Appendixes of “A Label-free Heterophily-guided Approach for Unsupervised Graph Fraud Detection”

Anonymous submission

Appendix A: Related Work in Detail

Graph Fraud Detection

Graph fraud detection (GFD) aims to identify fraudulent activities from graph-structured real-world systems, including financial fraud (Motie and Raahemi 2023), spamming (Deng et al. 2022), and fake reviews (Yu et al. 2022). Various studies have proposed different models to resolve the problem. For example, GraphConsis (Liu et al. 2020) tackles the inconsistency problem by computing consistency between nodes, while CARE-GNN (Dou et al. 2020) designs a similarity-aware neighbor selector to address camouflage behaviors. Similarly, PC-GNN (Liu et al. 2021a) proposes a neighbor sampler to balance label distribution. Although these GNN-based methods have achieved promising results, they suffer from heterophily caused by the camouflages problem (Dou et al. 2020), i.e., benign nodes tend to have heterophilic connections with normal nodes to make them indistinguishable from the majority.

Recent studies try to develop advanced GFD approaches with the consideration of heterophily. With the help of labels, a variety of techniques have been employed to mitigate the impact of heterophily. H^2-FDetector (Shi et al. 2022) trains a classifier to identify connection types, guiding attention-based aggregation, whereas GHRN (Gao et al. 2023a) prunes heterophily edges instead. BWGNN (Tang et al. 2022) examines the right shift phenomenon to design an expressive encoder. GAGA (Wang et al. 2023) combines group aggregation and learnable encoding to fully utilize label information. GDN (Gao et al. 2023b) reveals the heterophily shift across training and testing datasets and mitigates the issue through feature separation, while ConsisGAD (Chen et al. 2024b) directly utilizes homophily patterns for distinguishing normal and anomalous nodes. SEC-GFD (Xu et al. 2024) employs spectrum analysis to aggregate frequent bands separately, and PMP (Zhuo et al. 2024) resolved the problem in spatial domains by aggregating homophilic and heterophilic neighbors independently.

Although these methods achieve promising results in mitigating heterophily, their reliance on labels limits their applicability in scenarios where such labels are unavailable. Due to the significant costs of annotating labels in GFD-related scenarios (Sehwag, Chiang, and Mittal 2021), there is a compelling need to create unsupervised GFD methods.

Graph Anomaly Detection

Graph anomaly detection (GAD) is a broader concept than GFD, aiming to identify not only fraudsters but also any rare and unusual patterns that significantly deviate from the distribution of the majority in graph data. Therefore, GAD techniques can be directly applied to GFD, particularly within unsupervised learning settings. Due to the broad scope of GAD and the difficulty in obtaining real-world anomalies, many unsupervised GAD methods have been designed and evaluated on several datasets with injected anomalies. For example, DOMINANT (Ding et al. 2019) uses reconstruction error to estimate the abnormality of nodes, while CoLA (Liu et al. 2021b) employs contrastive learning to compute ego-neighbor disagreement as anomaly score, which is adapted by later works (Jin et al. 2021; Zheng et al. 2021; Duan et al. 2023a,b; Pan et al. 2023).

In spite of the decent performances they have achieved, these methods rely on the homophily assumption, which limits their applications since heterophily is a ubiquitous property in graphs (Zheng et al. 2022). Recent studies explore this insufficiency and suggest using estimated anomaly scores to mitigate the negative impacts of heterophily, like dropping edges (He et al. 2024; Qiao and Pang 2024) or adjusted message passing (Chen et al. 2024a). However, their understanding of heterophily is fundamental and lacks a systematic methodology for defining the metric. Moreover, their estimation of heterophily heavily relies on embeddings computed by message-passing GNNs, which may be unreliable in the first place (Zhu et al. 2022). Consequently, their estimated anomaly scores for GFD datasets could be poor, even worse than randomly generated scores (Zhao and Akoglu 2020). To fill the gap, our proposed HUGE first introduces a simple yet effective heterophily measure with a set of desired properties to address the aforementioned insufficiency, which can be further used to guide the learning of the downstream unsupervised GFD model.

