Cross-Domain Graph Anomaly Detection via Anomaly-Aware Contrastive Alignment

Qizhou Wang[1], **Guansong Pang**[*2], **Mahsa Salehi**[1], **Wray Buntine**[3,1], **Christopher Leckie**[4]
[1] Monash University
[2] Singapore Management University
[3] VinUniversity
[4] The University of Melbourne
{qizhou.wang, mahsa.salehi, wray.buntine}@monash.edu, gspang@smu.edu.sg, caleckie@unimelb.edu.au

Abstract

Cross-domain graph anomaly detection (CD-GAD) describes the problem of detecting anomalous nodes in an unlabelled target graph using auxiliary, related source graphs with labelled anomalous and normal nodes. Although it presents a promising approach to address the notoriously high false positive issue in anomaly detection, little work has been done in this line of research. There are numerous domain adaptation methods in the literature, but it is difficult to adapt them for GAD due to the unknown distributions of the anomalies and the complex node relations embedded in graph data. To this end, we introduce a novel domain adaptation approach, namely Anomaly-aware Contrastive alignmenT (ACT), for GAD. ACT is designed to jointly optimise: (i) *unsupervised contrastive learning* of normal representations of nodes in the target graph, and (ii) *anomaly-aware one-class alignment* that aligns these contrastive node representations and the representations of labelled normal nodes in the source graph, while enforcing significant deviation of the representations of the normal nodes from the labelled anomalous nodes in the source graph. In doing so, ACT effectively transfers anomaly-informed knowledge from the source graph to learn the complex node relations of the normal class for GAD on the target graph without any specification of the anomaly distributions. Extensive experiments on eight CD-GAD settings demonstrate that our approach ACT achieves substantially improved detection performance over 10 state-of-the-art GAD methods. Code is available at https://github.com/QZ-WANG/ACT.

Introduction

Detection of nodes that deviate significantly from the majority of nodes in a graph is a key task in graph anomaly detection (GAD). It has drawn wide research attention due to its numerous applications in a range of domains such as intrusion detection in cybersecurity, fraud detection in fintech and malicious user account detection in social network analysis. There are many shallow and deep methods (Akoglu, Tong, and Koutra 2015; Pang et al. 2021) that are specifically designed, or can be adapted for GAD. However, they are fully unsupervised approaches and often have notoriously high false positives due to the lack of knowledge about the anomalies of interest.

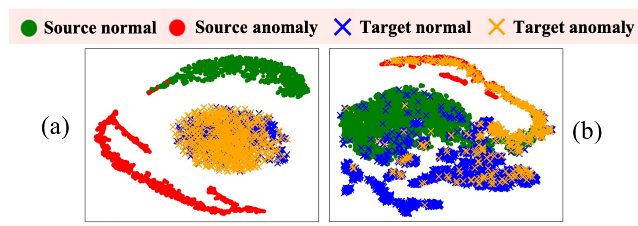

Figure 1: t-SNE visualisation of a CD-GAD dataset before (a) and after (b) our anomaly-aware contrastive alignment. Compared to (a) where the two domains show clear discrepancies in different aspects like anomaly distribution, in (b) our domain alignment approach effectively aligns the normal class, while pushing away the anomalous nodes in both source and target domains from the normal class.

We instead explore cross-domain (CD) anomaly detection approaches to address this long-standing issue. CD-GAD describes the problem of detecting anomalous nodes in an unlabelled target graph using auxiliary, related source graphs with labelled anomalous and normal nodes. The ground truth information in the source graph can provide important knowledge of true anomalies for GAD on the target graph when such supervision information from the source domain can be properly adapted to the target domain. The detection models can then be trained in an anomaly-informed fashion on the target graph, resulting in GAD models with substantially improved anomaly-discriminative capability, and thus greatly reducing the detection errors. Although such CD approaches can be a promising solution, little work has been done in this line of research.

There are numerous unsupervised domain adaptation (UDA) methods in the literature (Wilson and Cook 2020), but it is difficult to adapt them for GAD due to some unique challenges in GAD. The first challenge is that the distribution of different anomalies can vary within a dataset and across different datasets, and thus, the anomaly distribution often remains unknown in a target dataset. This challenges the popular assumption in UDA that the source and target domains have similar conditional probability distributions. Secondly, graph data contains complex node relations due to its topological structure and node attribute se-

*Corresponding author: Guansong Pang (gspang@smu.edu.sg).

mantics, leading to substantial discrepancies in graph structures (e.g., node degree distribution and graph density) and attribute spaces (e.g., feature dimensionality size) across different graph datasets (see Figure 1(a) for an example). These significant domain gaps in the raw input render many UDA methods ineffective, since they require more homogeneous raw input to effectively adapt the domain knowledge (e.g., pre-trained feature representation models can be directly applied to both source and target domains to extract relevant initial latent representations).

To address these two challenges, we introduce a novel domain adaptation approach, namely Anomaly-aware Contrastive alignmenT (ACT), for GAD. ACT is designed to jointly optimise: (i) *unsupervised contrastive learning* of normal representations of nodes in the target graph, and (ii) *anomaly-aware one-class alignment* that aligns these contrastive node representations and the representations of labelled normal nodes in the source graph data, while enforcing significant deviation of the representations of the normal nodes from the labelled anomalous nodes in the source graph. In doing so, ACT effectively transfers anomaly-informed knowledge from the source graph to enable the learning of the complex node relations of the normal class for GAD on the target graph without any specification of the anomaly distributions, as illustrated in Figure 1(b). We also show that after our domain alignment, self-labelling-based deviation learning can be leveraged on the domain-adapted representations of the target graph to refine the detection models for better detection performance.

In summary, our main contributions are as follows:

- We propose a novel approach, named anomaly-aware contrastive alignment (ACT), for CD-GAD. It synthesises anomaly-aware one-class alignment and unsupervised contrastive graph learning to learn anomaly-informed detection models on target graph data, substantially reducing the notoriously high false positives due to the lack of knowledge about true anomalies.
- We propose the use of self-labelling-based deviation learning on the target graph after the domain alignment to further refine our detection model, resulting in significantly enhanced detection performance.
- Large-scale empirical evaluation of ACT and 10 state-of-the-art (SOTA) competing methods is performed on eight real-world CD-GAD datasets to justify the superiority of ACT. These results also establish important performance benchmarks in this under-explored area.

