoriously difficult to handle due to the data heterogeneity from both structure and attribute perspectives [13]. As such, applying conventional domain adaptation techniques to our problem may result in unsatisfactory results as they are not tailored for attributed graphs. Therefore, the first challenge centers around how to model two arbitrarily structured attributed graphs from different domains and learn domain-invariant node representations for detecting anomalies. Second, in order to detect anomalies on the unlabeled target graph, one straightforward solution is to train a domain-adapted classifier as existing work shows [6], [9], [14]. However, the domain-adapted classifier may render unsatisfactory anomaly detection performance. Fig. 1 shows an example of detecting anomalies on attributed graphs in the cross-domain setting. As we can see, the labeled fraudulent reviewers in the Books domain (e.g., A_1) continuously spread promotion links instead of reviewing books, which can be treated as a typical type of anomalies. Although we are able to detect the anomalies that reveal similar behaviors (i.e., *shared anomalies*) in the Clothes domain (e.g., B_1) by domain

Manuscript received November 30, 2020; revised June 22, 2021; accepted August 14, 2021. Date of publication October 1, 2021; date of current version June 2, 2022. This work was supported in part by ONR under Grant N00014-21-1-4002. *(Corresponding author: Kaize Ding.)*

Kaize Ding and Huan Liu are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: kaize.ding@asu.edu; huan.liu@asu.edu).

Kai Shu is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: kshu@iit.edu).

Xuan Shan is with Kuaishou Technology Company Ltd., Beijing 100085, China (e-mail: shanxuan@kuaishou.com).

Jundong Li is with the Department of Electrical and Computer Engineering and the Department of Computer Science, University of Virginia, Charlottesville, VA 22904 USA (e-mail: jundong@virginia.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3110982.

Digital Object Identifier 10.1109/TNNLS.2021.3110982

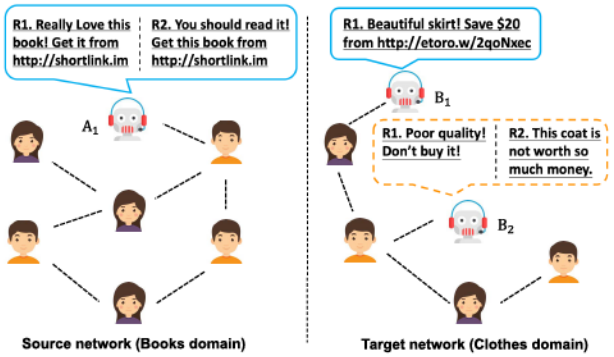


Fig. 1. Example of cross-domain graph anomaly detection. A_1 and B_1 can be considered as the *shared anomalies* since they show similar behaviors across two graphs from different domains, while B_2 is an instance of *unshared anomalies* since such type of anomalies only exists in the target graph.

adaptation, domain B has another type of fraudulent reviewers who generate negative reviews to sabotage the reputation of targeted products (e.g., B_2). The domain-adapted classifier may not work well for detecting such type of anomalies (i.e., *unshared anomalies*) since they do not appear in the source domain graph. Therefore, the second challenge lies in how to spot both the *shared* and *unshared anomalies* on the target graph simultaneously.

In this article, we propose cross-domain anomaly detection on attributed networks (COMMANDER), a novel end-to-end framework that consists of four principled components to address the above challenges. For the first challenge, COMMANDER employs a shared graph attentive encoder building on top of the graph attention networks [15] to learn node representations of both source and target attributed graphs. Meanwhile, by deceiving the domain discriminator to distinguish the domain assignment of nodes, the graph attentive encoder gradually maps node representations from both source and target graphs to a domain-invariant feature space. For the second challenge, COMMANDER can detect the *shared anomalies* with the domain-adapted anomaly classifier trained from the labeled source graph. Meanwhile, COMMANDER uses an attribute decoder to spot the *unshared anomalies* by measuring the attribute reconstruction error of each node. As such, the synergistic collaboration between anomaly classifier and attribute decoder empowers COMMANDER to achieve superior anomaly detection performance on the target graph. To summarize, our contributions of this study are as follows.

1) *Problem:* To the best of our knowledge, we are the first to study the novel problem of cross-domain graph anomaly detection. In particular, we emphasize its importance and give a formal problem definition.
2) *Algorithm:* We develop an end-to-end framework for cross-domain graph anomaly detection. The proposed framework bridges the domain discrepancy between two attributed graphs and detects both the shared and unshared anomalies on the target graph.
3) *Evaluation:* We perform extensive experiments on real-world datasets to verify the effectiveness of our proposed model. The experimental results demonstrate its superior performance for cross-domain graph anomaly detection.

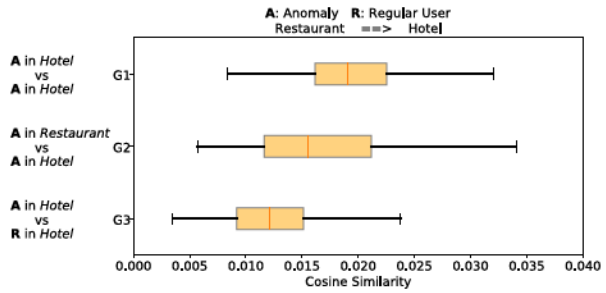


Fig. 2. Cross-domain data analysis *w.r.t.* feature similarity between different user groups.

II. PROBLEM DEFINITION

To legibly describe the studied problem, we follow the commonly used notations throughout this article. Specifically, we use lowercase letters to denote scalars (e.g., λ), boldface lowercase letters to denote vectors (e.g., $\mathbf{x}$), boldface uppercase letters to denote matrices (e.g., $\mathbf{X}$), and calligraphic fonts to denote sets (e.g., $\mathcal{V}$).

Given an attributed graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ denotes the set of nodes $\{v_1, v_2, \ldots, v_n\}$ and $\mathcal{E}$ denotes the set of edges $\{e_1, e_2, \ldots, e_m\}$, d-dimensional attributes of n nodes are denoted by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times d}$. Therefore, the attributed graph can also be represented as $G = (\mathbf{X}, \mathbf{A})$ for simplicity. Here, $\mathbf{A} = \{0, 1\}^{n \times n}$ is an adjacency matrix where $\mathbf{A}_{i,j} = 1$ indicates that there is an edge between node v_i and node v_j; otherwise, $\mathbf{A}_{i,j} = 0$.

In order to provide more interpretable results, graph anomaly detection is commonly considered as a ranking problem [3], [13]. Accordingly, we define the problem of cross-domain graph anomaly detection as follows.