Appendix B: Proof of Existing Unsupervised Edge-level Heterophily Metrics

In this section, we validate the statements listed in Table 1 by offering mathematical proofs or presenting counterexamples to illustrate their limitations in Graph Fraud Detection (GFD).

Euclidean Distance

Theorem 1. *Euclidean distance is unbounded. It satisfies minimal agreement, monotonicity and equal attribute tolerance.*

Proof. It is trivial to prove that the Euclidean distance is unbounded and satisfies minimal agreement, as the Euclidean distance between two identical vectors is zero and a zero distance implies the vectors are identical.

To prove equal attribute tolerance, we have:

$$\begin{aligned}\|[\mathbf{x}_a^T \| k]^T - [\mathbf{x}_b^T \| k]^T\|_2 =& \sqrt{\sum_{i=0}^{d}(x_{a,i} - x_{b,i})^2 + (k-k)^2} \\ =& \sqrt{\sum_{i=0}^{d}(x_{a,i} - x_{b,i})^2} \\ =& \|\mathbf{x}_a - \mathbf{x}_b\|_2.\end{aligned} \tag{A.1}$$

Hence, the Euclidean distance satisfies equal attribute tolerance.

To prove monotonicity, we first compute the derivative of the Euclidean distance with respect to (w.r.t.) $\mathbf{x}_{a,i}$:

$$\begin{aligned}f' =& \frac{\partial(\|\mathbf{x}_a - \mathbf{x}_b\|_2)}{\partial x_{a,i}} \\ =& \frac{\partial\big(\sqrt{\sum_{j=1}^{d}(x_{a,j} - x_{b,j})^2}\big)}{\partial x_{a,i}} \\ =& \frac{x_{a,i} - x_{b,i}}{\|\mathbf{x}_a - \mathbf{x}_b\|_2}.\end{aligned} \tag{A.2}$$

Hence, the derivative f' is positive when $x_{a,i} - x_{b,i} > 0$, and f' is negative otherwise. Therefore, Euclidean distance satisfies the conditions for monotonicity. □

Cosine Distance

Theorem 2. *Cosine distance is bounded. It does not satisfies minimal agreement, monotonicity and equal attribute tolerance.*

Proof. The range of cosine distance is $[-1, 1]$, which proves its boundedness. Meanwhile, the cosine distance between parallel vectors with different norm is always -1, indicating that it does not satisfy minimal agreement or monotonicity. Furthermore, it fails to meet the criteria of equal attribute tolerance. For example, $\text{cd}([1,2]^T, [3,4]^T) = 0.016$, while $\text{cd}([1,2,5]^T, [3,4,5]^T) = 0.070$, where $\text{cd}(\cdot,\cdot)$ denotes cosine distance. □

Attribute Heterophily Rate

Theorem 3. *The attribute heterophily rate satisfies both boundedness and minimal agreement. It does not satisfies monotonicity and equal attribute tolerance.*

Proof. Given that the range of the indicator function $\mathbf{1}$ is $\{0, 1\}$, it is straightforward to prove that the attribute heterophily rate is bounded by [0, 1] and satisfies minimal agreement. However, as the indicator function only checks whether the two values are identical, it fails to satisfy monotonicity. For equal attribute tolerance, we have:

$$\begin{aligned}\text{AHR}\left(\begin{bmatrix}\mathbf{x}_a \\ k\end{bmatrix}, \begin{bmatrix}\mathbf{x}_b \\ k\end{bmatrix}\right) =& \frac{\sum_{i=1}^{d}\mathbf{1}[x_{a,i} \neq x_{b,i}] + \mathbf{1}[k \neq k]}{d+1} \\ =& \frac{d\text{AHR}(\mathbf{x}_a, \mathbf{x}_b) + 1}{d+1},\end{aligned} \tag{A.3}$$

where $\text{AHR}(\cdot,\cdot)$ denotes attribute heterophily rate. Hence, the attribute heterophily rate does not satisfy equal attribute tolerance unless $\text{AHR}(\mathbf{x}_a, \mathbf{x}_b) = 1$, which occurs only when $\mathbf{x}_a = \mathbf{x}_b$. □

Appendix C: Proof of Harmonic Label-Free Heterophily

In this section, we provide a detailed proof to show that our edge-level harmonic label-free heterophily (HALO) satisfies boundedness, minimal agreement and equal attribute tolerance, while it satisfies monotonicity under a relaxed constraint, where the length of feature vectors are constant.