Related Work

Graph Anomaly Detection

GAD methods typically adopt unsupervised learning due to the scarcity of labelled anomalies (Ma et al. 2021). Earlier non-deep-learning-based methods employ various measures (Gao et al. 2010; Perozzi and Akoglu 2016; Peng et al. 2018; Li et al. 2017) to identify anomalies. Recent GAD methods predominantly use Graph Neural Networks (GNNs) due to their strong learning capacity and are shown to be more effective. Ding et al. (2019) and Chen et al. (2020) employed graph auto-encoders to define anomaly scores using reconstruction error. Contrastive learning (Liu et al. 2021), adversarial learning (Chen et al. 2020), and other representation learning approaches (Zhao et al. 2020; Bandyopadhyay, Vivek, and Murty 2020; Wang et al. 2022) have been explored for GAD. However, they are unsupervised methods and focused on single-domain GAD. Limited work has been done on CD-GAD. Two most related studies are (Ding et al. 2021, 2022). Ding et al. (2021) adapts a meta-learning approach to address the problem, while Ding et al. (2022) combine a graph autoencoder and adversarial learning for CD-GAD. However, they suffer from limitations such as parameter sharing of cross-domain feature learners and unstable performance in the domain alignment.

Unsupervised Domain Adaptation

UDA aims to leverage labelled source data to improve similar tasks in an unlabelled domain. A popular approach is to reduce domain discrepancies, measured by some predefined metrics such as MMD (Gretton et al. 2006; Long et al. 2016) and Wasserstein Distance (Shen et al. 2018; Lee et al. 2019). Adversarial learning is also widely used by UDA methods (Ganin and Lempitsky 2015; Tzeng et al. 2015, 2017; Bousmalis et al. 2017; Hoffman et al. 2018; Saito et al. 2018a; Xiao and Zhang 2021), which learns domain-invariant representations in a competing training scheme. Some recent methods focus on class-wise alignment (Xie et al. 2018; Saito et al. 2018b). These approaches have been recently adapted to graph data, e.g., by adversarial graph learning (Zhang et al. 2019; Wu et al. 2020; Wu, Pan, and Zhu 2022) or graph proximity preserved representation learning (Shen et al. 2020). Nevertheless, these methods are primarily designed for CD settings with class-balanced data and relatively small domain discrepancy, rendering them inapplicable for GAD.

ACT: The Proposed Approach

Problem Statement

We consider unsupervised CD-GAD on attributed graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph with n nodes, where $\mathcal{V}, \mathcal{E}$, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ are its node set, edge set and feature matrix, respectively. In the unsupervised CD setting, in addition to a target graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t \in \mathbb{R}^{n_t \times d_t})$ with n_s nodes without any class labels, a labelled source graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{X}_s)$ is also available, which contains n_s nodes with their features $\mathbf{X}_s \in \mathbb{R}^{n_s \times d_s}$ and their normal/anomaly labels $Y_s \in \mathbb{R}^{n_s \times 1}$. Our task is to leverage both $\mathcal{G}_s$ and $\mathcal{G}_t$ to develop an anomaly scoring function ϕ_t, such that:

$$\phi_t(\mathcal{G}_t, v_j) \gg \phi_t(\mathcal{G}_t, v_i) \quad \forall (v_j \in \mathcal{V}_t^{\text{out}}) \wedge (v_i \in \mathcal{V}_t^{\text{in}}), \quad (1)$$

where $\mathcal{V}_t^{\text{in}}$ and $\mathcal{V}_t^{\text{out}}$ are respective normal and anomalous node sets, satisfying $\mathcal{V}_t^{\text{in}} \cup \mathcal{V}_t^{\text{out}} = \mathcal{V}_t$ and $\mathcal{V}_t^{\text{in}} \cap \mathcal{V}_t^{\text{out}} = \emptyset$.

We focus on neural-network-based anomaly scoring functions $\phi(\cdot; \Theta) : \mathcal{X} \to \mathbb{R}$, which can be seen as a combination of a feature representation learner $\psi(\cdot; \Theta_f) : \mathcal{X} \to \mathcal{Z}$ and a anomaly scoring function $\eta(\cdot; \Theta_g) : \mathcal{Z} \to \mathbb{R}$, where $\mathcal{X}$ is the input space, $\mathcal{Z} \in \mathbb{R}^M$ is the intermediate node representation space and $\Theta = \{\Theta_f, \Theta_g\}$ are the learnable parameters

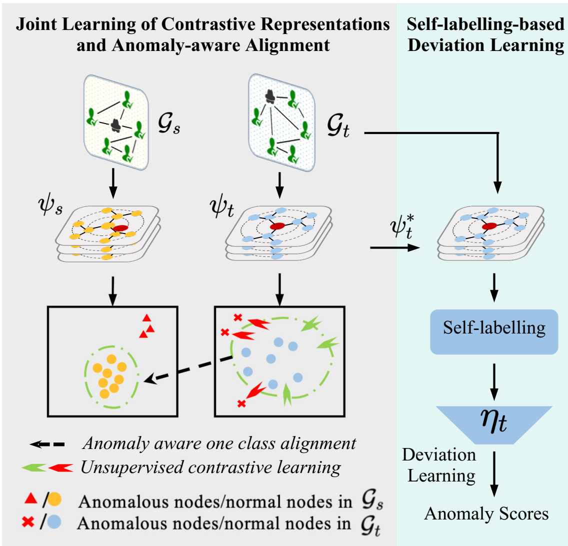

Figure 2: Overview of our proposed approach ACT.

of ϕ. Then we aim to learn the following anomaly scoring mapping:

$$\phi_t(\mathcal{G}_t, v; \Theta) = \eta_t(\psi_t(\mathcal{G}_t, v; \Theta_f^t); \Theta_g^t), \tag{2}$$

with the support from ψ_s and η_s trained on the labelled source graph data. Two main challenges here include the unknown anomaly distribution in the target data, and the complex discrepancies in graph structures and semantic attribute spaces among different graphs.

Overview of ACT

To address the above two challenges, we propose the approach Anomaly-aware Contrastive alignmenT (ACT). The key idea is to adapt the anomaly-discriminative knowledge from a labelled source graph to learn anomaly-informed detection models on the target graph, reducing the high detection error rates in unsupervised detection models that are lacking knowledge about the anomalies of interest.

As illustrated in Figure 2, ACT learns such anomaly-informed models on the unlabelled target graph using two major components. It first performs a joint optimisation of anomaly-aware one-class domain alignment and unsupervised contrastive node representation learning on the target graph, resulting in an expressive node representation mapping ψ_t^* that is domain-adapted for GAD on the target graph.

