Appendix of "Truncated Affinity Maximization for Graph Anomaly Detection"

Anonymous Author(s)
Affiliation
Address
email

A One-Class Homophily Phenomenon

Fig. 1 shows the one-class homophily phenomenon on the rest of four datasets, in which ACM is a dataset with injected anomalies while the other three datasets contain real anomalies. The results here are consistent with that in the main text.

Given the graph along with the ground truth labels, following [3, 10], the homophily and heterophily are defined as the ratio of the edges connecting the node with the same classes and different classes, respectively:

$$\begin{cases} X_{\text{hetero}}(v) = \frac{1}{|\mathcal{N}(v)|} \left|\{u : u \in \mathcal{N}(v), y_u \neq y_v\}\right| \\ X_{\text{homo}}(v) = \frac{1}{|\mathcal{N}(v)|} \left|\{u : u \in \mathcal{N}(v), y_u = y_v\}\right| \end{cases} \tag{1}$$

where y_v is the label to denote whether the node v is an anomaly or not.

B Description of Datasets

We conduct the experiments on two real-world dataset with injected anomalies and four real-world with genuine anomalies in diverse online shopping services, social networks, and citation networks, including BlogCatalog [15], ACM[14], Amazon [2], Facebook [16], Reddit and YelpChi [5]. The statistical information including the number of nodes, edge, the dimension of the feature, and the anomalies rate of the datasets can be found in Tab. 1.

Particularly, BlogCatalog is a social blog directory where users can follow each other. Each node represents a user, and each link indicates the following relationships between two users. The attributes of nodes are the tags that describe users and their blogs. ACM is a citation graph dataset where the nodes denote the published papers and the edge denotes the citations relationship between the papers. The attributes of each node are the content of the corresponding paper. BlogCatalog and ACM are popular GAD datasets where the anomalies are injected ones, including structural anomalies and contextual anomalies, which are created following the prior work [8]. Amazon is a graph dataset capturing the relations between users and product reviews. Following [2, 17], three different user-user graph datasets are derived from Amazon using different adjacency matrix construction approaches. In this work, we focus on the Amazon-UPU dataset that connects the users who give reviews to at least one same product. The users with less than 20% are treated as anomalies. Facebook [16] is a social network where users build relationships with others and share their same friends. Reddit is a network of forum posts from the social media Reddit, in which the user who has been banned from the platform is annotated as an anomaly. Their post texts were converted to the vector as their attribute. YelpChi includes hotel and restaurant reviews filtered (spam) and recommended (legitimate) by Yelp. Following [11, 13], three different graph datasets derived from Yelp using different connections in user, product review text, and time. In this work, we only use YelpChi-RUR which connects reviews posted by the same user. Note that considering it's difficult to conduct an evaluation on the isolated nodes in the graph, they were removed before modeling.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