Problem 1 (Cross-Domain Graph Anomaly Detection): Given a labeled attributed graph $G^s = (\mathbf{X}^s, \mathbf{A}^s)$ from the source domain and another unlabeled attributed graph $G^t = (\mathbf{X}^t, \mathbf{A}^t)$ from the target domain, here, we follow previous works and assume that G^s and G^t share the same feature space but do not have overlapped nodes or edges. The objective is to learn an anomaly detection model, which is capable of generalizing the knowledge from the labeled graph G^s, to detect the anomalies on the target graph $\mathbf{G}^t$. Ideally, anomalous nodes should be ranked on higher positions over normal nodes in the returned list.

III. PRELIMINARIES

A. Anomaly Analysis Across Domains

To gain insight into the relations between anomalies in a single domain or across different domains, we conduct an initial exploration on a pair of real-world datasets covering two different domains (i.e., *Hotel* and *Restaurant*) in Yelp (the details of the datasets are introduced in Section V-A). There are regular users and anomalies in both domains. In this analysis, we regard *Hotel* as our target domain for which we want to detect anomalies. As shown in Fig. 2, we compare the cosine similarity between different user pairs. Note that each user is represented with a feature vector constructed with the bag-of-word features from all his/her reviews. For group 3 (G3), we calculate the similarity between each anomaly and all the regular users in *Hotel* and show the average value

for each anomaly. Compared with G1, in which we show the average similarity between each anomaly and the other anomalies in the same domain, the values in G3 are significantly smaller. Such discrepancy between anomalies and regular users–which represent the majority of users in the platform—can be utilized for anomaly detection under the unsupervised setting. To investigate whether the labeled anomalies in the source domain (*Restaurant* in this case) can give guidance to anomaly detection in *Hotel*, we evaluate the similarities between anomalies across these two domains (shown in G2). The fact that the anomalies in *Hotel* are closer to anomalies in *Restaurant* than regular users in *Hotel* demonstrates that the supervised information from the source domain (*Restaurant*) can be potentially leveraged for detecting anomalies in the target domain (*Hotel*). However, the values of similarity in G2 are still smaller than those in G1, meaning that there exist some anomalies in *Hotel* revealing unshared patterns compared with anomalies in *Restaurant*. We observe similar data patterns in other pairs of cross-domain datasets, which motivates our design of COMMANDER.

B. Graph Neural Networks (GNNs)

Recently, GNNs have demonstrated their remarkable performance in different graph learning tasks [16]–[19]. The early proposed GNNs extend the operation of convolution on graph-structured data in the spectral domain for network representation learning. In the meantime, many prevailing GNN models that follow the neighborhood aggregation strategy have been proposed and are analogous to the Weisfeiler–Lehman (WL) graph isomorphism test. Specifically, the representation of a node is computed by iteratively aggregating representations of its local neighbors. Formally, a GNN layer can be defined as

$$\begin{aligned}\mathbf{h}_i^l &= \textsc{Transform}^l\left(\mathbf{h}_i^{l-1}, \mathbf{h}_{\mathcal{N}_i}^l\right)\\ \mathbf{h}_{\mathcal{N}_i}^l &= \textsc{Aggregate}^l\left(\left\{\mathbf{h}_j^{l-1} \,\middle|\, \forall j \in \mathcal{N}_i\right\}\right)\end{aligned} \tag{1}$$

where $\mathbf{h}_i^l$ is the node representation of node i at layer l and $\mathcal{N}_i$ is the local neighbor set of node i. AGGREGATE and TRANSFORM are two key functions of GNNs and have a series of possible implementations [15], [16], [20].

By stacking multiple GNN layers, the learned node representations are able to capture the long-range node dependencies in the input graph, which mitigates the network sparsity issue beyond the observed links among nodes.

IV. PROPOSED APPROACH

In this section, we present the details of the proposed framework that consists of four dedicated components (see Fig. 3): 1) a graph attentive encoder; 2) a domain discriminator; 3) an anomaly classifier; and 4) an attribute decoder. Specifically, COMMANDER accomplishes domain adaptation on attributed graphs with the graph attentive encoder and domain discriminator. The anomaly classifier and attribute decoder are employed to detect anomalies on the target attributed graph synergistically.

A. Domain Adaptation on Attributed Graphs

Deep domain adaptation has recently drawn much attention with the booming development of deep neural networks (DNNs). Those deep domain adaptation methods have been proven to be effective in different learning tasks, such as image classification, sentiment classification, and text matching [6], [9]. The main intuition behind these methods is to learn the domain-invariant representations of combined samples from both source and target domains. In order to perform cross-domain anomaly detection on attributed graphs, we propose to follow a prevalent line of study [9], [21], [22] and first employ a shared encoder to extract the latent representation of each node in both G^s and G^t. However, apart from the image or text data that we can directly feed the combined samples from both source and target domains into a shared feature extractor, different attributed graphs have distinctive topological structures. Thus, it is unclear that how we can model two arbitrarily structured attributed graphs using a shared encoder.

1) Graph Attentive Encoder (Enc): To counter this problem, we build our shared encoder grounded on the graph attention networks (GATs) [15]. GAT is an attention-based GNN model that allows specifying fine-grained weights when aggregating information from neighbors (as shown in Fig. 3). Formally, in each layer l, node v_i integrates the features of neighboring nodes to obtain representations of layer $l+1$ via

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i \cup v_i} \alpha_{ij} \mathbf{W}\mathbf{h}_j^{(l)}\right) \tag{2}$$

where σ denotes the nonlinear activation function (e.g., ReLU), $\mathcal{N}_i$ denotes the set of neighbors for v_i, and α_{ij} is the attention coefficient between node v_i and node v_j, which can be computed as

$$\alpha_{ij} = \frac{\exp\left(\sigma\left(\mathbf{a}^{\mathrm{T}}[\mathbf{W}\mathbf{h}_i^{(l)} \oplus \mathbf{W}\mathbf{h}_j^{(l)}]\right)\right)}{\sum_{k \in \mathcal{N}_i \cup v_i} \exp\left(\sigma\left(\mathbf{a}^{\mathrm{T}}[\mathbf{W}\mathbf{h}_i^{(l)} \oplus \mathbf{W}\mathbf{h}_k^{(l)}]\right)\right)} \tag{3}$$

where $\oplus$ is the concatenation operation and the attention vector $\mathbf{a}$ is a trainable weight vector that assigns importance to the different neighbors of node v_i, allowing the model to highlight the features of the important neighboring node that is more task-relevant.

The benefits of using graph attention networks are mainly twofold.