As a recap, our HALO is defined as:

$$\text{HALO}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\|\hat{\mathbf{x}}_a - \hat{\mathbf{x}}_b\|_2}{(\|\hat{\mathbf{x}}_a\|_2^2 + \|\hat{\mathbf{x}}_b\|_2^2 + \epsilon)^{\frac{1}{2}}}, \tag{A.4}$$

where $\hat{\mathbf{x}}_a = \text{abs}(\mathbf{x}_a - \mathbf{x}_b) \odot \mathbf{x}_a$, $\hat{\mathbf{x}}_b = \text{abs}(\mathbf{x}_a - \mathbf{x}_b) \odot \mathbf{x}_b$ are preprocessed attributes to emphasize heterophilic patterns. The abs, $\odot$, ϵ denote element-wise absolute value, multiplication, a small positive number to keep the denominator larger than zero, respectively.

Theorem 4 (Boundedness)**.** *HALO satisfies boundedness.*

Proof. It is trivial to prove HALO has a minimum value of 0. For the proof of the upper bound, with the triangle inequality of vectors, we have:

$$\begin{aligned}\|\hat{\mathbf{x}}_a - \hat{\mathbf{x}}_b\|_2^2 =& \sum_{i=1}^{d}\hat{x}_{a,i}^2 - 2\hat{x}_{a,i}\hat{x}_{b,i} + \hat{x}_{b,i}^2 \\ \leq& \|\hat{\mathbf{x}}_a\|_2^2 + \|\hat{\mathbf{x}}_b\|_2^2 + 2\|\hat{\mathbf{x}}_a\|_2\|\hat{\mathbf{x}}_b\|_2 \\ \leq& 3(\|\hat{\mathbf{x}}_a\|_2^2 + \|\hat{\mathbf{x}}_b\|_2^2).\end{aligned} \tag{A.5}$$

Therefore, HALO is bounded by 0 and $\sqrt{3}$. □

Theorem 5 (Minimal Agreement)**.** *HALO satisfies minimal agreement.*

Proof. Since 0 is the lower bound of HALO and the denominator is always larger than zero, we only need to prove that the numerator reaches 0 if and only if the two attributes are identical. Given that the Euclidean distance satisfies minimal agreement, our HALO also satisfies minimal agreement. □

Theorem 6 (Equal Attribute Tolerance)**.** *HALO satisfies equal attribute tolerance.*

Proof. Since $\hat{\mathbf{x}}_a = \text{abs}(\mathbf{x}_a - \mathbf{x}_b) \odot \mathbf{x}_a$, $\hat{\mathbf{x}}_b = \text{abs}(\mathbf{x}_a - \mathbf{x}_b) \odot \mathbf{x}_b$, we have:

$$\begin{aligned} \|[\mathbf{x}_a^T \hat{|} k]^T\|_2 =& \sqrt{\sum_{i=1}^{d} \hat{x}_{a,i}^2 + (k-k)k} \\ =& \|\hat{\mathbf{x}}_a\|_2 \end{aligned} \tag{A.6}$$

Similarly, $\|[\mathbf{x}_b^T \hat{|} k]^T\|_2 = \|\hat{\mathbf{x}}_b\|_2$. By equation (A.1), we have $\|[\mathbf{x}_a^T \hat{|} k]^T - [\mathbf{x}_b^T \hat{|} k]^T\|_2 = \|\hat{\mathbf{x}}_a - \hat{\mathbf{x}}_b\|_2$. Hence, by substituting these equations into equation (A.4) we prove that HALO$([\mathbf{x}_a^T|k]^T, [\mathbf{x}_b^T|k]^T) =$ HALO$(\mathbf{x}_a, \mathbf{x}_b)$, i.e. our HALO satisfies equal attribute tolerance.