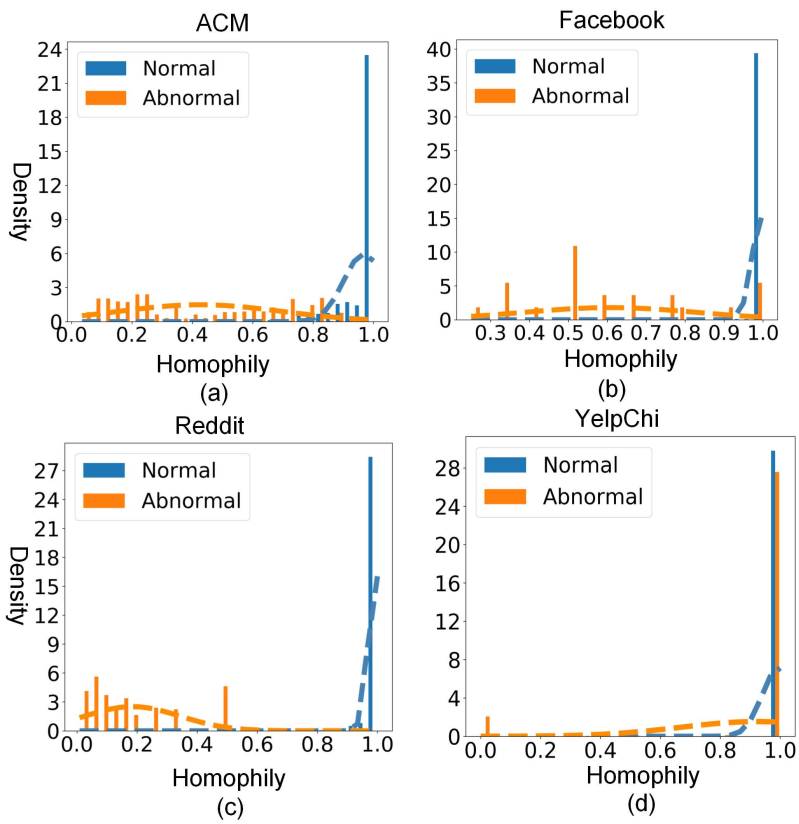

Figure 1: Homophily distribution of normal nodes and abnormal nodes on the rest of four datasets

Table 1: Key statistics of the datasets. The real-world datasets with Injected/Real anomalies(I/R).

Data set	Type	R/I	Nodes	Edges	Attributes	Anomalies(Rate)
BlogCatalog	Social Networks	I	5,196	171,743	8,189	300(5.77%)
ACM	Citation Networks	I	16,484	71,980	8,337	597(3.63%)
Amazon	Co-review	R	11,944	175,608	25	796(6.66%)
Facebook	Social Networks	R	4,039	88,234	576	27(0.67%)
Reddit	Social Networks	R	10,984	175,608	64	366(3.33%)
YelpChi	Co-review	R	45,954	49,315	32	1,217(2.65%)

C Additional Experimental Results

C.1 Hyperparameter Analysis

This section analyzes the sensitivity of TAM w.r.t. two key hyperparameters, including the regularization hyperparameter λ and the ensemble parameters T. The results on λ and T are shown in Fig. 2 and Fig. 3, respectively.

Ensemble Parameter T. As shown in Fig. 2 with increasing T, TAM generally performs better and becomes stable around $T \approx 4$. This is mainly because the use of more ensemble models on truncated graphs reduces the impact of the randomness of truncation and increases the probability of weakening the affinity of abnormal nodes to its neighbors, and this effect would diminish when T is sufficiently large. The average of local affinity from multiple truncated graph sets is more conducive to anomaly detection.