In the second phase, ACT performs self-labelling-based deviation learning, in which an off-the-shelf anomaly detector is used on top of the domain-adapted representation space ψ_t^* to identify pseudo anomalies that are subsequently employed to learn an anomaly scoring neural network on the target graph via a deviation loss.

After domain alignment, the source-domain-based anomaly scoring network η_s can also be used to produce the pseudo anomalies for the subsequent deviation learning, but it is generally less effective than using the off-the-shelf anomaly detector on ψ_t (see Suppl. Material). Thus, the latter approach is used by default.

Joint Learning of Contrastive Representations and Anomaly-Aware Alignment

We aim to achieve anomaly-aware one-class domain alignment in the presence of a large domain gap in graph structure, node attribute semantics, and anomaly distributions. Many UDA methods exploit pretrained representation learners or parameter sharing to initialise target node representations so that they are reasonably aligned with their corresponding source classes. However, this does not apply to graph data due to high discrepancies across different graphs.

To address this challenge, we introduce a batch-sampling-based joint learning approach to perform an optimal-transport-based domain alignment $\mathcal{L}_{\text{dom}}$ with unsupervised contrastive graph learning $\mathcal{L}_{\text{con}}$ by optimising the following loss function:

$$\mathcal{L}_{\text{joint}}(\mathbf{Z}_s, \mathbf{Z}_t) = \mathcal{L}_{dom}(\mathbf{Z}_s, \mathbf{Z}_t) + \mathcal{L}_{\text{con}}(\mathbf{Z}_t), \tag{3}$$

where $\mathbf{Z}_s = \psi_s(\mathcal{G}_s, \mathbf{B}_s; \Theta_f^s)$ and $\mathbf{Z}_t = \psi_t(\mathcal{G}_t, \mathbf{B}_t; \Theta_f^t)$ are the respective node representations of a sampled source node batch $\mathbf{B}_s$ and a target node batch $\mathbf{B}_t$. Below we introduce each term of Eq. (3) in detail.

Unsupervised Contrastive Learning of Normal Representations of Nodes on the Target Graph Our unsupervised contrastive learning aims to (i) achieve initial representations of regular patterns embedded in the majority of nodes (i.e., normal representations of nodes) on the target graph and (ii) correct misalignment of node representations during the joint learning. To this end, we adopt a topology-based contrastive loss based on the common *graph homophily phenomenon* – similar nodes are more likely to attach to each other than dissimilar ones – to learn the representation of target nodes. The phenomenon of homophily is assumed to be widely applied to most nodes of a graph. Thus, we use this property to define normal nodes as the ones that are consistent with their neighbourhood, and the nodes that violate the assumption are considered to be abnormal otherwise. Accordingly, we use this property to devise the unsupervised contrastive learning loss as:

$$\begin{aligned} \mathcal{L}_{\text{con}}(\mathbf{Z}_t) = & -\log\left(\sigma({\mathbf{Z}_t^u}^\top \mathbf{Z}_t^v)\right) \\ & - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log\left(\sigma(-{\mathbf{Z}_t^u}^\top \mathbf{Z}_t^{v_n})\right), \end{aligned} \tag{4}$$

where $\mathbf{Z}_t = \psi_t(\mathcal{G}_t, \mathbf{B}_t; \Theta_f^t)$ are the representations of a target node batch $\mathbf{B}_t$, parameterised by Θ_f^t; $\mathbf{B}_t$ consists of the target nodes $\mathbf{B}_t^u$, their positive examples $\mathbf{B}_t^v$ that occur in the first-order neighbourhood of u, and their negative examples $\mathbf{B}_t^{v_n}$ sampled from non-neighbour node set P_n; $\mathbf{Z}_t^u$, $\mathbf{Z}_t^v$, and $\mathbf{Z}_t^{v_n}]$ are the representations of $\mathbf{B}_t^u$, $\mathbf{B}_t^v$ and $\mathbf{B}_t^{v_n}$, respectively; $\mathbf{Z_t} = [\mathbf{Z}_t^u, \mathbf{Z}_t^v, \mathbf{Z}_t^{v_n}]$ is the concatenation of the three representations. By minimising Eq. 4, the target node representation mapping ψ_t is enforced to learn the regularity representations of the nodes, which can also help correct possible misalignment of the target nodes when jointly optimising with the following domain alignment.

Anomaly-aware One-class Domain Alignment Since the target anomaly distribution can be substantially dissimilar to the source anomaly class, we propose to focus on aligning the normal class between the two domains, with the anomaly class information in the source graph to support this one-class alignment.

The choice of domain discrepancy description is crucial for the alignment. The probability-based measures and adversarial-learning-based approaches are two popular solutions (Wilson and Cook 2020). In our case, the former approach is more appropriate than the latter one as the adversarial learning can be easily affected by the two main challenges mentioned above. The Wasserstein metric has been shown to be more promising among all probability-based discrepancy measures because it considers the underlying geometry of the probability space. It can provide reasonable measures in extreme cases, such as distributions that do not share support (Lee et al. 2019) or provide stable gradients for points that lie in low probability regions (Shen et al. 2018). Thus, we use the Wasserstein distance to measure the domain discrepancy of the normal class in the feature representation space of the two domains in an unsupervised way, while at the same time having anomaly-aware normal class representation learning in the source domain. In particular, we define the one-class alignment loss as:

$$\mathcal{L}_{\mathrm{dom}}(\mathbf{Z}_s, \mathbf{Z}_t) = W_p(\mathbf{Z}_s, \mathbf{Z}_t), \tag{5}$$

where W_p is the Wasserstein distance and defined as:

$$W_p(\mathbf{P}_s, \mathbf{P}_t) = \inf_{\gamma \in \Pi} \left\{ \left(\Big(\mathop{\mathbb{E}}_{\mathbf{z}_s \sim \mathbf{P}_s, \mathbf{z}_t \sim \mathbf{P}_t} d(\mathbf{z}_s, \mathbf{z}_t)^p \Big)^{\frac{1}{p}} \right) \right\}, \tag{6}$$

where Ω_s and Ω_t are two domains on a metric space Ω, which are respectively related to two different probability distributions $\mathbf{P}_s$ and $\mathbf{P}_t$; $\gamma \in \pi$ is the set of all probabilistic coupling between Ω_s and Ω_t; and $d(\mathbf{z}_s, \mathbf{z}_t)^p$ specifies the cost of moving any $\mathbf{z}_s \in \Omega_s$ to $\mathbf{z}_t \in \Omega_t$. We use the Sinkhorn (Cuturi 2013) approximation of 2-Wasserstein distance for efficient estimation of the distance d.