□

Theorem 7 (Monotonicity)**.** *HALO satisfies monotonicity where the length of feature vectors are constant.*

Proof. To facilitate the analysis of the monotonicity of HALO, we have:

$$\begin{aligned} h &= \frac{\|\hat{\mathbf{x}}_a - \hat{\mathbf{x}}_b\|_2}{(\|\hat{\mathbf{x}}_a\|_2^2 + \|\hat{\mathbf{x}}_b\|_2^2 + \epsilon)^{\frac{1}{2}}} \\ &= \frac{\sqrt{\sum_{k=1}^{n} (\hat{x}_{a,k} - \hat{x}_{b,k})^2}}{(\|\hat{\mathbf{x}}_a\|_2^2 + \|\hat{\mathbf{x}}_b\|_2^2 + \epsilon)^{\frac{1}{2}}} \\ &= \frac{\sqrt{(\hat{x}_{a,i} - \hat{x}_{b,i})^2 + \sum_{k=1,k\neq i}^{n} (\hat{x}_{a,k} - \hat{x}_{b,k})^2}}{(\|\hat{\mathbf{x}}_a\|_2^2 + \|\hat{\mathbf{x}}_b\|_2^2 + \epsilon)^{\frac{1}{2}}} \end{aligned} \tag{A.7}$$

To simplify the prove and analysis, we assume the length of attributes, i.e., $\|\hat{\mathbf{x}}_a\|_2$ and $\|\hat{\mathbf{x}}_b\|_2$ are constant values to relax the constraint of monotonicity, which can be achieved by feature normalization in practice. Hence, we can compute the derivative of h:

$$\begin{aligned} \frac{\partial h}{\partial \hat{x}_{a,i}} &= \frac{\hat{x}_{a,i} - \hat{x}_{b,i}}{(\|\hat{\mathbf{x}}_a\|_2^2 + \|\hat{\mathbf{x}}_b\|_2^2 + \epsilon)^{\frac{1}{2}} \sqrt{\sum_{k=1}^{n} (\hat{x}_{a,k} - \hat{x}_{b,k})^2}} \\ &= m(\hat{x}_{a,i} - \hat{x}_{b,i}), \end{aligned} \tag{A.8}$$

where m is a positive number. Therefore, HALO satisfies monotonicity under the relaxed constraint.

Smart fraudsters often hide themselves among benign users by manipulating their attributes to appear less distinguishable. Therefore, we rescale the attributes to uncover the hidden suspicious patterns. While this operation relaxes the constraint of monotonicity, it is better suited for unsupervised graph fraud detection scenarios. □

Appendix D: Algorithm Complexity

In this section, we discuss the time complexity of each component in HUGE respectively. We denote the dimension of embedding as d_e. For computing HALO, the complexity is $\mathcal{O}(md)$. For the alignment-based fraud detection module, the forward pass through MLP layers and GNN layers have a complexity of $\mathcal{O}(ndd_e)$ and $\mathcal{O}(md_e^2)$ respectively. The time complexity of computing fraud score is $\mathcal{O}(md_e)$. To compute the ranking loss, the complexity is $\mathcal{O}(n|B|d_e)$, and the complexity for computing asymmetric alignment loss is the same as computing fraud score. To sum up, the heterophily estimation, training and testing complexity of HUGE are $\mathcal{O}(md)$, $\mathcal{O}\big((nd + md_e + n|B|)d_e\big)$ per epoch, and $\mathcal{O}\big((nd+m)d_e\big)$, respectively. The training algorithm of HUGE is summarized in Algorithm 1.

Params.	Dataset	Range
Epoch	Amazon, Facebook, Reddit, YelpChi	300 (Fixed)
	AmazonFull, YelpChiFull	$\{5, 10\}$
lr	Amazon, Facebook, Reddit, YelpChi, AmazonFull	$\{0.001, 0.0005, 0.0001\}$
	YelpChiFull	$\{0.001, 0.0001, 0.00001\}$
α	Amazon, Facebook, Reddit, YelpChi	$\{0, 0.5, 1, 1.5, 2\}$
	AmazonFull, YelpChiFull	$\{0, 1, 2, 3\}$

Table A.1: Range of grid search.

Datasets	Epoch	lr	α
Amazon	300	0.0005	0.5
Facebook	300	0.0005	0.5
Reddit	300	0.0005	0.5
YelpChi	300	0.0005	0.5
AmazonFull	10	0.0005	1
YelpChiFull	5	0.00001	3

Table A.2: Hyper-parameters of HUGE.