1) Graph attention networks employ a trainable aggregator function to learn the representation of each node, which eliminates the dependency on the global graph structure. In this way, our shared encoder is capable of learning node representations for both G^s and G^t [15].
2) Since malicious users might build spurious connections with normal users to camouflage their noxious intentions, graph attention networks can better assess the abnormality of each node by specifying fine-grained attention on the neighboring nodes.

Thus, the graph attentive encoder is able to learn high-quality node representations from the two attributed graphs G^s and G^t.

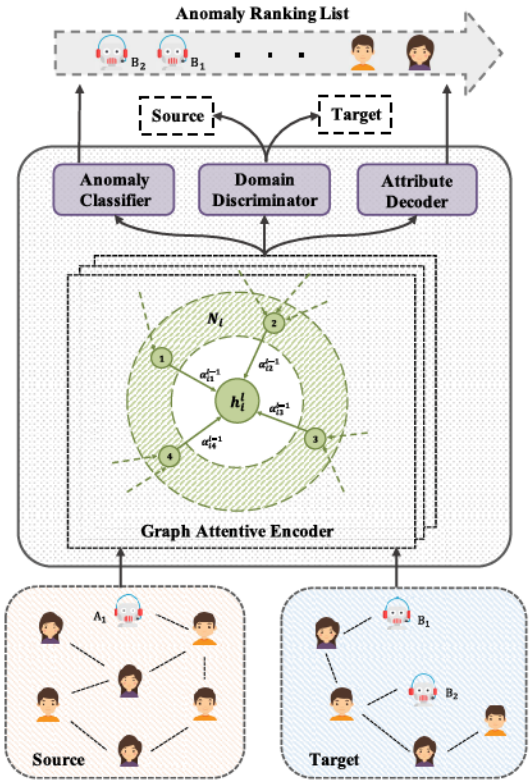


Fig. 3. Overview of the COMMANDER framework for cross-domain graph anomaly detection. Figure is best viewed in color.

Moreover, we build the graph attentive encoder Enc with multiple GAT layers

$$\mathbf{h}_i^{(1)} = \sigma\left(\sum_{j\in\mathcal{N}_i\cup v_i} \alpha_{ij}^{(1)}\mathbf{W}^{(1)}\mathbf{x}_j\right)$$
$$\cdots$$
$$\mathbf{z}_i = \sigma\left(\sum_{j\in\mathcal{N}_i\cup v_i} \alpha_{ij}^{(L)}\mathbf{W}^{(L)}\mathbf{h}_j^{(L-1)}\right) \tag{4}$$

where $\mathbf{z}_i$ is the latent representation of node i. In this way, the graph attentive encoder Enc can capture the nonlinearity of topological structure and nodal attributes. Following previous domain adaptation works [23], [24], we use Enc as a shared architecture and encodes G^s and G^t one by one in each epoch. In this way, the graph attentive encoder is able to map the learned node representations from two graphs to an aligned embedding space and further enables knowledge transfer across graphs from different domains.

2) Domain Discriminator (Dis): In order to further perform domain adaptation on two attributed graphs from different domains, we adopt the idea of adversarial machine learning [25] to perform adversarial domain adaptation [14], [26] in a two-player minimax game. As shown in Fig. 3, the first player is the domain discriminator Dis, which tries to distinguish whether an embedded node is from the source domain or the target domain, and the second player is the graph attentive encoder Enc, which is adversarially trained to deceive the domain discriminator. The domain discriminator Dis is built with a feed-forward layer with tanh nonlinearity, followed by a sigmoid function:

$$\mathbf{o}_i^D = \tanh\left(\mathbf{W}^D\mathbf{z}_i + \mathbf{b}^D\right)$$
$$\hat{y}_i = \text{sigmoid}\left(\mathbf{u}^{\mathrm{T}}\mathbf{o}_i^D\right) \tag{5}$$

where $\mathbf{W}^D$ and $\mathbf{b}^D$ denote the trainable parameter matrix and bias, respectively, and $\mathbf{o}_i^D$ is the output of the feed-forward layer. Here, $\mathbf{u}$ is another trainable weight vector and $\hat{y}_i$ is the predicted domain label. The adversarial domain loss can be mathematically formulated as

$$\mathcal{L}_D = -\frac{1}{N_D}\sum_{i=1}^{N_D}\left[d_i \log \hat{y}_i + (1-d_i)\log(1-\hat{y}_i)\right] \tag{6}$$

where N_D denotes the number of all the nodes in both G^s and G^t. Here, d_i represents the domain label of node i and $\hat{y}_i$ is the predicted domain label.

Since our goal is to bridge the domain discrepancy between two graphs, here, we choose to maximize the above cross-entropy loss. In other words, after the feature encoding phase, the domain label of nodes would not be accurately recognized by the domain discriminator, and the shared graph attentive encoder would be able to extract domain-invariant node representations from both source graph G^s and target graph G^t.

B. Cross-Domain Anomaly Detection

In Section IV-A, we have discussed how to bridge the domain discrepancy between two attributed graphs from different domains. This section introduces how to detect both *shared anomalies* and *unshared anomalies* on the target graph G^t.

1) Anomaly Classifier (Clf): Following the idea of other domain adaptation learning tasks [22], we train an anomaly classifier *Clf* right after the shared graph attentive encoder to distinguish whether a node from G^s is an anomaly or not. *Clf* is built with a feed-forward layer with tanh nonlinearity, followed by a sigmoid function:

$$\mathbf{o}_i^C = \tanh\left(\mathbf{W}^C\mathbf{z}_i + \mathbf{b}^C\right)$$
$$\bar{y}_i = \text{sigmoid}\left(\mathbf{v}^{\mathrm{T}}\mathbf{o}_i^C\right) \tag{7}$$

where $\mathbf{W}^C$ and $\mathbf{b}^C$ are the trainable parameter matrix and bias and $\mathbf{v}$ is a trainable weight vector. Specifically, the anomaly classification loss can be defined as the binary cross entropy

$$\mathcal{L}_C = -\frac{1}{N_C}\sum_{i=1}^{N_C}\left[y_i \log \bar{y}_i + (1-y_i)\log(1-\bar{y}_i)\right] \tag{8}$$

where N_C denotes the number of nodes sampled from the labeled graph G^s and y_i and $\bar{y}_i$ denote the ground truth anomaly label and the predicted anomaly label of node i, respectively. Note that here, we sample an equal number of normal nodes and abnormal nodes from G^s for addressing data imbalance. The shared graph attentive encoder maps data from different domains to a domain-invariant feature space by deceiving the domain discriminator, and then, the domain-adapted anomaly classifier can be directly used for detecting the *shared anomalies* on the target attributed graph.