Meanwhile, to leverage the anomaly information to learn normal class representations in the source domain without enforcing any assumptions on the anomaly distribution, we use a loss function, called deviation loss (Pang, Shen, and van den Hengel 2019). It enforces the clustering of normal nodes in the representation space w.r.t. a given prior, while making that of the anomalous nodes significantly deviate from the representations of normal nodes. Specifically, the loss adapted to our problem is given as follows:

$$\begin{aligned} L(\mathbf{z}_v, \mathbf{Z}_s, \mu, \sigma) =& (1-y)|\mathrm{dev}(\mathbf{z}_v, \mathbf{Z}_s, \mu, \sigma)| + \\ & y \times \max(0, a - \mathrm{dev}(\mathbf{z}_v, \mathbf{Z}_s, \mu, \sigma)), \end{aligned} \tag{7}$$

where $\mathbf{Z}_s = \psi_s(\mathcal{G}_s, \mathbf{B}_s; \Theta_f^s)$ are the feature representations of nodes in the source graph; $y = 1$ if v is an anomalous node and $y = 0$ otherwise; $\mathbf{z}_v \in \mathbf{Z}_s$; a is a confidence interval-based margin; and $\mathrm{dev}(\mathbf{z}_v, \mathbf{Z}_s, \mu, \sigma)$ is a Z-Score-based deviation function:

$$\mathrm{dev}(\mathbf{z}_v, \mathbf{Z}_s, \mu, \sigma) = \frac{\eta_s(\mathbf{z}_v, \mathbf{Z}_s; \Theta_g^s) - \mu}{\sigma}, \tag{8}$$

where μ and σ are two hyperparameters from a Gaussian prior $\mathcal{N}(\mu, \sigma^2)$. Following (Pang, Shen, and van den Hengel 2019), $\mu = 0$, $\sigma = 1$ and $a = 5$ are used in our implementation. Eq. (7) is minimised via the same mini-batch gradient descent approach as in the original paper. Note that the Gaussian prior in the deviation loss is made on the normal class rather than the anomaly class, so there is no specification of the anomaly distribution.

During training, a simultaneous optimisation of $\mathcal{L}_{\mathrm{con}}(\mathbf{Z}_t)$, $\mathcal{L}_{\mathrm{dom}}(\mathbf{Z}_s, \mathbf{Z}_t)$ and $L(\mathbf{z}_v, \mathbf{Z}_s, \mu, \sigma)$ can lead to unstable performance. In our implementation, we first learn $\mathbf{Z}_s$ by minimising $L(\mathbf{z}_v, \mathbf{Z}_s, \mu, \sigma))$, and then we fix $\mathbf{Z}_s$ and perform alternating optimisation of $\mathcal{L}_{\mathrm{con}}(\mathbf{Z}_t)$ and $\mathcal{L}_{\mathrm{dom}}(\mathbf{Z}_s, \mathbf{Z}_t)$.

Self-Labelling-Based Deviation Learning

After the one-class alignment above, we obtain the domain-adapted representation space ψ_t^* of the target graph and the source-domain-based anomaly scoring network η_s. Even though joint learning achieves good alignments, mismatches may still exist, which may be caused by the uncertain initial state in ψ_t and the large initial discrepancy between the source and target graph distributions. Thus, directly using η_s to perform anomaly detection on the target graph can also be unstable. To mitigate such effects, we propose the use of self-labelling-based deviation learning on the target graph. The self labelling is used to refine the learned prior knowledge of anomalies by focusing on nodes with high prediction confidence in each class to generalise the heuristics of their corresponding class distributions. Inspired by (Pang et al. 2018), we apply *Cantelli's* Inequality based thresholding method for self labelling, which is used to obtain a set of pseudo anomalies $\mathcal{O}_{\mathrm{out}}$ via:

$$\mathcal{O}_{\mathrm{out}} = \{v | \mathbf{s}_{\mathcal{G}_t}(v) > \mathrm{mean}_{\mathbf{s}_{\mathcal{G}_t}} + \alpha \times \mathrm{std}_{\mathbf{s}_{\mathcal{G}_t}}, \forall v \in \mathcal{G}_t\}, \tag{9}$$

where $\mathbf{s}_{\mathcal{G}_t} = \mathcal{M}(\mathcal{G}_t, \psi_t^*)$ is a score vector that contains the anomaly scores of all nodes yielded by an off-the-shelf anomaly detector $\mathcal{M}$ on the representation space ψ_t^*; $\mathbf{s}_{\mathcal{G}_t}(v)$ returns the anomaly score of the node $v \in \mathcal{G}_t$; $\mathrm{mean}_{\mathbf{s}_{\mathcal{G}_t}}$ and $\mathrm{std}_{\mathbf{s}_{\mathcal{G}_t}}$ are the mean and standard deviation of all the scores in $\mathbf{s}_{\mathcal{G}_t}$; $\alpha > 0$ is a user-defined hyperparameter.

In addition to pseudo anomaly detection, to perform deviation learning as in Eq. 7, we also need a set of pseudo normal nodes in the target graph. Unlike pseudo anomaly identification, the identification of pseudo normal nodes is trivial, since the majority of nodes are assumed to be normal. We simply select the bottom q percentile p_{1-q} ranked nodes w.r.t. $\mathbf{s}_{\mathcal{G}_t}$ as the pseudo normal nodes; and the final GAD performance is insensitive to q. After that, we use the pseudo labelled samples to re-learn the ψ_t by minimising the deviation loss in Eq. (7) with $\mathbf{Z}_s$ replaced with the pseudo anomalous and normal target nodes. In doing so, it can largely reduce the effect of potential misaligned node representations in the domain alignment, since the self labelling helps effectively reduce the false positives. This optimisation accordingly produces the target-domain-based anomaly scoring network η_t, which is used together with the newly learned ψ_t to perform anomaly detection on the target graph.