Appendix E: Implementation Details

Environment

HUGE is implemented with the following libraries and their respective versions: Python 3.9.18, CUDA version 11.7, PyTorch 2.0.0, DGL 1.1.2, torch-cluster 1.6.3, torch-sparse 0.6.18.

Hardware Configuration

All the experiments are conducted on a Windows desktop equipped with an AMD Ryzen 5800X processor, 32 GB of RAM, and a single NVIDIA GeForce RTX4090 GPU with 24GB of VRAM.

Hyper Parameters

For the backbone of HUGE, we use two layers of MLPs followed by one layer of GNN, with 128 hidden perceptrons. In addition, we summarize other hyper-parameters of HUGE in Table A.1. By default, we utilize a learning rate of 0.0005, epoch of 300, batch size of 8192 and set α to 0.5. For the

Metric	Method	Dataset					
		Amazon	Facebook	Reddit	YelpChi	AmazonFull	YelpChiFull
AUROC	DOMINANT	0.4845±0.0064	0.4372±0.0046	0.5588±0.0043	0.3993±0.0030	0.4496±0.0683	OOM
	CoLA	0.4734±0.0044	0.8366±0.0268	**0.6032±0.0057**	0.4336±0.0177	0.2110±0.0036	<u>0.4911±0.0008</u>
	ANEMONE	0.5757±0.0056	0.8385±0.0104	0.5853±0.0047	0.4432±0.0129	0.6056±0.0037	0.4601±0.0017
	GRADATE	0.5539±0.0063	0.8809±0.0100	0.5671±0.0066	OOM	OOM	OOM
	ADA-GAD	0.5240±0.0008	0.0756±0.0007	0.5610±0.0002	OOM	OOM	OOM
	GADAM	0.6167±0.0133	<u>0.9539±0.0067</u>	0.5809±0.0053	0.4177±0.0168	0.4457±0.0205	0.4797±0.0085
	TAM	<u>0.7126±0.0111</u>	0.8895±0.0057	0.5748±0.0021	<u>0.5473±0.0022</u>	<u>0.6442±0.0232</u>	OOM
	HUGE (ours)	**0.8516±0.0029**	**0.9760±0.0008**	<u>0.5906±0.0042</u>	**0.6013±0.0076**	**0.8892±0.0026**	**0.5767±0.0064**
AUPRC	DOMINANT	0.0589±0.0007	0.0194±0.0002	0.0371±0.0002	0.0385±0.0002	0.0574±0.0081	OOM
	CoLA	0.0683±0.0007	0.2155±0.0417	0.0440±0.0015	0.0448±0.0018	0.0536±0.0013	<u>0.1419±0.0004</u>
	ANEMONE	0.1054±0.0054	0.2320±0.0299	0.0417±0.0008	0.0473±0.0017	<u>0.2396±0.0133</u>	0.1305±0.0007
	GRADATE	0.0892±0.0021	0.3560±0.0300	0.0389±0.0007	OOM	OOM	OOM
	ADA-GAD	0.1108±0.0006	0.0122±0.0000	0.0382±0.0021	OOM	OOM	OOM
	GADAM	0.0857±0.0050	<u>0.3630±0.0144</u>	<u>0.0465±0.0018</u>	0.0423±0.0015	0.0566±0.0027	0.1351±0.0028
	TAM	<u>0.2915±0.0373</u>	0.1937±0.0172	0.0438±0.0005	**0.0780±0.0014**	0.2188±0.0314	OOM
	HUGE (ours)	**0.6672±0.0077**	**0.3674±0.0091**	**0.0511±0.0022**	<u>0.0708±0.0026</u>	**0.7668±0.0022**	**0.1869±0.0026**

Table A.3: Complete results (mean ± std) of main experiment. The best and second-best results are in **bold** and <u>underlined</u>, respectively. OOM indicates out-of-memory on a 24GB GPU.

