HCT: A Hierarchical Contrastive Learning Framework for Transferable Graph Anomaly Detection

Jiawei Ye[1], Hongyi Li[1], Qinlin Xie[1], Sicheng Liang[1], Yu Liu[1], and Jie Wu[1](✉)

Fudan University, Shanghai, 200433, China
jwye@fudan.edu.cn,{hyli24,xieqinlin00,scliang23,yuliu24}@m.fudan.edu.cn,
jwu@fudan.edu.cn

Abstract. Graph anomaly detection (GAD) aims to identify abnormal nodes that differ from the majority within a graph and has been widely applied in real-world applications, where solutions based on graph neural network (GNN) have recently achieved remarkable success. However, GNN struggles to adapt to variations in the underlying data distributions, limiting its practical applicability. Existing efforts either train separate models for each dataset, rely heavily on source data, or overlook graph heterogeneity in GAD tasks, leading to challenges in transferability and generality. Therefore, how to effectively establish the underlying normal patterns and enable anomaly detection across graphs with varying feature and structure distributions remains an under-explored problem. To tackle these challenges, this paper proposes HCT, a general GAD framework for cross-graph transfer learning. Specifically, we first introduce node-feature disparity-based ranking and feature mapping to align anomaly features across graphs. Moreover, we employ a hierarchical contrastive learning framework to capture and transfer anomaly patterns effectively. HCT extracts deep structure information from the source graph at the node, subgraph, and view levels while employing a lightweight, trainable network module in the target graph to minimize cross-graph structure differences via contrastive learning. Besides, we design a structure-enhanced regularization objective to improve model adaptation in label-scarce scenarios. Extensive experiments on four real-world datasets demonstrate the effectiveness of HCT against state-of-the-art baselines with 1.63%∼8.05% average performance improvement across both settings, showcasing its strong generality and adaptability.

Keywords: Graph Anomaly Detection · Transfer Learning · Contrastive Learning.

1 Introduction

Graph anomaly detection (GAD) aims at identifying abnormal nodes that show significant deviations from the majority of nodes in a graph. It has garnered considerable research attention due to its broad real-world applications, such

as fraud detection [1], spam review identification [2] and rumor detection [3]. Thanks to exceptional performance in handing high-dimensional features and complex interdependent relations on graphs, the Graph Neural Network (GNN) has recently been introduced into GAD with promising progress [4]. However, since graphs are non-Euclidean, with diverse structures and node attributes across different graphs, GNN-based GAD faces challenges when confronted with substantial variations in the underlying data distributions [5]. Hence, how to effectively model normal patterns and distinguish anomaly nodes in different graphs has become an urgent problem.

Existing GNN-based GAD explorations can be divided into supervised and unsupervised approaches. Supervised GAD methods detect anomaly node patterns through message passing/aggregation optimization [6, 1] or distribution correlation between graph and high frequency spectral [7, 8], assuming the availability of sufficient labeled data. In contrast, unsupervised GAD methods rely on non-label capture graph anomaly patterns through unsupervised learning techniques such as graph reconstruction [9] and contrastive learning [10, 11]. Unfortunately, existing mainstream solutions require training separate detection models for each dataset, leading to high training costs and challenges in adapting to new graphs, which might be impractical for large-scale real-world scenarios.

Recently, the pretrain-finetune paradigm has shown great potential in graph-based tasks with GNN [12–14]. It leverages unsupervised pre-training to inject generalizable graph knowledge into GNNs, which can then be fine-tuned for effective generalization across different graphs without training from scratch. However, current studies [15] focus on the neighborhood homophily assumption that a node and its neighborhood nodes share similar labels while graphs typically exhibit neighborhood heterogeneity in GAD tasks, which may degrade GAD performance.