Nevertheless, one critical issue is that not all anomalies share similar characteristics across graphs from different domains. As discussed in the previous sections, some specific types of anomalies that exist in G^t may not appear in G^s. Thus, solely relying on a classifier trained on the labeled source graph cannot accurately trace such *unshared anomalies*, rendering unsatisfactory anomaly detection performance on the target attributed graph.

2) Attribute Decoder (Dec): As suggested by recent studies [13], [27], [28], the reconstruction error between original data and estimated data is a strong indicator to show the abnormality of each data instance. The intuition is that anomalies usually cannot be well reconstructed from the observed data and have large reconstruction errors since their patterns deviate significantly from the majority. Therefore, we build an attribute decoder *Dec* following the graph attentive encoder for reconstructing two attributed graphs. Since node dependency information is inherently encoded in each GAT layer, we propose to reconstruct the node attributes for simplicity. Specifically, we build *Dec* with multiple GAT layers

$$\begin{aligned} \mathbf{h}_i^{(L+1)} &= \sigma\left(\sum_{j\in\mathcal{N}_i\cup v_i} \alpha_{ij}^{(L+1)}\mathbf{W}^{(L+1)}\mathbf{z}_j\right) \\ &\cdots \\ \tilde{\mathbf{x}}_i &= \sigma\left(\sum_{j\in\mathcal{N}_i\cup v_i} \alpha_{ij}^{(\bar{L})}\mathbf{W}^{(\bar{L})}\mathbf{h}_j^{(\bar{L}-1)}\right) \end{aligned} \tag{9}$$

where $\tilde{\mathbf{x}}_i$ is the estimated attribute of node v_i. The reconstruction error computed by this deep autoencoder network provides a precise assessment of node abnormality [13], [29], [30] and enables us to spot the *unshared anomalies*. Specifically, the reconstruction loss can be defined as

$$\mathcal{L}_R = ||\widetilde{\mathbf{X}}^s - \mathbf{X}^s||_F^2 + ||\widetilde{\mathbf{X}}^t - \mathbf{X}^t||_F^2 \tag{10}$$

where $\widetilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_n]$ denotes the reconstructed attribute matrix of a graph.

In this way, our anomaly classifier and attribute decoder are able to synergistically perform anomaly detection on the target attributed graph. Intuitively, the anomaly classifier would spot the *shared anomalies* with high precision; meanwhile, the attribute decoder is capable of providing complementary insight for detecting the *unshared anomalies*. As another benefit, the incorporation of the attribute decoder can also improve the feature learning quality of the graph attentive encoder through backpropagation and relieve the overfitting problem when training the anomaly classifier [31].

C. Model Learning

So far, we have introduced the architecture of our framework COMMANDER for solving the problem of cross-domain graph anomaly detection. This joint architecture requires dedicated training objective for each component. The complete objective function can be formulated as follows:

$$\begin{aligned} \mathcal{L} &= -\mathcal{L}_D + \mathcal{L}_C + \mathcal{L}_R \\ &= \frac{1}{N_D}\sum_{i=1}^{N_D}[d_i \log \hat{y}_i + (1-d_i)\log(1-\hat{y}_i)] \\ &\quad - \frac{1}{N_C}\sum_{i=1}^{N_C}[y_i \log \bar{y}_i + (1-y_i)\log(1-\bar{y}_i)] \\ &\quad + ||\widetilde{\mathbf{X}}^s - \mathbf{X}^s||_F^2 + ||\widetilde{\mathbf{X}}^t - \mathbf{X}^t||_F^2. \end{aligned} \tag{11}$$

We summarize the training procedure of COMMANDER in Algorithm 1. By minimizing the dedicated objective functions, COMMANDER gradually closes the domain shift between G^s and G^t and learns a powerful anomaly detector. All the parameters of COMMANDER are optimized by the standard backpropagation algorithm [31]. Specifically, for each node, we use the output from *Clf* as a learned weight to reweight the reconstruction errors from *Dec*, and the final anomaly score of node v_i can be formulated as

$$\text{score}(v_i) = \bar{y}_i||\tilde{\mathbf{x}}_i - \mathbf{x}_i||_2^2 \tag{12}$$

where $\bar{y}_i \in [0, 1]$ and the final scores represent the node abnormality computed by both the anomaly classifier and the attributed decoder.

Algorithm 1 Training Process of COMMANDER

Input: G^s, G^t, N_D, N_C, α, epoch.
Output: Anomaly scores of all nodes in G^t.

```
while i < epoch do
    // Adversarial domain adaptation training
    Sample N_D nodes from G^s and G^t;
    Compute the adversarial domain loss according to Eq. (5);
    Take gradient steps and update the parameters;
    // Anomaly classification training
    Sample N_C nodes from G^s;
    Compute the anomaly classification loss according to Eq. (7);
    Take gradient steps and update the parameters;
    // Graph reconstruction training
    Compute the reconstruction loss according to Eq. (9);
    Take gradient steps and update the parameters;
Compute anomaly score of each node in G^t using Eq. (11)
```

D. Complexity Analysis

Our proposed framework COMMANDER is composed of four principled components introduced in the previous section. In particular, the graph attentive encoder and attribute decoder are built with an L-layer graph attention network [15]. As shown in [15], the time complexity of each graph attentional layer can be expressed as $O(ndd' + md')$, where d is the dimensionality of the input feature and d' is the dimensionality of output feature. For the anomaly classifier and domain discriminator, those two components are built with L' fully connected layers, and the corresponding time complexity of each fully connected layer can be expressed as $O(dd')$. As $m \gg n$ in general, the computational complexity of COMMANDER is linear with respect to the number of edges.

TABLE I
STATISTICS OF THE REAL-WORLD DATASETS

	YelpHotel ⇌ YelpRes		YelpNYC ⇌ Amazon	
# nodes	5,196	5,102	21,040	18,601
# edges	171,743	239,738	303,949	274,458
# attributes	8,000	8,000	10,000	10,000
# anomalies	250	275	1000	750

V. EXPERIMENTS

In order to verify the effectiveness of our proposed framework, in this section, we conduct empirical evaluations on various real-world attributed graph datasets.

A. Experiment Settings

1) Evaluation Datasets: To evaluate the performance of different methods, we adopt two pairs of real-world datasets for evaluation. All the datasets are public and have been widely used for graph anomaly detection problems [32], [33]. The dataset statistics are listed in Table I and we summarize the details of those two dataset pairs as follows.