Measures	Amazon	Facebook	AmazonFull	YelpChiFull
HUGE (w/ HALO)	**0.8516±0.0029**	**0.9760±0.0008**	**0.8892±0.0026**	**0.5767±0.0064**
w/ Euc. Dist.	0.7128±0.0061	0.9251±0.0074	0.8746±0.0026	0.5763±0.0064
w/ Cos. Dist.	0.7018±0.0052	0.9668±0.0024	0.8735±0.0026	0.5762±0.0063
w/ AHR	0.8171±0.0064	0.9438±0.0029	0.8622±0.0024	0.5765±0.0064
w/o Alignment	0.5614±0.0019	0.9487±0.0026	0.8746±0.0033	0.5295±0.0204
w/o GNN	0.5548±0.0024	0.9401±0.0022	0.7005±0.0136	0.5533±0.0088

Table A.4: Complete results (mean±std) of ablation study of key designs in HUGE.

large datasets Amazon-Full and YelpChi-Full, we reduce the batch size to 512 to avoid out-of-memory issue. we also adjust the training epochs to 10 and 5 respectively to accelerate training, while increase the α to place greater emphasis on neighbor information. We obtain the best combination of hyper-parameters via grid search. The details of grid search are summarized in Table A.2. For all experiments, we record the average performance and standard deviation over five runs, using random seeds $\{0, 1, 2, 3, 4\}$.

Appendix F: More Results

In this section, we propose additional experimental results that are not included in the main paper due to the length constraints. The complete results of the main experiments and ablation study are included in Table A.3 and Table A.4, respectively.

References

Chen, J.; Zhu, G.; Yuan, C.; and Huang, Y. 2024a. Boosting Graph Anomaly Detection with Adaptive Message Passing. In *The Twelfth International Conference on Learning Representations*.

Chen, N.; Liu, Z.; Hooi, B.; He, B.; Fathony, R.; Hu, J.; and Chen, J. 2024b. Consistency Training with Learnable Data Augmentation for Graph Anomaly Detection with Limited Supervision. In *The Twelfth International Conference on Learning Representations*.

Deng, L.; Wu, C.; Lian, D.; Wu, Y.; and Chen, E. 2022. Markov-driven graph convolutional networks for social spammer detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12310–12322.

Ding, K.; Li, J.; Bhanushali, R.; and Liu, H. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM international conference on data mining*, 594–602. SIAM.

Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; and Yu, P. S. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of*

Algorithm 1: HUGE

Input: Attributed Graph, $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, E: Training epochs, B: Batch size, α: Alignment parameter, lr: Learning rate.

Output: Fraud scores of all nodes $\mathbf{s}$.

```
Compute the label-free heterophily $\mathbf{h} = \{h_i\}$ for each
  node $v_i \in \mathcal{V}$ using HALO.
Randomly initialize the parameters of the MLP and
  GNN encoders.
// Training phase.
for $epoch = 1, ..., E$ do
    $\mathcal{B} \leftarrow$Randomly split $V$ into batches of size $B$.
    for $b = (v'_1, ..., v'_B) \in \mathcal{B}$ do
        $\mathbf{E} \leftarrow \mathrm{MLP}(\{\mathbf{x}_{v'_1}, ..., \mathbf{x}_{v'_B}\})$
        $\mathbf{E}_{\text{nei}} \leftarrow$Compute embeddings for neighbors
          of $b$ using MLP encoder.
        $\mathbf{A}_{\text{nei}} \leftarrow$Obtain neighborhood subgraphs of $b$.
        $\bar{\mathbf{E}} = \mathrm{GNN}(\mathbf{E} \| \mathbf{E}_{\text{nei}}, \mathbf{A}_{\text{nei}})$
        Calculate the predicted scores $\mathbf{s}, \bar{\mathbf{s}}$ using $\mathbf{E}, \bar{\mathbf{E}}$ via
          Equation (4).
        Calculate the ranking loss $\mathcal{L}_{\text{rank}}, \bar{\mathcal{L}}_{\text{rank}}$ using $\mathbf{s}, \bar{\mathbf{s}}, \mathbf{h}$
          via Equation (8).
        Calculate the asymmetric alignment loss $\mathcal{L}_{\text{align}}$ us-
          ing $\mathbf{E}, \bar{\mathbf{E}}$ via Equation (10).
        $\mathcal{L} \leftarrow \mathcal{L}_{\text{rank}} + \bar{\mathcal{L}}_{\text{rank}} + \alpha\mathcal{L}_{\text{align}}$
        Back-propagate $\mathcal{L}$ to update the parameters of
          MLP and GNN with learning rate lr.
    end for
end for
// Inference phase.
$\mathbf{E} \leftarrow \mathrm{MLP}(\mathbf{X})$
Calculate the predicted scores $\mathbf{s}$ using $\mathbf{E}$ via Equation
  (4).
return $\mathbf{s}$
```