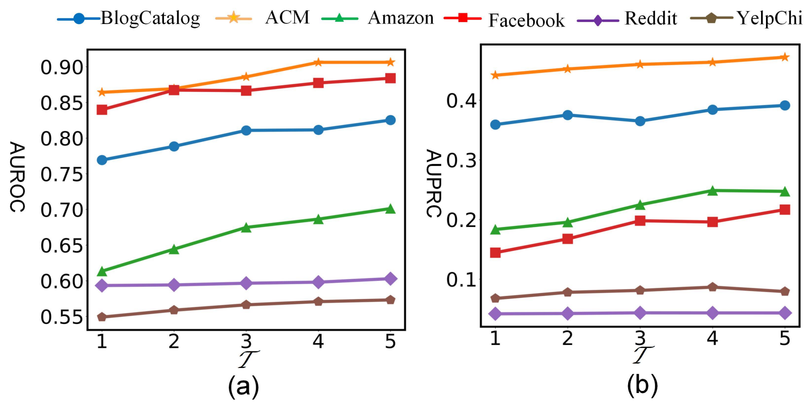

Figure 2: AUROC and AUPRC results w.r.t. # of ensemble parameter T

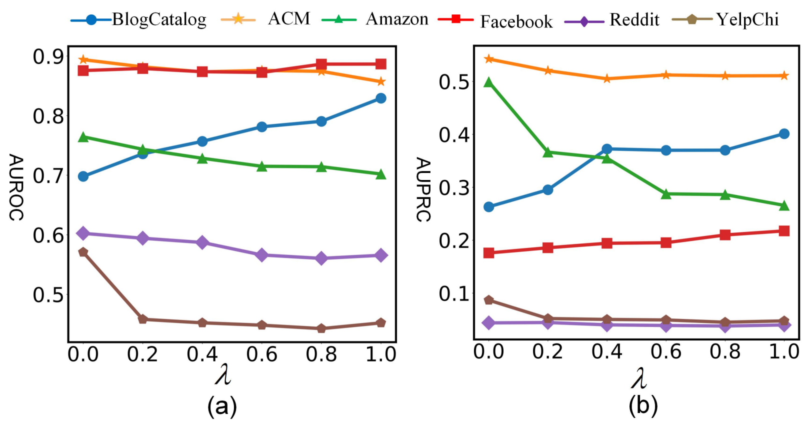

Figure 3: AUROC and AUPRC results w.r.t. # of regularization hyperparameter λ

Regularization Hyperparameter λ. In order to evaluate the effectiveness of λ, we adopt different values to adjust the weight of the regularization. From Fig. 3, we can see that for BlogCatalog and Facebook, adding the regularization improves the effectiveness of the model by a large margin. This is mainly because the use of regularization can prevent all nodes from having identical feature representations. For most real-world datasets with genuine anomalies, the regularization does not significantly improve the effectiveness of the model while decreasing the performance as the increasing of λ. The main reason is Amazon, Reddit, and YelpChi are real-world datasets with diverse attributes, and the role of regularization items is not reflected during affinity maximization.

C.2 Complexity Analysis

This subsection analyzes the time complexity of TAM. Specifically, the distance calculation takes $\mathcal{O}(md_0)$, m is the number of non-zero elements in the adjacent matrix $\mathbf{A}$, and d_0 is the dimension of attributes for each node. The graph truncation in TAM takes $2N\eta$, where N is the number of nodes and η is the average degree in the graph. In LAMNet, we build a GCN for each truncated graph, which takes $\mathcal{O}(md_1h)$, where h and d_1 denotes the number of feature maps and feature dimensions in graph convolution operation, respectively. The construction of a GCN takes $\mathcal{O}(md_1h)$. LAMNet also needs to compute all connected pairwise similarities, which takes $\mathcal{O}(d_1m)$. Thus, the overall complexity of TAM is $\mathcal{O}(md_0 + (d_1m + md_1h + 2N\eta)KT)$, where K is the truncation depth and T is the ensemble parameter. The complexity is lower than the time complexity in many existing GNN-based graph anomaly detection methods based on the subgraph sampling and hop counting [18, 4].