	Method	CD-GAD Dataset								
		RES → HTL	**NYC → HTL**	**HTL → RES**	**NYC → RES**	**RES → NYC**	**HTL → NYC**	**AMZ → NYC**	**NYC → AMZ**	**Average**
AUC-ROC	DGI+LOF	<u>0.778±0.009</u>		<u>0.850±0.026</u>		0.612±0.015			0.728±0.015	0.742±0.016
	DGI+IF	0.667±0.009		0.843±0.017		**0.844±0.007**			0.537±0.015	0.723±0.009
	ANOM	0.186±0.002		0.417±0.009		0.536±0.001			0.496±0.003	0.409±0.004
	DOM	0.694±0.000		0.767±0.000		0.692±0.000			<u>0.867±0.000</u>	0.755±0.000
	ADONE	0.738±0.035		0.477±0.024		0.623±0.036			0.847±0.052	0.671±0.037
	GAAN	0.644±0.010		0.668±0.041		0.406±0.006			0.861±0.015	0.645±0.018
	COLA	0.485±0.034		0.555±0.063		0.811±0.006			0.496±0.003	0.587±0.027
	ADDA	0.624±0.064	0.589±0.109	0.787±0.144	0.726±0.345	0.750±0.076	0.750±0.062	0.697±0.126	0.640±0.051	0.684±0.118
	CMDR (s)	0.690±0.009	N/A	0.774±0.007	N/A	N/A	N/A	0.699±0.006	0.859±0.007	<u>0.756±0.007</u>
	CMDR (u)	0.699±0.009	0.707±0.009	0.763±0.024	0.780±0.017	0.694±0.006	0.693±0.002	0.695±0.001	0.848±0.009	0.751±0.012
	ACT	**0.804±0.006**	**0.792±0.018**	**0.892±0.015**	**0.948±0.014**	<u>0.831±0.005</u>	<u>0.830±0.005</u>	<u>0.830±0.002</u>	**0.925±0.004**	**0.868±0.009**
AUC-PR	DGI+LOF	<u>0.247±0.016</u>		0.269±0.036		0.133±0.006			0.097±0.008	0.186±0.017
	DGI+IF	0.194±0.013		<u>0.296±0.031</u>		**0.366+0.014**			0.042±0.003	<u>0.226±0.009</u>
	ANOM	0.053±0.000		0.040±0.001		0.091±0.001			0.037±0.000	0.055±0.001
	DOM	0.216±0.000		0.264±0.000		0.145±0.000			0.252±0.000	0.219±0.000
	ADONE	0.244±0.029		0.183±0.031		0.155±0.029			<u>0.259±0.076</u>	0.210±0.041
	GAAN	0.152±0.006		0.089±0.017		0.039±0.001			0.203±0.035	0.121±0.015
	COLA	0.082±0.009		0.109±0.011		0.128±0.003			0.037±0.000	0.089±0.006
	ADDA	0.227±0.028	0.171±0.062	0.260±0.126	0.254±0.140	<u>0.254±0.140</u>	0.181±0.021	0.239±0.110	0.051±0.002	0.177±0.057
	CMDR (s)	0.210±0.007	N/A	0.268±0.006	N/A	N/A	N/A	0.145±0.001	0.242±0.019	0.216±0.008
	CMDR (u)	0.216±0.008	0.207±0.009	0.253±0.025	0.267±0.015	0.144±0.003	0.144±0.002	0.145±0.001	0.220±0.024	0.209±0.014
	ACT	**0.287±0.006**	**0.284±0.010**	**0.330±0.018**	**0.477±0.065**	0.249±0.012	<u>0.241±0.009</u>	<u>0.243±0.003</u>	**0.497±0.020**	**0.358±0.002**

Table 1: AUC-ROC and AUC-PR (±std) comparison. 'N/A' indicates that CMDR (s) cannot work on datasets with different numbers of node attributes in the two domains. The boldfaced and underlined are the best and second-best results, respectively.

Experiments

Datasets

Eight CD-GAD settings based on four real-world GAD datasets, including *YelpHotel* (HTL), *YelpRes* (RES), *YelpNYC* (NYC) and *Amazon* (AMZ)[1], are created as follows, with each setting having two related datasets as the source and target domains.

YelpHotel (HTL) ⇌ *YelpRes* (RES). These two datasets are Yelp online review graphs in the Chicago area for accommodation and dining businesses. A node represents a reviewer and an edge indicates two reviewers have reviewed the same business. Reviewers with filtered reviews by Yelp anti-fraud filters are regarded as anomalies. Each of the datasets can serve as either source or target domain. The primary domain shift here is the course of business.

YelpNYC (NYC)⇌ *Amazon* (AMZ). These are also review graphs. YelpNYC is collected from New York City for dining businesses, while Amazon is for E-commerce reviews. Anomalies are users with multiple reviews identified using crowd-sourcing efforts. The domain gap here is greater than HTL ⇌ RES as these two datasets are less co-related.

YelpRes (RES) ⇌ *YelpNYC* (NYC). The primary domain shift here is geographical location, as both graphs are for dining business reviews. This pair presents additional significant challenges due to their heterogeneous feature spaces and a large difference in graph size.

YelpHotel (HTL) ⇌ *YelpNYC* (NYC). It is similar to RES ⇌ NYC, however, with more substantial domain gaps in not only geographical locations but also their business types (dining venues vs. accommodation).

[1]Statistics of each dataset are given in Suppl. Material

Competing Methods and Evaluation Metrics

We consider 10 SOTA competing methods from two related lines of research: unsupervised GAD and CD methods. Two unsupervised GAD methods are based on the combination of LOF (Breunig et al. 2000) and iForest (IF) (Liu, Ting, and Zhou 2008) and node embedding via Deep Graph Infomax (DGI) (Veličković et al. 2018). Further, we also include five recent unsupervised GAD methods : ANOMALOUS (ANOM) (Peng et al. 2018), DOMINANT (DOM) (Ding et al. 2019), AdONE (Bandyopadhyay, Vivek, and Murty 2020), GGAN (Chen et al. 2020) and COLA (Liu et al. 2021). They are included to examine whether ACT can benefit from the source domain information for unsupervised GAD on the target domain. For CD methods, we choose COMMANDER (Ding et al. 2022) (CMDR for short) and ADDA – a popular general domain adaptation method (Wilson and Cook 2020). As the original CMDR, termed CMDR (s), adopts a shared representation learner for both domains, we derive a variant of CMDR, termed CMDR (u), that can work in two domains with different feature spaces by learning separate graph representation learners for each domain.