1) *YelpHotel ⇌ YelpRes:* YelpHotel and YelpRes are collected from Yelp on two major business domains, i.e., hotel and restaurant, in the Chicago area [32]. For each dataset, users are considered as nodes and a link will be created if two users commented on the same hotel or same restaurant. By using the Yelp antifraud filter, the users from each dataset can be separated into two classes: anomaly (authors of filtered reviews) and regular users (authors with no filtered reviews), which can be considered as the ground truth labels.
2) *YelpNYC ⇌ Amazon:* To further study the effect of different levels of domain discrepancy on the performance improvements, we also adopt another pair of attributed graphs collected from two different platforms (domains with higher discrepancy), i.e., Yelp and Amazon. Specifically, YelpNYC collects data for the restaurants located in New York City [32]. Amazon is another attributed graph collected from an E-commerce platform in [33]. In this dataset, a user is flagged as a fraudulent user if he/she has reviewed two or more products that have been targeted by crowdsourcing efforts [33]; otherwise, the user is considered as legitimate.

For all the datasets above, we apply bag-of-words model [34] to obtain the attributes of each node. The vocabulary is built on top of the textual contents related to the nodes from both source and target graphs. With the processed datasets, we are able to conduct the evaluation across four domain shifts in our experiments, including **YelpHotel** → **YelpRes**, **YelpRes** → **YelpHotel**, **YelpNYC** → **Amazon**, and **Amazon** → **YelpNYC**. Notably, "A → B" represents the task, which aims at detecting anomalies on the target domain attributed graph B, by adapting the knowledge from the labeled source domain attributed graph A. In addition, as anomalies usually consist of a small portion of a dataset, we randomly sampled out part of the spammers or fraudulent reviewers to make our experiments more realistic and challenging.

2) Compared Methods: In the experiments, we compare the proposed framework COMMANDER with several state-of-the-art representative anomaly detection methods. Specifically, **LOF** [35] detects anomalies at the contextual level and only considers nodal attributes. **ConOut** [36] detects anomalies in the local context by determining its subgraph and its relevant subset of attributes. **AMEN** [37] uses both attribute and graph structure information to detect anomalous neighborhoods. Specifically, it analyzes the abnormality of each node from the ego-network point of view. **DOMINANT** [13] is the state-of-the-art model for detecting anomalies on attributed graphs. By developing a graph convolutional network (GCN)-based autoencoder, the reconstruction errors can be used for spotting anomalies. **ADDA** [14] is an adversarial domain adaptation model for image classification. We adopt the architecture of this model to conduct cross-domain graph anomaly detection by omitting the graph structures.

Since cross-domain graph anomaly detection remains an understudied task, it is worth mentioning that none of the above methods is exactly developed for solving our studied problem. Since no labels are available on the target graph, we first select four state-of-the-art baselines (i.e., LOF, ConOut, AMEN, and DOMINANT) for unsupervised anomaly detection on attributed graphs. We directly run each of them on the target graph and report the corresponding detection performance to make a fair comparison. In addition, we also compare with ADDA, which is a state-of-the-art domain adaptation method. As it is not designed for graph-based anomaly detection problem, we omit the topological structure and use the probability predicted by ADDA to rank all the nodes on the target graph.

3) Implementation Details: The proposed model is implemented in TensorFlow and optimized with Adam optimizer [38]. For the graph attentive encoder, we use two graph attention layers with 128 and 32 dimensions and are both activated by the ReLU function [39]. The attribute decoder is a single-layer neural network with 128 neurons, in which the ReLU function is used to activate the hidden layer and the linear function is used to activate the output layer. As for the domain discriminator, it is a single-layer neural network with 16 neurons using the tanh activation function for the hidden layer and the sigmoid activation function in its output layer. The anomaly classifier is implemented using the same way. While optimizing the attribute decoder loss $\mathcal{L}_R$, we set the learning rate to 0.001. For optimizing both the adversarial domain loss $\mathcal{L}_D$ and anomaly classification loss $\mathcal{L}_C$, we use the initial learning rate of 0.005 and reduce it to 0.001 after training for 50 epochs. We choose the parameter α with the best performance for each domain shift scenario, and the details can be found in Section IV-D. We grid search for the parameter α in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} and select 0.5 for achieving the overall best results on different datasets.

4) Evaluation Metrics: For the problem of graph anomaly detection, previous research usually considers it as a ranking problem [13], [36]. Following this line of work, we use three standard evaluation metrics to measure the performance of different anomaly detection algorithms.

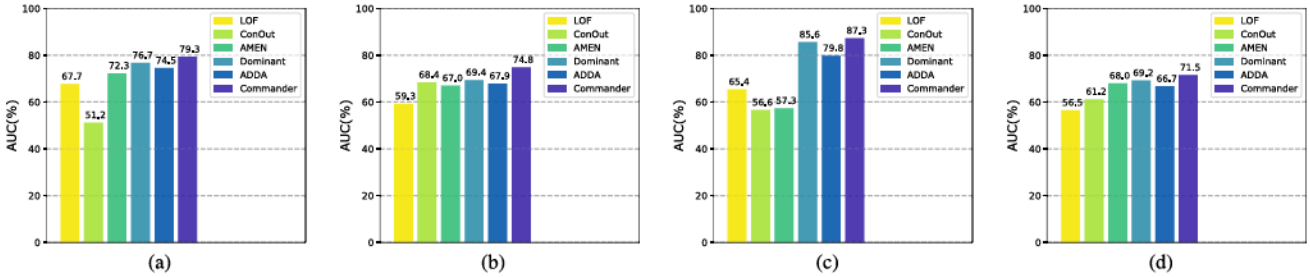


Fig. 4. Results of cross-domain graph anomaly detection w.r.t. AUC scores. (a) YelpHotel→YelpRes. (b) YelpRes→YelpHotel. (c) YelpNYC→Amazon. (d) Amazon→YelpNYC.

TABLE II

RESULTS OF CROSS-DOMAIN GRAPH ANOMALY DETECTION W.R.T. PRECISION@K

Precision@K												
	YelpHotel → YelpRes			YelpRes → YelpHotel			YelpNYC → Amazon			Amazon → YelpNYC		
K	50	150	250	50	150	250	50	150	250	50	150	250
LOF	0.460	0.260	0.176	0.440	0.213	0.172	0.140	0.073	0.052	0.380	0.200	0.168
ConOut	0.260	0.107	0.064	0.480	0.280	0.216	0.040	0.020	0.012	0.660	0.407	0.328
AMEN	0.040	0.073	0.092	0.160	0.113	0.080	0.020	0.013	0.012	0.580	0.333	0.264
DOMINANT	0.580	0.327	0.236	0.560	0.320	0.224	0.480	0.433	0.444	0.620	0.407	0.320
ADDA	0.460	0.233	0.176	0.500	0.247	0.172	0.380	0.220	0.184	0.540	0.353	0.312
COMMANDER	**0.620**	**0.360**	**0.244**	**0.600**	**0.347**	**0.228**	**0.500**	**0.460**	**0.456**	**0.680**	**0.420**	**0.332**