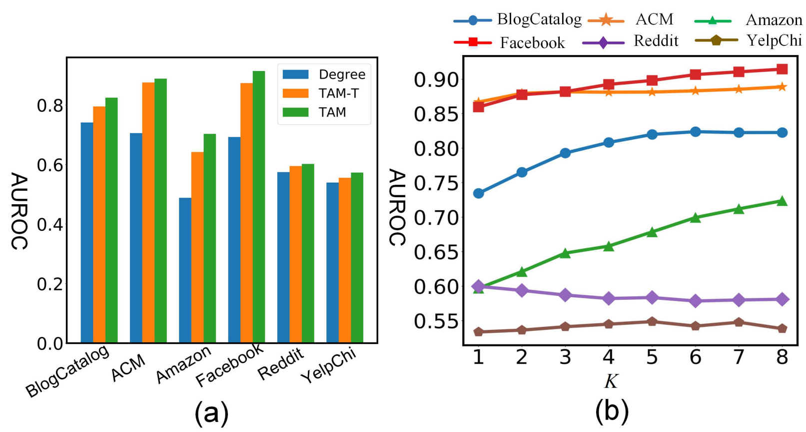

Figure 4: TAM results w.r.t. # (aggregation score) of graph truncation depth K

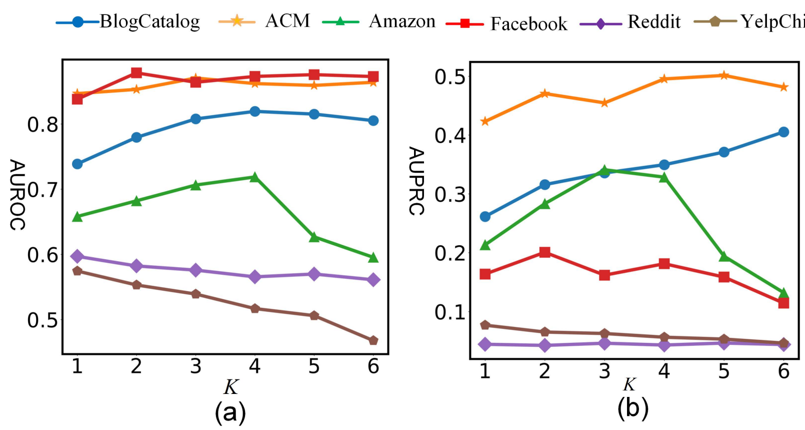

Figure 5: AUROC and AUPRC results of TAM using single-truncation scale-based anomaly scores

C.3 Anomaly Scoring

AUROC Results of TAM and Its Variants. We present the AUPRC results of TAM and its two variants, **Degree** and **TAM-T** in the main text, where TAM can consistently and significantly outperform both variants. Similar observation can also be found from the AUROC results in Fig. 4(a). Fig. 4(b) shows the AUROC results of anomaly scoring by aggregating the anomaly scores under all truncation scales/depths. Similar to the AUPRC results in the main text, the anomaly scores obtained from different truncation scales/depths can largely improve the detection performance and the results become stable with increasing graph truncation depth K.

Anomaly Scoring Using Multi-scale Truncation vs Single-scale Truncation. Fig. 5 shows the results of TAM that performs anomaly scoring based on a single-scale graph truncation rather than the default multi-scale graph truncation. As shown in Fig. 5, the increase of K improves the performance on large datasets such as the Amazon datasets, but it often downgrades the performance on the datasets such as Reddit and YelpChi. The main reason is that the node attributes in these datasets are more similar than the other datasets, restricting the effect of graph truncation. However, the opposite case can occur on the other datasets. To tackle this issue, we define the overall anomaly score as an average score over the anomaly scores obtained from the LAMNets built using truncated graphs under all truncation depths/scales. This resulting multi-scale anomaly score, as shown in Fig. 4(b), performs much more stably than the single-scale anomaly score.

Table 2: AUROC and AUPRC results of TAM using shared-weight LAMNets (TAM-S) vs non-shared-weight LAMNets (TAM).

Metric	Method	Dataset					
		BlogCatalog	ACM	Amazon	Facebook	Reddit	YelpChi
AUROC	TAM-S	$0.8170_{\pm 0.002}$	$0.8826_{\pm 0.003}$	$0.7044_{\pm 0.008}$	$\mathbf{0.9165}_{\pm 0.005}$	$0.6008_{\pm 0.002}$	$0.5407_{\pm 0.008}$
	TAM	$\mathbf{0.8248}_{\pm 0.003}$	$\mathbf{0.8878}_{\pm 0.024}$	$\mathbf{0.7064}_{\pm 0.010}$	$0.9144_{\pm 0.008}$	$\mathbf{0.6023}_{\pm 0.004}$	$\mathbf{0.5643}_{\pm 0.007}$
AUPRC	TAM-S	$0.3908_{\pm 0.002}$	$0.4960_{\pm 0.001}$	$0.2597_{\pm 0.002}$	$0.2087_{\pm 0.006}$	$\mathbf{0.0459}_{\pm 0.003}$	$0.0691_{\pm 0.002}$
	TAM	$\mathbf{0.4182}_{\pm 0.005}$	$\mathbf{0.5124}_{\pm 0.018}$	$\mathbf{0.2634}_{\pm 0.008}$	$\mathbf{0.2233}_{\pm 0.016}$	$0.0446_{\pm 0.001}$	$\mathbf{0.0778}_{\pm 0.009}$

C.4 Sharing Weights LAMNet vs. No Sharing Weights LAMNet

In our experiments, the weight parameters in LAMNets are independent from each other by default, i.e., GNNs in LAMNets are independently trained. In this section, we compare TAM with its variant **TAM-S** where all LAMNets use a single GNN backbone with shared weight parameter. The results are shown in Tab. 2. It is clear that TAM performs consistently better than, or comparably well to, TAM-S across the six datasets.