How to effectively establish the underlying normal patterns and enable anomaly detection over different distribution graphs is an under-explored problem, which is non-trivial due to three main challenges: (1) Cross-graph Feature Alignment: Different graphs exhibit significant variations in semantic space and feature dimensions. Current methods [16, 17] rely on source graphs to provide signals, but in real-world scenarios, these signals may be inaccessible due to regulatory and privacy constraints. (2) Anomaly Pattern Learning: Existing transfer learning methods [12, 18] often neglect the detailed exploration of generic anomaly patterns. Moreover, the structure differences between graphs make it challenging to effectively mine and transfer these patterns across diverse graphs. (3) Graph Label Scarcity: GNN-based GAD typically focuses on single-dataset settings, achieving outstanding performance by relying on sufficient labels in the graphs, which are not always available in real-world scenarios.

To tackle these challenges, we present HCT, a novel general GAD framework based on hierarchical contrastive learning, which enables effective transfer across cross-graph domains. For cross-graph feature alignment, we introduce the node-feature disparity to align feature anomaly semantics and dimensions across different graphs, enabling transfer without reliance on source graph sig-

nals. For anomaly detection, hierarchical contrastive learning is employed to deeply mine and transfer anomaly information. During pre-training, we construct a multi-level contrastive learning network based on graph augmentation, capturing anomaly information at the node, subgraph, and view levels to enhance normal patterns modeling and anomaly patterns understanding. During fine-tuning, we leverage low-rank adaptation (LoRA) [19] due to its success in large language model adaptability to transfer anomaly detection by adding a lightweight and trainable network, while using contrastive learning to shorten structure differences in different graphs. Additionally, we propose a structure-enhanced regularization objective that exploits graph neighborhood heterogeneity to enhance the model's adaptability on graphs with scarce labels. Consequently, we find that HCT demonstrates strong detection performance compared to baselines, with 1.63%∼8.05% average performance improvement across public and 10-shot settings. Furthermore, it surpasses training-from-scratch methods over 10% absolute improvement on some datasets. Generally, the contributions are as follows:

- We introduce a novel general GAD framework HCT, which leverages node-feature disparity for feature alignment, enabling migration without relying on source graph signals. In addition, it employs a hierarchical contrastive strategy to capture deep anomaly patterns.
- We propose an efficient transfer strategy that employs LoRA for anomaly pattern transfer, with contrastive learning to reduce cross-graph structural differences, and incorporates structure-enhanced regularization to improve adaptability in label-scarce scenarios.
- Extensive experiments on four large-scale real-world datasets demonstrate the superiority of HCT over state-of-the-art methods, showing significant performance in generality and adaptability.

2 Related Work

Graph Anomaly Detection. In this paper, we focus on anomaly detection on undirected attributed graphs, where anomalies involve either feature differences from neighboring nodes or dissimilar nodes being tightly connected. With the significant improvements of GNNs in graph data mining, GNN-based GAD [20] has garnered widespread attention. Existing mainstream solutions train separate detection models for each dataset. Supervised GNNs utilize message passing and aggregation or graph-high frequency distribution correlation to uncover anomaly patterns, such as BWGNN [21] applies localized band-pass filters to manage higher frequency anomalies, while AMNet [8] captures both low-frequency and high-frequency signals. Unsupervised GNNs leverage contrastive learning [10, 11], graph reconstruction [9], or auxiliary objectives [22] to train models without any labeled data. In real-world large-scale graph data scenarios, training separate models for each dataset leads to high training costs and difficulties in quickly adapting to new domains. While some GAD methods [9, 16] attempt to

apply cross-domain transfer, their reliance on source graph limits their generalizability. Unlike existing methods, our proposed HCT enables fine-tuning on target datasets without needing joint fine-tuning with the source graph, allowing for rapid adaptation in data-scarce scenarios.