TABLE III

RESULTS OF CROSS-DOMAIN GRAPH ANOMALY DETECTION W.R.T. RECALL@K

Recall@K												
	YelpHotel → YelpRes			YelpRes → YelpHotel			YelpNYC → Amazon			Amazon → YelpNYC		
K	50	150	250	50	150	250	50	150	250	50	150	250
LOF	0.084	0.142	0.160	0.088	0.128	0.172	0.009	0.015	0.017	0.019	0.030	0.042
ConOut	0.047	0.058	0.058	0.096	0.168	0.216	0.003	0.004	0.004	0.033	0.061	0.082
AMEN	0.007	0.040	0.084	0.032	0.068	0.080	0.001	0.003	0.004	0.029	0.050	0.066
DOMINANT	0.105	0.178	0.215	0.112	0.192	0.224	0.032	0.087	0.148	0.031	0.061	0.080
ADDA	0.084	0.127	0.160	0.100	0.148	0.172	0.025	0.044	0.061	0.027	0.053	0.078
COMMANDER	**0.113**	**0.196**	**0.222**	**0.120**	**0.208**	**0.228**	**0.033**	**0.092**	**0.152**	**0.034**	**0.063**	**0.083**

1) *AUC:* As a widely used evaluation metric in anomaly detection methods [13], [28], [40], AUC value is the area under the ROC curve, representing the probability that a randomly chosen abnormal node is ranked higher than a normal node. If AUC approaches 1, the method is of high quality for detecting anomalies.
2) *Precision@K:* As each anomaly detection method outputs a ranking list according to the anomalous scores of different nodes, we use Precision@K to measure the proportion of true anomalies that a specific detection method discovered in its top K ranked nodes.
3) *Recall@K:* This metric measures the proportion of true anomalies that a specific detection method discovered in the total number of ground truth anomalies.

B. Evaluation Results

First, we evaluate the performance of the proposed framework COMMANDER and other unsupervised baseline methods on four different domain shifts. The results with respect to AUC scores are presented in Fig. 4. We also report the Precision@K scores and Recall@K scores in Tables II and III, respectively. From a comprehensive view, we can clearly find that our approach COMMANDER achieves considerable improvements over the state-of-the-art unsupervised methods on all the domain shifts. Take AUC value as an example, the performance of COMMANDER is 2.6% higher than the best baseline on the YelpHotel → YelpRes case, and the corresponding improvements on YelpHotel → YelpRes, YelpNYC → Amazon, and Amazon → YelpNYC are reported with 5.4%, 1.7%, and 1.6%, respectively. Meanwhile, our approach consistently outperforms the best performing baselines according to Precision@K and Recall@K results, which indicates that COMMANDER is capable of discovering more anomalous nodes in its top return lists and once again demonstrates the effectiveness of our approach.

Note that the unsupervised methods, including LOF, ConOut, and AMEN, cannot achieve competitive results in comparison. In particular, the performance of LOF is limited by its inability of modeling node dependencies. We also observe that AMEN performs poorly in the task of ranking anomalous nodes. One explanation is that AMEN is designed for detecting anomalous neighborhoods rather than nodes. Even though DOMINANT performs best among all the unsupervised methods due to the excellent expressive power of GCN, it is still largely behind our approach as it is unable to accurately spot those *shared anomalies* by utilizing labeled data from the source graph.

Next, we compare the performance of the domain adaptation method ADDA with our proposed framework COMMANDER. With the reported results (w.r.t. AUC scores), we observe that

COMMANDER outperforms ADDA by a significant margin, reaching around 10%–20% relative improvement in most cases. Meanwhile, as shown in Tables II and III, COMMANDER is able to discover more true anomalies on its top anomaly ranking list than ADDA. There are two major reasons that result in the ineffectiveness of ADDA for the studied problem. First, node dependency information is indispensable for assessing the abnormality of a node, while ADDA cannot model such information modality. Second, ADDA is unable to detect the *unshared anomalies* on the target graph since it is not tailored for anomaly detection problems. On the contrary, our approach COMMANDER is able to detect *unshared anomalies* on the target graph using the Attribute Decoder *Dec*.

In addition, the results show that our approach is able to achieve larger improvements in the first two domain shifts than the last two. Compared with the attributed graphs YelpHotel and YelpRes, the attributed graphs YelpNYC and Amazon are not only from two different business domains but also from two different platforms. Thus, this observation implies that the model performance is strongly associated with the degree of domain discrepancy. In brief, smaller domain discrepancy could be easier adapted, leading to better cross-domain anomaly detection performance.

C. Ablation Study

To investigate how much is the contribution of each component, in this section, we design the ablation study and show the corresponding experimental results. Specifically, we compare our proposed framework COMMANDER with the following three variants.

1) *Clf:* We exclude the domain discriminator and attribute decoder from COMMANDER and only use the anomaly classifier to detect anomalies on the target domain attributed graph G^t.
2) *Clf + Dis:* We exclude the attribute decoder from the proposed framework COMMANDER and use the anomaly classifier and domain discriminator to detect anomalies on the target domain attributed graph G^t.
3) *Dec:* We exclude the anomaly classifier and domain discriminator from the proposed framework COMMANDER and only employ attribute decoder for detecting anomalies on the target domain attributed graph G^t.
4) *w/o GAT:* We replace the GAT layers in COMMANDER with GCN layers to examine the effectiveness of using GAT for anomaly detection.

The comparison results on YelpHotel → YelpRes and YelpRes → YelpHotel are shown in Table IV, and the results on YelpNYC → Amazon and Amazon → YelpNYC are shown in Table V. Due to the space limit, we only show the results in terms of Precison@50 and AUC in our ablation study. From the reported results, we make the following observations.

1) By examining the performance of *Clf* on four domain shifts, we can clearly find that it performs poorly overall. On the contrary, the variant *Clf + Dis* improves the detection performance to a large extent with the join of *Dis*, which demonstrates that an anomaly classifier trained on the G^s cannot be directly used on G^t without domain adaptation.
2) Comparing to the variant *Clf + Dis*, *Dec* achieves superior detection performance in our experiments. The reasonable explanation is that the attribute decoder provides a more comprehensive assessment and is capable of detecting both *shared anomalies* and *unshared anomalies* to some extent.
3) By replacing the GAT layers in the COMMANDER framework with vanilla GCN layers, the performance decreases a noticeable margin, which shows the advantage of using a graph attention mechanism for detecting anomalies.
4) Since *Clf + Dis* and *Dec* considerably improve the detection performance, they still cannot achieve competitive results with our approach COMMANDER in the evaluations. It validates our assumption that *Dis* assists the anomaly classifier *Clf* to detect the *shared anomalies*; meanwhile, *Dec* is the key component to detect those *unshared anomalies* on the target graph.