the 29th ACM international conference on information & knowledge management, 315–324.

Duan, J.; Wang, S.; Zhang, P.; Zhu, E.; Hu, J.; Jin, H.; Liu, Y.; and Dong, Z. 2023a. Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 7459–7467.

Duan, J.; Xiao, B.; Wang, S.; Zhou, H.; and Liu, X. 2023b. Arise: Graph anomaly detection on attributed networks via substructure awareness. *IEEE transactions on neural networks and learning systems*.

Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023a. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference 2023*, 1528–1538.

Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023b. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, 357–365.

He, J.; Xu, Q.; Jiang, Y.; Wang, Z.; and Huang, Q. 2024. ADA-GAD: Anomaly-Denoised Autoencoders for Graph Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8481–8489.

Jin, M.; Liu, Y.; Zheng, Y.; Chi, L.; Li, Y.-F.; and Pan, S. 2021. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 3122–3126.

Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2021a. Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*, 3168–3177.

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

Liu, Z.; Dou, Y.; Yu, P. S.; Deng, Y.; and Peng, H. 2020. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 1569–1572.

Motie, S.; and Raahemi, B. 2023. Financial fraud detection using graph neural networks: A systematic review. *Expert Systems With Applications*, 122156.

Pan, J.; Liu, Y.; Zheng, Y.; and Pan, S. 2023. PREM: A Simple Yet Effective Approach for Node-Level Graph Anomaly Detection. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1253–1258. IEEE.

Qiao, H.; and Pang, G. 2024. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. In *Advances in Neural Information Processing Systems*, volume 36.

Sehwag, V.; Chiang, M.; and Mittal, P. 2021. Ssd: A unified framework for self-supervised outlier detection. *arXiv preprint arXiv:2103.12051*.

Shi, F.; Cao, Y.; Shang, Y.; Zhou, Y.; Zhou, C.; and Wu, J. 2022. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM web conference 2022*, 1486–1494.

Tang, J.; Li, J.; Gao, Z.; and Li, J. 2022. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, 21076–21089. PMLR.

Wang, Y.; Zhang, J.; Huang, Z.; Li, W.; Feng, S.; Ma, Z.; Sun, Y.; Yu, D.; Dong, F.; Jin, J.; et al. 2023. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference 2023*, 406–416.

Xu, F.; Wang, N.; Wu, H.; Wen, X.; Zhao, X.; and Wan, H. 2024. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9214–9222.

Yu, S.; Ren, J.; Li, S.; Naseriparsa, M.; and Xia, F. 2022. Graph learning for fake review detection. *Frontiers in Artificial Intelligence*, 5: 922589.

Zhao, L.; and Akoglu, L. 2020. On Using Classification Datasets to Evaluate Graph Outlier Detection: Peculiar Observations and New Insights. *CoRR*, abs/2012.12931.

Zheng, X.; Wang, Y.; Liu, Y.; Li, M.; Zhang, M.; Jin, D.; Yu, P. S.; and Pan, S. 2022. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*.

Zheng, Y.; Jin, M.; Liu, Y.; Chi, L.; Phan, K. T.; and Chen, Y.-P. P. 2021. Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12220–12233.

Zhu, J.; Jin, J.; Loveland, D.; Schaub, M. T.; and Koutra, D. 2022. How does heterophily impact the robustness of graph neural networks? theoretical connections and practical implications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2637–2647.

Zhuo, W.; Liu, Z.; Hooi, B.; He, B.; Tan, G.; Fathony, R.; and Chen, J. 2024. Partitioning message passing for graph fraud detection. In *The Twelfth International Conference on Learning Representations*.