Graph Contrastive Learning (GCL) focuses on uncovering the inherent similarities and differences between objects in graphs, aiming to extract universal graph knowledge. Recently, GCL has emphasized mining graph information from different levels. GraphCL [23] leverages view augmentation strategies to enhance node representations by contrasting augmented subgraphs. GRADATE [24] further explores subgraph representation learning by designing cross-view contrastive losses to capture local features and structure information. Unlike previous works, we consider hierarchical anomaly feature mining, and innovatively introduce view-level contrastive learning during the pre-training phase.

Graph Transfer Learning aims to pre-train a GNN and apply it to various datasets. The pretrain-finetune paradigm [12, 25], which involves pre-training a GNN on a source graph and then fine-tuning it on a target graph, has attracted significant attention due to its ability to transfer knowledge without requiring direct relationships between the source and target graphs. For instance, GCC [15] focuses on pre-training to develop a more general GNN. Besides, GraphControl [18] and GraphLoRA [13] emphasize fine-tuning to adapt the pre-trained GNN to different graphs. Most relevant to our work is GraphLoRA, which freezes the pre-trained GNN and utilizes LoRA-based contrastive learning to facilitate knowledge transfer. In contrast to GraphLoRA, our approach further considers graph neighborhood heterogeneity during the fine-tuning stage and incorporates a structure-enhanced objective to improve adaptability for cross-graph anomaly detection.

3 Methodology

3.1 Problem Formulation

Notations. In the following section, we formalize the GAD task. For the input, the notation $G = (V, E)$ denotes the given undirected graph, where $V = \{v_1, v_2, \ldots, v_n\}$ represent the node set with n nodes and $E = \{(v_i, v_j)|v_i, v_j \in V\}$ is the edge set. In addition, the node feature matrix $X \in \mathbb{R}^{n \times d}$ represents the node attributes, where each node in V has a feature vector of d-dimensional attributes. The adjacency matrix $A \in \{0, 1\}^{n \times n}$ encodes the graph structure, where $A_{ij} = 1$ indicates the presence of an edge between nodes v_i and v_j. D denotes the degree matrix of A.

Graph Neural Networks. Major GNNs adopt message-passing networks, where neighboring nodes exchange and aggregate information to share and update node features, capturing both local relationships and global information in the graph. In this paper, we use GCN [26] as the basic module, where the hidden representation at the $\ell + 1$-th layer can be defined as:

$$h_i^{(\ell+1)} = \sigma\left(h_i^{(\ell)} W^{(\ell)}\right). \tag{1}$$

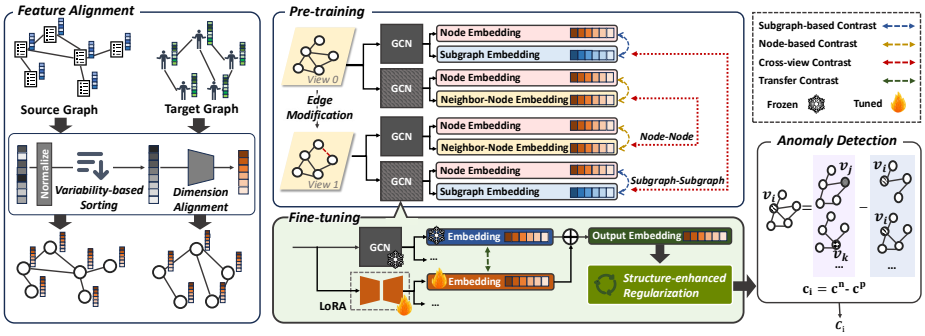

Fig. 1. The overall framework of HCT. Notably, networks under two views use the same architecture and share parameters.

$$H_i^{(\ell+1)} = \sigma\left(D_i^{-\frac{1}{2}} A_i D_i^{-\frac{1}{2}} H_i^{(\ell)} W^{(\ell)}\right). \tag{2}$$

where $h_i^{(\ell)}$ and $H_i^{(\ell)}$ represent the node hidden-layer representation and the subgraph hidden-layer representation, respectively. $\sigma(\cdot)$ is nonlinear transformation, $D_i^{-\frac{1}{2}} A_i D_i^{-\frac{1}{2}}$ indicates the normalization of the adjacency matrix, $W^{(\ell)}$ denotes the network parameters.