TABLE IV
ABLATION RESULTS ON TWO CROSS-DOMAIN SETTINGS: YELPHOTEL → YELPRES AND YELPRES → YELPHOTEL

Methods	YelpHotel → YelpRes		YelpRes → YelpHotel	
	Pre@50	AUC	Pre@50	AUC
Clf	0.280	0.461	0.220	0.431
Clf+Dis	0.500	0.758	0.420	0.688
Dec	0.540	0.765	0.540	0.695
w/o GAT	0.580	0.776	0.580	0.722
COMMANDER	0.620	0.793	0.600	0.748

TABLE V
ABLATION RESULTS ON TWO CROSS-DOMAIN SETTINGS: YELPNYC → AMAZON AND AMAZON → YELPNYC

Methods	YelpNYC → Amazon		Amazon → YelpNYC	
	Pre@50	AUC	Pre@50	AUC
Clf	0.040	0.558	0.320	0.445
Clf+Dis	0.420	0.812	0.560	0.677
Dec	0.480	0.848	0.600	0.696
w/o GAT	0.460	0.857	0.640	0.702
COMMANDER	0.500	0.873	0.680	0.715

To summarize, the ablation study illustrates that the absence of any component will inevitably jeopardize the anomaly detection performance of COMMANDER on G^t. With all the principled components, the proposed framework largely outperforms all the variants under four domain shifts.

VI. RELATED WORK

A. Graph-Based Anomaly Detection

Graph-based anomaly detection methods have a specific focus on graph-structured data. Previous research mostly studies the problem of anomaly detection on plain graphs [3]. As graph structure is the only available information modality in a plain graph, this category of anomaly detection methods tries to exploit the graph structure information to spot anomalies from different perspectives [41], [42]. For instance, SCAN [41] is one of the first methods that target to find structural anomalies in graphs. In recent days, attributed

graphs have been widely used to model a wide range of complex systems due to their superior capacity for handling data heterogeneity. In addition to the observed node-to-node interactions, attributed graphs also encode a rich set of features for each node. Therefore, anomaly detection on attributed graphs has drawn increasing research attention in the community, and various methods have been proposed [36], [43], [44]. Among them, ConOut [36] identifies the local context for each node and performs anomaly ranking within the local context. AMEN [37] aims to discover anomalous neighborhoods on attributed graphs by considering the ego-network information for each node. More recently, researchers also propose to solve the problem of anomaly detection on attributed graphs using GNNs due to its strong modeling power [13], [45]–[47]. For instance, DOMINANT [13] achieves superior performance over other shallow methods by building a deep autoencoder architecture on top of the GCNs. Zhao *et al.* [47] proposed a novel loss function to train GNNs for anomaly detectable node representations. However, the aforementioned methods merely focus on a single graph and are unable to transfer the knowledge of anomalies from an auxiliary related domain.

B. Deep Domain Adaptation

Domain adaptation [48] aims at mitigating the generalization bottleneck introduced from domain shift. With the rapid growth of DNNs, deep domain adaptation has drawn much attention lately. In general, deep domain adaptation methods are trying to locate a domain-invariant feature space that can reduce the differences between the source and target domains. This goal is accomplished either by transforming the features from one domain to be closer to the other domain or projecting both domains into a domain-invariant latent space [9], [22], [49]. For instance, Tzeng *et al.* [50] leveraged an adaptation layer and a domain confusion loss to learn the domain-invariant representations. TLDA [51] is a deep autoencoder-based model, which tries to learn to domain-invariant representations and useful for label classification. Inspired by the idea of generative adversarial network (GAN) [25], researchers also propose to perform domain adaptation in an adversarial training paradigm [9], [14], [22], [23]. By exploiting a domain discriminator to distinguish the domain labels while learning deep features to confuse the discriminator, DANN [23] achieves superior domain adaptation performance. ADDA [14] learns a discriminative representation using labeled source domain data and then maps the target data to the same space through an adversarial loss. Later on, researchers also try to apply domain adaptation techniques on graph-structured data [24], [52]–[54] to handle the domain discrepancy between source and target graphs. For example, DANE [52] applies a shared weight GCN architecture with constraints of adversarial learning regularization, enabling cross-network knowledge transfer for unsupervised network embedding. Similarly, UDA-GCN [24] further proposes dual GCNs to capture both the local and global consistency relationships of each graph and uses an intergraphed-based attention mechanism to better represent each node. However, cross-domain anomaly detection remains unsolved in the graph learning community.

VII. Conclusion

In this article, we propose a novel anomaly detection framework called COMMANDER to tackle the problem of graph anomaly detection under the cross-domain setting. The proposed framework consists of four principled components: graph attentive encoder, anomaly classifier, domain discriminator, and attribute decoder. These components are tightly coupled to bridge the domain discrepancy between two attributed graphs from different domains and then perform accurate anomaly detection on the target attributed graph. We perform extensive experiments to corroborate the effectiveness of the proposed COMMANDER framework.

References

[1] G.-J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. S. Huang, "Exploring context and content links in social media: A latent space method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 850–862, May 2012.

[2] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *Proc. KDD*, 2008, pp. 990–998.

[3] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining Knowl. Discovery*, vol. 29, pp. 626–688, Jul. 2015.

[4] X. Hu, J. Tang, Y. Zhang, and H. Liu, "Social spammer detection in microblogging," in *Proc. IJCAI*, 2013.

[5] J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," *Comput. Secur.*, vol. 57, pp. 47–66, Mar. 2016.

[6] C. Qu *et al.*, "Learning to selectively transfer: Reinforced transfer learning for deep text matching," in *Proc. WSDM*, 2019, pp. 699–707.

[7] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," 2020, *arXiv:2007.02500*. [Online]. Available: http://arxiv.org/abs/2007.02500

[8] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, 2010, pp. 213–226.

[9] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," 2014, *arXiv:1409.7495*. [Online]. Available: http://arxiv.org/abs/1409.7495

[10] J. Hoffman *et al.*, "LSDA: Large scale detection through adaptation," in *Proc. NeurIPS*, 2014, pp. 3536–3544.

[11] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

[12] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. ICML*, 2011.

[13] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proc. SDM*, 2019, pp. 594–602.

[14] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017.

[17] K. Ding, J. Wang, J. Li, K. Shu, C. Liu, and H. Liu, "Graph prototypical networks for few-shot learning on attributed networks," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 295–304.