We employ two popular, complementary performance metrics for AD, the Area Under Receiver Operating Characteristic Curve (AUC-ROC) and the Area Under Precision-Recall Curve (AUC-PR), which are holistic metrics that quantify the performance of an AD model across a wide range of decision thresholds. Larger AUC-ROC (or AUC-PR) indicates better performance.

Implementation Details

Our model ACT is implemented with a three-layer GraphSAGE (Hamilton, Ying, and Leskovec 2017) within which

256 and 64 hidden dimensions are chosen for ψ_s and ψ_t respectively. The source model is trained for 50 epochs using a learning rate of 10^{-3}. The domain alignment is performed for 50 epochs using a learning rate of 10^{-4}. The same learning rate is also used in self-labelling-based deviation learning, wherein IF is used as the off-the-shelf detector $\mathcal{M}$. The optimisation is done in mini-batches of 128 target (centre) nodes using the ADAM optimiser (Kingma and Ba 2014) . We use the sample size of 25 and 10 for the two hidden layers during message passing. In self labelling, $\alpha = 2.5$ and $q = 25$ are used by default. These neural network settings and training methods are used throughout all the settings of our experiments. All the results are averaged over five independent runs using random seeds. The model settings and training of the competing methods are based on default/recommended choices of their authors.

Detection Performance on Real-World Datasets

We compare ACT with 10 SOTA competing methods on eight real-world CD settings, with the results shown in Table 1, where 'A → B' represents the use of a source dataset A for GAD on a target dataset B; and unsupervised anomaly detectors use only the target data.

Overall Performance ACT performs stably across all eight settings and substantially outperforms all competing methods by at least 11% and 13% in average AUC-ROC and AUC-PR, respectively. In particular, benefiting from the anomaly-aware alignment and self-labelling-based deviation learning, ACT demonstrates consistent superiority over the competing CD methods on all eight datasets. Unsupervised detectors work well only on very selective datasets where their definition of anomaly fits well with the underlying anomaly distribution, e.g., the method IF on NYC, and they become unstable and ineffective otherwise. By contrast, ACT learns anomaly-informed models with the relevant anomaly supervision from the source data, and thus, it can perform stably and work well across the datasets.

Semantic Domain Gap For CD-GAD, using different source graphs results in similar performance in most cases. However, in some cases, one source can be more informative than the others, e.g., the results of ACT on NYC → RES vs. HTL → RES, indicating a closer domain gap between NYC and RES than that between HTL and RES.

Heterogeneous Structure/Attribute Inputs ACT can effectively handle scenarios where the source and the target have a large difference in graph structure and/or node attribute dimension, such as NYC → HTL and NYC → RES. By contrast, ADDA and CMDR (u) fail to work effectively in such cases (CMDR (s) is inapplicable as it requires a shared feature learner on the two domains).

Effectiveness of Utilising Source Domain Data

This subsection provides an in-depth empirical investigation of the importance of source data to CD-GAD by answering two key questions below.

How much source domain data is required by ACT to outperform SOTA unsupervised detectors? To answer this question, we evaluate the performance of ACT on four representative datasets of different data complexities using five percentages of labelled source nodes: 0.5%, 5%, 25%, 50% and 100% (the rest of the nodes are treated as unlabelled data during training). The results are illustrated in Figure 3, with CMDR (s) and the best unsupervised result per dataset as baselines. It is impressive that even when a very small percentage (e.g., 0.5% or 5%) of labelled source data is used, ACT can perform better, or on par with, these strong baselines, demonstrating strong capability in unleashing the relevant information hidden in the source data. This capability is further verified by the increasing AUC-ROC and AUC-PR of ACT when the amount of source data used increases. Nevertheless, caution is required when the labelled source data is too small (e.g., 0.5% labelled source data corresponds to 21 nodes to 105 nodes for the four datasets), since ACT can perform unstably in such cases.

In addition to the domain alignment component, another major factor in the superior performance of ACT here is the self-labelling deviation learning (see our Ablation Study). Therefore, the second question below is investigated.

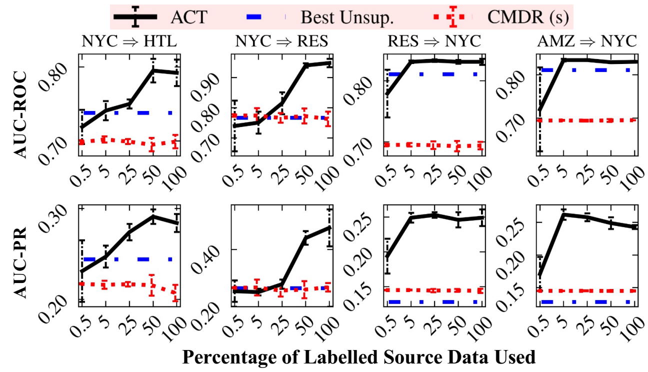

Figure 3: Efficiency of ACT utilising labelled source data, with best CD/unsupervised models as baselines.

		RES → HTL	HTL → RES	NYC → RES	NYC → AMZ
AUC-ROC	ANOM*	0.434±0.025	0.594±0.051		0.434±0.006
	DOM*	0.737±0.005	0.914±0.008		0.912±0.002
	ADONE*	0.674±0.059	0.825±0.057		0.775±0.089
	GGAN*	0.664±0.010	0.851±0.014		0.855±0.027
	COLA*	0.522±0.028	0.683±0.070		0.730±0.012
	Ours	**0.804±0.006**	0.892±0.015	**0.948±0.014**	**0.925±0.004**
AUC-PR	ANOM*	0.098±0.009	0.126±0.015		0.038±0.002
	DOM*	0.277±0.004	0.366±0.013		0.383±0.006
	ADONE*	0.243±0.034	0.288±0.026		0.195±0.140
	GGAN*	0.247±0.011	0.296±0.016		0.297±0.051
	COLA*	0.163±0.043	0.224±0.017		0.096±0.010
	Ours	**0.287±0.006**	0.330±0.018	**0.477±0.065**	**0.497±0.020**

Table 2: Self-labelling deviation learning on our ACT-based domain-adapted feature space vs. the original feature space. METHOD* means the use of METHOD to perform self labelling in the original feature space and then performs exactly the same deviation learning as in ACT using Eq. (7).