D Description of algorithms

D.1 Competing Methods

- iForest [6] builds multiple trees to isolate the data based on the node's feature. It has been widely used in outlier detection.
- ANOMALOUS [12] proposes a joint network to conduct the selection of attributes in the CUR decomposition and residual analysis. It can avoid the adverse effects brought by noise.
- DOMINANT [1] leverages the auto-encoder for graph anomaly detection. It consists of an encoder layer and a decoder layer which construct the feature and structure of the graph. The reconstruction errors from the feature and structural module are combined as the anomaly score.
- HCM-A [4] constructs an anomaly indicator by estimating hop count based on both global and local contextual information. It also employs Bayesian learning in predicting the shortest path between node pairs.
- CoLA [8] exploits the local information in a contrastive self-supervised framework. They define the positive pair and negative pair for a target node. The anomaly score is defined as the difference value between its negative and positive score.
- SL-GAD [18] constructs two modules including generative attribute regression and multi-view contrastive for anomaly detection based on CoLA. The anomaly score is generated from the degree of mismatch between the constructed and original features and the discrimination scores.
- ComGA [9] designs a tailor GCN to learn distinguishable node representations by explicitly capturing community structure.

Their implementation is taken directly from their official web pages or the widely-used PyGOD library [7]. The links to the source code pages are as follows:

- iForest: https://github.com/pygod-team/pygod
- ANOMALOUS: https://github.com/pygod-team/pygod
- DOMIANT: https://github.com/kaize0409/GCN_AnomalyDetection_pytorch
- HCM-A: https://github.com/TienjinHuang/GraphAnomalyDetection
- CoLA: https://github.com/GRAND-Lab/CoLA:
- SL-GAD: https://github.com/yixinliu233/SL-GAD
- ComGA:https://github.com/XuexiongLuoMQ/ComGA

Algorithm 1 NSGT

Input: Attributed Graph, $G\!=(\mathcal{V},\mathcal{E})$, Distance Matrix, $\mathbf{D}$
Output: A truncated graph structure $\tilde{\mathcal{E}}$.

/* Regard the graph as a direct graph */
Initialize the directed graph truncation indicator $e_{(v_i,v_j)} = 1$ iff $(v_i, v_j) \in \mathcal{E}$
Find the mean d_{mean} from $\mathbf{D}$ using $d_{mean} = \frac{1}{m}\sum_{(v_i,v_j)\in\varepsilon} d_{ij}$
for each v in $\mathcal{V}$ **do**
  Find the maximum $d^{(v)}_{\max}$ from $\{dis(v,v'), (v,v') \in \mathcal{E}\}$
  if $d^{(v)}_{\max} > d_{mean}$ **then**
    Randomly sample r from $[d^{(v)}_{\max}, d_{mean}]$ for node v
    for each v' in $\{v', (v,v') \in \mathcal{E}\}$ **do**
      if $dis(v,v') > r$ **then**
        Plan to cut the edge v to v', i.e., $e_{(v,v')} \leftarrow 0$
      end if
    end for
  end if
end for
/* The edge will be removed only when both connected nodes see the edge as non-homophily edge */
for each v in $\mathcal{V}$ **do**
  for each v' in $\{v', (v,v') \in \mathcal{E}\}$ **do**
    Cut the edge between v and v', $\widetilde{\mathcal{E}} = \mathcal{E}\setminus((v,v') \cup (v',v))$; iff $e_{(v,v')} = e_{(v',v)} = 0$
  end for
end for
return The truncated graph structure $\tilde{\mathcal{E}}$

D.2 Pseudo Codes of TAM.

The training algorithms of TAM are summarized in Algorithm 1 and Algorithm 2. Algorithm 1 describes the process of NSGT. Algorithm 2 describes the training process of TAM.

References

[1] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 594–602. SIAM, 2019.

[2] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 315–324, 2020.

[3] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 357–365, 2023.