[18] K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: Hypergraph attention networks for inductive text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020.

[19] K. Ding, Q. Zhou, H. Tong, and H. Liu, "Few-shot network anomaly detection via cross-network meta-learning," in *Proc. Web Conf.*, Apr. 2021, pp. 2448–2456.

[20] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NeurIPS*, 2017, pp. 1–11.

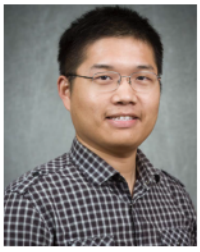
[21] Z. Li, Y. Wei, Y. Zhang, and Q. Yang, "Hierarchical attention transfer network for cross-domain sentiment classification," in *Proc. AAAI*, 2018.

[22] Y. Shu, Z. Cao, M. Long, and J. Wang, "Transferable curriculum for weakly-supervised domain adaptation," in *Proc. AAAI*, 2019, pp. 4951–4958.

[23] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.
[24] M. Wu, S. Pan, C. Zhou, X. Chang, and X. Zhu, "Unsupervised domain adaptive graph convolutional networks," in *Proc. Web Conf.*, Apr. 2020, pp. 1457–1467.
[25] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 1–9.
[26] S. Pan, R. Hu, S.-F. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2475–2487, Jun. 2020.
[27] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1511–1519.
[28] J. Li, H. Dani, X. Hu, and H. Liu, "Radar: Residual analysis for anomaly detection in attributed networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2152–2158.
[29] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 665–674.
[30] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Proc. ECML-PKDD*, 2017m, pp. 36–51.
[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[32] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proc. KDD*, 2015, pp. 985–994.
[33] P. Kaghazgaran, J. Caverlee, and A. Squicciarini, "Combating crowdsourced review manipulators: A neighborhood-based approach," in *Proc. WSDM*, 2018, pp. 306–314.
[34] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, pp. 43–52, Aug. 2010.
[35] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.
[36] P. I. Sánchez, E. Müller, O. Irmler, and K. Böhm, "Local context selection for outlier ranking in graphs with multiple numeric node attributes," in *Proc. 26th Int. Conf. Sci. Stat. Database Manage. (SSDBM)*, 2014, pp. 1–12.
[37] B. Perozzi and L. Akoglu, "Scalable anomaly ranking of attributed neighborhoods," in *Proc. SIAM Int. Conf. Data Mining*, Jun. 2016, pp. 207–215.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.
[40] Z. Peng, M. Luo, J. Li, H. Liu, and Q. Zheng, "ANOMALOUS: A joint modeling approach for anomaly detection on attributed networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.
[41] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "SCAN: A structural clustering algorithm for networks," in *Proc. KDD*, 2007, pp. 824–833.
[42] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Proc. PAKDD*, 2010, pp. 410–421.
[43] E. Muller, P. I. Sanchez, Y. Mülle, and K. Bohm, "Ranking outlier nodes in subspaces of attributed graphs," in *Proc. IEEE 29th Int. Conf. Data Eng. Workshops (ICDEW)*, Apr. 2013, pp. 216–222.
[44] K. Ding, J. Li, and H. Liu, "Interactive anomaly detection on attributed networks," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 357–365.
[45] K. Ding, J. Li, N. Agarwal, and H. Liu, "Inductive anomaly detection on attributed networks," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1288–1294.
[46] S. Bandyopadhyay, S. V. Vivek, and M. N. Murty, "Outlier resistant unsupervised deep architectures for attributed network embedding," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 25–33.
[47] T. Zhao, C. Deng, K. Yu, T. Jiang, D. Wang, and M. Jiang, "Error-bounded graph anomaly loss for GNNs," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1873–1882.
[48] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[49] J. Yu *et al.*, "Modelling domain relationships for transfer learning on retrieval-based question answering systems in E-commerce," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 682–690.
[50] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: http://arxiv.org/abs/1412.3474
[51] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. IJCAI*, 2015.
[52] Y. Zhang, G. Song, L. Du, S. Yang, and Y. Jin, "DANE: Domain adaptive network embedding," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019.
[53] Q. Dai, X. Shen, X.-M. Wu, and D. Wang, "Network transfer learning via adversarial domain adaptation with graph convolution," 2019, *arXiv:1909.01541*. [Online]. Available: http://arxiv.org/abs/1909.01541
[54] X. Shen, Q. Dai, F.-L. Chung, W. Lu, and K.-S. Choi, "Adversarial deep network embedding for cross-network node classification," in *Proc. AAAI*, 2020, pp. 2991–2999.

Kaize Ding received the bachelor's and master's degrees in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with Arizona State University, Tempe, AZ, USA.

His research interests are broadly in data mining and machine learning, with a particular focus on graph neural networks, few-shot learning, and self-supervised learning. He has published over 20 articles at top conferences and journals.



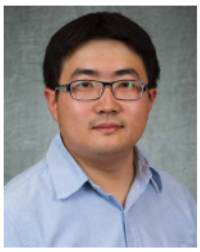
Kai Shu received the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in July 2020.

He is a Gladwin Development Chair Assistant Professor with the Department of Computer Science, Illinois Institute of Technology (IIT), Chicago, IL, USA. His research lies in machine learning, data mining, social computing, and their applications in disinformation, education, and healthcare.

Dr. Shu was awarded the ASU Fulton Schools of Engineering Dean's Doctoral Dissertation Award and the CIDSE Doctoral Fellowship 2015 and 2020.

Xuan Shan received the bachelor's and master's degrees in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2013 and 2016, respectively.

He is a Senior Software Engineer at Kuaishou Technology Company Ltd., Beijing. He was a Software Engineer at Microsoft (Asia) Software Technology Center (STCA). His research interest lies in search and information retrieval.



Jundong Li (Member, IEEE) is an Assistant Professor with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA, USA. His research interests are in data mining and machine learning, with a particular focus on graph mining and causality learning. His works on feature selection and graph representation learning are among the most cited articles in ACM CSUR, WSDM, SDM, and CIKM within the past five years according to Google Scholar Metrics. He was selected for the AAAI 2021 New Faculty Highlights Program.

Huan Liu (Fellow, IEEE) is a Professor of computer science engineering at Arizona State University, Tempe, AZ, USA. Before joining Arizona State University, he worked at Telecom Australia Research Labs and was on the faculty with the National University of Singapore, Singapore. His research focuses on developing computational methods for data mining, machine learning, and social computing, and designing efficient algorithms to enable effective problem solving ranging from basic research, text mining, to real-world applications.

Dr. Liu is a fellow of AAAI, AAAS, and ACM.