Can we just perform self-labelling deviation learning on the target domain directly, without using any source domain data? The answer is clearly negative. This can be observed by our empirical results in Table 2, where ACT

	AUC-ROC			AUC-PR		
	$\mathcal{L}_{\text{con}}$	$\mathcal{L}_{\text{dom}}$	$\mathcal{L}_{\text{joint}}$	$\mathcal{L}_{\text{con}}$	$\mathcal{L}_{\text{dom}}$	$\mathcal{L}_{\text{joint}}$
RES $\rightarrow$ HTL	0.485	**0.610**	0.608	0.099	0.216	**0.216**
NYC $\rightarrow$ HTL	0.534	0.609	**0.682**	0.125	0.211	**0.246**
HTL $\rightarrow$ RES	0.552	0.862	**0.880**	0.069	**0.308**	0.296
NYC $\rightarrow$ RES	0.596	0.662	**0.961**	0.316	0.160	**0.444**
HTL $\rightarrow$ NYC	0.615	0.671	**0.773**	**0.179**	0.145	0.171
RES $\rightarrow$ NYC	0.437	0.675	**0.753**	0.096	0.143	**0.163**
AMZ $\rightarrow$ NYC	0.578	0.578	**0.617**	0.101	0.134	**0.154**
NYC $\rightarrow$ AMZ	0.389	0.640	**0.880**	0.033	0.078	**0.587**
Average	0.523	0.663	**0.790**	0.127	0.179	**0.338**

Table 3: Anomaly-aware contrastive alignment vs. separate contrastive learning/anomaly-aware alignment.

is compared with five deviation-learning-enhanced unsupervised competing methods on four representative settings that cover adaptations between graphs of similar/different sizes and attributes. The results show that although self-labelling deviation learning helps achieve performance improvements on several datasets compared to the results of the original five unsupervised methods in Table 1, ACT still outperforms these five enhanced baselines by substantial margins in both AUC-ROC and AUC-PR. These results indicate that there is crucial anomaly knowledge adapted from the source data in the domain alignment stage in ACT; such knowledge cannot be obtained by working on only the target data.

Ablation Study

Joint Contrastive Graph Representation Learning and Anomaly-aware Alignment We first evaluate the importance of synthesising contrastive learning on the target graph and anomaly-aware domain alignment ($\mathcal{L}_{\text{joint}}$) in ACT, compared to the use of the individual contrastive learning ($\mathcal{L}_{\text{con}}$) or anomaly-aware alignment ($\mathcal{L}_{\text{dom}}$). The results are reported in Table 3, which shows that the joint learning enables significantly better adaptation of anomaly knowledge in the source domain to the target domain, substantially outperforming the use of $\mathcal{L}_{\text{con}}$ or $\mathcal{L}_{\text{dom}}$ across the eight settings. $\mathcal{L}_{\text{joint}}$ outperforms the two ACT variants by at least 12% and 14% in average AUC-ROC and AUC-PR respectively. The joint learning is advantageous because the contrastive learning models the regular patterns of the nodes in the target graph (i.e., learning the representations of normal nodes), while the anomaly-aware domain alignment allows the use of labelled anomaly and normal nodes in the source data to improve the normal representations in the target data. Optimising these two objectives independently fails to work effectively due to their strong reliance on each other.

Self-labelling-based Deviation Learning We then evaluate the importance of the self-labelling-based deviation learning component in ACT, with two variants of ACT, η_s and ACT-IF. η_s directly uses the source-domain-based anomaly detector η_s, while ACT-IF uses IF on the domain-adapted feature representation space of the target data to detect anomalies; both of which are done after the anomaly-aware contrastive alignment, but they do not involve the deviation learning.

Table 4 shows the comparison results, from which we can observe that the self-labelling-based deviation learning component in ACT largely outperforms η_s and ACT-IF, achieving average improvement by at least 8% in AUC-ROC and 2% in AUC-PR. The improvement can be attributed to the capability of the self labelling in identifying true anomalies in the target data with high confidence predictions, which enhances the representation learning of the normal and anomalous nodes in the subsequent deviation learning. The large improvement in AUC-ROC and relatively small improvement in AUC-PR indicate that ACT is more effective in reducing false positives than increasing true positives.

	AUC-ROC			AUC-PR		
	η_s	ACT-IF	ACT	η_s	ACT-IF	ACT
RES $\rightarrow$ HTL	0.608	0.740	**0.804**	0.216	0.261	**0.287**
NYC $\rightarrow$ HTL	0.682	0.744	**0.792**	0.246	0.257	**0.284**
HTL $\rightarrow$ RES	0.880	0.843	**0.892**	0.296	0.262	**0.330**
NYC $\rightarrow$ RES	**0.961**	0.955	0.948	0.444	0.427	**0.477**
HTL $\rightarrow$ NYC	0.773	0.781	**0.831**	0.171	0.166	**0.249**
RES $\rightarrow$ NYC	0.753	0.748	**0.830**	0.163	0.158	**0.241**
AMZ $\rightarrow$ NYC	0.617	0.792	**0.830**	0.154	0.191	**0.243**
NYC $\rightarrow$ AMZ	0.880	0.821	**0.925**	**0.587**	0.556	0.497
Average	0.790	0.785	**0.868**	0.338	0.333	**0.358**

Table 4: Self-labelling deviation learning in ACT vs. domain-adapted anomaly detector η_s and unsupervised detector IF on the adapted target feature space.

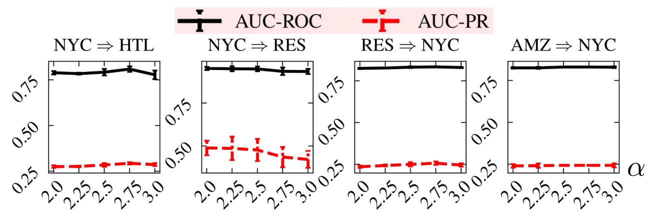

Figure 4: Sensitivity test results w.r.t. α.

Anomaly Thresholding Sensitivity

This section studies the sensitivity of ACT w.r.t. the anomaly thresholding hyperparameter α in Eq. (9), which determines the characteristics of the pseudo anomalies (e.g., the number and quality). The results with varying α settings are reported in Figure 4. In general, ACT maintains stable performance across the value range of α in $[2.0, 3.0]$, suggesting its good stability on datasets with different characteristics.

Conclusion

In this paper, we present Anomaly-aware Contrastive alignmenT (ACT) for CD-GAD, which connects an optimal-transport-based discrepancy measure and graph-structure-based contrastive loss to leverage prior AD knowledge from a source graph as a joint learning scheme. The resulting model achieves anomaly-aware one-class alignment under severe data imbalance and different source and target distributions. A self-labelling approach to deviation learning is further proposed to refine the learned source of AD knowledge. These two components result in significant GAD improvement on various real-world cross domains. In our future work, we plan to explore the use of multiple source graphs under the ACT framework.