[4] Tianjin Huang, Yulong Pei, Vlado Menkovski, and Mykola Pechenizkiy. Hop-count based self-supervised anomaly detection on attributed networks. *arXiv preprint arXiv:2104.07917*, 2021.

[5] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1269–1278, 2019.

[6] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39, 2012.

Algorithm 2 TAM

Input: Graph, $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, N: Number of nodes, L: Number of layers, E: Training epochs, T: Ensemble parameter, K: Truncation depth.

Output: Anomaly scores of all nodes s.

Compute the Euclidean distance $\mathbf{D} = \{d_{ij}\}$ for each connected node pair $(v_i, v_j) \in \mathcal{E}$
Randomly initialize GNN $(\mathbf{h}_1^{(0)}, \mathbf{h}_2^{(0)}, ..., \mathbf{h}_N^{(0)}) \leftarrow \mathbf{X}$, $\mathcal{E}^{(1,0)}, , ..., \mathcal{E}^{(T,0)} \leftarrow \mathcal{E}$
for $k = 1, \cdots, K$ **do**
  for $t = 1, \cdots, T$ **do**
    /* Graph truncation and update the graph structure */
    $\mathcal{E}^{(t,k)} = NSGT(\mathcal{V}, \mathcal{E}^{(t,k-1)}, \mathbf{D})$
    /* LAMNet */
    for $epoch = 1, \cdots, E$ **do**
      for each v in $\mathcal{V}$ **do**
        for $l = 1, \cdots, L$ **do**
          $\mathbf{h}_{v,l} = \phi(\mathbf{h}_{v,l-1}; \Theta_{t,k})$
          $\mathbf{h}_{v,l} = \mathrm{ReLU}\left(\mathrm{AGG}(\{\mathbf{h}_{v',l} : (v, v') \in \mathcal{E}^{(t,k)}\})\right)$
        end for
        Calculate $f_{TAM}(v_i; \Theta_{t,k}, \mathbf{A}, \mathbf{X})$ by Eq (5).
      end for
      /* Affinity Maximization */
      Minimize $\sum_{v_i \in \mathcal{V}} \left(f_{TAM}(v_i; \Theta_{t,k}, \mathbf{A}, \mathbf{X}) + \lambda \frac{1}{|\mathcal{V} \setminus \mathcal{N}(v_i)|} \sum_{v_k \in \mathcal{V} \setminus \mathcal{N}(v_i)} \mathrm{sim}\left(\mathbf{h}_i, \mathbf{h}_k\right) \right)$
      Update $\mathbf{\Theta}_{t,k}$ by using stochastic gradient descent
    end for
  end for
end for
/* Aggregation Score */
return Anomaly Score by $s\left(v\right) = \frac{1}{T \times K} \sum_{k=1}^{K} \sum_{t=1}^{T} f_{TAM}(v_i; \Theta_{t,k}^{*}, \mathbf{A}, \mathbf{X}))$

[7] Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, et al. Bond: Benchmarking unsupervised outlier node detection on static attributed graphs. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[8] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6):2378–2392, 2021.

[9] Xuexiong Luo, Jia Wu, Amin Beheshti, Jian Yang, Xiankun Zhang, Yuan Wang, and Shan Xue. Comga: Community-aware attributed graph anomaly detection. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 657–665, 2022.

[10] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021.

[11] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. What yelp fake review filter might be doing? In *Proceedings of the international AAAI conference on web and social media*, volume 7, pages 409–418, 2013.

[12] Zhen Peng, Minnan Luo, Jundong Li, Huan Liu, and Qinghua Zheng. Anomalous: A joint modeling approach for anomaly detection on attributed networks. In *IJCAI*, pages 3513–3519, 2018.

[13] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pages 985–994, 2015.

[14] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998, 2008.

[15] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826, 2009.

[16] Zhiming Xu, Xiao Huang, Yue Zhao, Yushun Dong, and Jundong Li. Contrastive attributed network anomaly detection with data augmentation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 444–457. Springer, 2022.

[17] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 689–698, 2020.

[18] Yu Zheng, Ming Jin, Yixin Liu, Lianhua Chi, Khoa T Phan, and Yi-Ping Phoebe Chen. Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.