Acknowledgements

This work was supported in part by the MASSIVE HPC facility (www.massive.org.au), The University of Melbourne's Research Computing Services and the Petascale Campus Initiative. Guansong Pang is supported in part by the Singapore Ministry of Education (MoE) Academic Research Fund (AcRF) Tier 1 grant (21SISSMU031).

References

Akoglu, L.; Tong, H.; and Koutra, D. 2015. Graph based anomaly detection and description: a survey. *DMKD*, 29(3): 626–688.

Bandyopadhyay, S.; Vivek, S. V.; and Murty, M. 2020. Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proc. WSDM*, 25–33.

Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; and Krishnan, D. 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proc. CVPR*, 3722–3731.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *Proc. SIGMOD*, 93–104.

Chen, Z.; Liu, B.; Wang, M.; Dai, P.; Lv, J.; and Bo, L. 2020. Generative Adversarial Attributed Network Anomaly Detection. In *Proc. CIKM*, 1989–1992.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Adv. NIPS*, 26.

Ding, K.; Li, J.; Bhanushali, R.; and Liu, H. 2019. Deep anomaly detection on attributed networks. In *Proc. SDM*, 594–602. SIAM.

Ding, K.; Shu, K.; Shan, X.; Li, J.; and Liu, H. 2022. Cross-Domain Graph Anomaly Detection. *IEEE TNNLS*, 33(6): 2406–2415.

Ding, K.; Zhou, Q.; Tong, H.; and Liu, H. 2021. Few-Shot Network Anomaly Detection via Cross-Network Meta-Learning. In *Proc. WWW*, 2448–2456.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *Proc. ICML*, 1180–1189.

Gao, J.; Liang, F.; Fan, W.; Wang, C.; Sun, Y.; and Han, J. 2010. On community outliers and their efficient detection in information networks. In *Proc. SIGKDD*, 813–822.

Gretton, A.; Borgwardt, K.; Rasch, M.; Schölkopf, B.; and Smola, A. 2006. A kernel method for the two-sample-problem. *Adv NIPS*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Adv. NIPS*, 30.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *Proc. ICML*, 1989–1998.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *Proc. of ICLR*.

Lee, C.-Y.; Batra, T.; Baig, M. H.; and Ulbricht, D. 2019. Sliced Wasserstein discrepancy for unsupervised domain adaptation. In *Proc. CVPR*, 10285–10295.

Li, J.; Dani, H.; Hu, X.; and Liu, H. 2017. Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In *Proc. IJCAI*, 2152–2158.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *Proc. ICDM*, 413–422. IEEE.

Liu, Y.; Li, Z.; Pan, S.; Gong, C.; Zhou, C.; and Karypis, G. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE TNNLS*, 33(6): 2378–2392.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised domain adaptation with residual transfer networks. *Proc. NIPS*.

Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q. Z.; Xiong, H.; and Akoglu, L. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Tran. TKDE*.

Pang, G.; Cao, L.; Chen, L.; and Liu, H. 2018. Learning Representations of Ultrahigh-Dimensional Data for Random Distance-Based Outlier Detection. In *Proc. SIGKDD*, 2041–2050.

Pang, G.; Shen, C.; Cao, L.; and Hengel, A. V. D. 2021. Deep learning for anomaly detection: A review. *ACM CSUR*, 54(2): 1–38.

Pang, G.; Shen, C.; and van den Hengel, A. 2019. Deep anomaly detection with deviation networks. In *Proc. SIGKDD*, 353–362.

Peng, Z.; Luo, M.; Li, J.; Liu, H.; and Zheng, Q. 2018. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In *IJCAI*, 3513–3519.

Perozzi, B.; and Akoglu, L. 2016. Scalable anomaly ranking of attributed neighborhoods. In *Proc. ICDM*, 207–215.

Saito, K.; Ushiku, Y.; Harada, T.; and Saenko, K. 2018a. Adversarial Dropout Regularization. *Proc. ICLR*.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018b. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proc. CVPR*, 3723–3732.

Shen, J.; Qu, Y.; Zhang, W.; and Yu, Y. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Proc. AAAI*, 4058–4065.

Shen, X.; Dai, Q.; Chung, F.-l.; Lu, W.; and Choi, K.-S. 2020. Adversarial deep network embedding for cross-network node classification. In *Proc. AAAI*, 2991–2999.

Tzeng, E.; Hoffman, J.; Darrell, T.; and Saenko, K. 2015. Simultaneous deep transfer across domains and tasks. In *Proc. CVPR*, 4068–4076.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *Proc. CVPR*, 7167–7176.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2018. Deep graph infomax. *Proc. ICLR*.

Wang, Q.; Salehi, M.; Low, J. S.; Buntine, W.; and Leckie, C. 2022. ENDASh: Embedding Neighbourhood Dissimilarity with Attribute Shuffling for Graph Anomaly Detection. In *PAKDD*, 17–29.

Wilson, G.; and Cook, D. J. 2020. A survey of unsupervised deep domain adaptation. *ACM Tran. TIST*, 11(5): 1–46.

Wu, M.; Pan, S.; Zhou, C.; Chang, X.; and Zhu, X. 2020. Unsupervised Domain Adaptive Graph Convolutional Networks. In *Proc. WWW*, 1457–1467.

Wu, M.; Pan, S.; and Zhu, X. 2022. Attraction and Repulsion: Unsupervised Domain Adaptive Graph Contrastive Learning Network. *IEEE Tran. TETCI*.

Xiao, N.; and Zhang, L. 2021. Dynamic weighted learning for unsupervised domain adaptation. In *Proc. CVPR*, 15242–15251.

Xie, S.; Zheng, Z.; Chen, L.; and Chen, C. 2018. Learning semantic representations for unsupervised domain adaptation. In *Proc. CVPR*, 5423–5432.

Zhang, Y.; Song, G.; Du, L.; Yang, S.; and Jin, Y. 2019. Dane: Domain adaptive network embedding. In *Proc. IJCAI*.

Zhao, T.; Deng, C.; Yu, K.; Jiang, T.; Wang, D.; and Jiang, M. 2020. Error-Bounded Graph Anomaly Loss for GNNs. In *Proc. CIKM*, 1873–1882.