Problem Statement. The GAD model aims to learn an anomaly scoring function $f : G \to C$ that differentiates abnormal nodes V_a from normal nodes V_n within a given graph G, where V_a and V_n satisfy $V_a \cup V_n = V$, $V_a \cap V_n = \emptyset$. C is the anomaly score, with higher values indicating a higher likelihood of anomaly. In this paper, we focus on transferable GAD, which leverages anomaly knowledge from a pre-trained model on the source graph G_s and applies it to the target graph G_t with different data distributions. Assuming the source graph is unlabeled and the target graph has limited labeled nodes, HCT aims to transfer the pre-trained model from G_s and fine-tune it on G_t for anomaly detection. The optimization objective using target training nodes can be expressed as follows:

$$f_\phi^* = \arg\min_\phi \mathcal{L}\left(f_\phi\left(X_t, A_t\right), Y_t\right). \tag{3}$$

where $\mathcal{L}$ is the fine-tuning loss function, X_t and A_t represent the node feature matrix and adjacency matrix in G_t, and Y_t denotes training labels available for G_t. The function $f_\phi(\cdot) = p_\phi \circ g_\theta(\cdot)$, where p_ϕ is the tunable module and g_θ is the frozen pre-trained model.

3.2 HCT Overview

The overall framework of HCT is illustrated in Figure 1. First, we design a feature alignment module to map features between the source and target graphs. In this module, node-feature disparity is introduced to capture anomaly semantics, aligning the feature anomaly semantics and dimensions across different

graphs through anomaly-based ranking and weighted mapping, reducing the distribution discrepancy between the source and target graphs. Next, we propose a hierarchical graph contrastive network to train a pre-trained model on the source graph, which innovatively employs cross-view contrastive learning on node and subgraph to uncover more local anomaly information for detection. Subsequently, we introduce a structure-aware transfer learning strategy for transferring anomaly information to the target graph. Inspired by LoRA [19], we apply low-rank adaptation to the pre-trained contrastive learning network with contrastive learning to minimize the structure differences between the source and target graphs. In this process, structure-enhanced regularization leverages label and graph neighborhood heterogeneity to enhance adaptability in scenarios with limited labels on the target graph. Finally, we combine the various anomaly information to calculate the anomaly score for each node in the target graph.

3.3 Feature Alignment

The graph from different domains typically exhibits significant differences in features when performing anomaly detection. For example, in social reviews, features may include user profiles and comment content, while in financial transactions, features might represent customer transaction behaviors. Therefore, the primary task in GAD transfer learning is to align the features into a common feature space. Feature alignment generally includes two main parts: semantic and dimension alignment. Previous work [13] aligns features by designing specialized function to minimize difference in node feature distributions with the data requirement for both the source graph and the target graph. However, it is not always feasible when source graph data is unavailable. To this end, we introduce a discrepancy-based feature alignment module, which achieves feature alignment by abstracting anomaly semantics without requiring joint training with the source graph. It consists of two phases: discrepancy-based feature ranking that aligns anomaly semantics, and feature mapping that aligns dimensionality.

Discrepancy-based Feature Ranking. The goal of GAD is to identify anomaly nodes within the graph, which exhibit significant feature disparity compared to normal nodes. In GAD, high-frequency graph signals tend to play a more crucial role in detection [7, 27, 21], showing that features with greater disparity across nodes are more important for distinguishing anomaly patterns. Therefore, node-feature disparity is introduced to measure the importance of each feature for GAD. Given a graph G with a feature matrix X, the node-feature disparity of its features can be defined as:

$$dis_k(\mathcal{N}(X)) = \frac{1}{|E|} \sum_{(v_i,v_j)\in E} \left(\mathcal{N}(X_{ik}) - \mathcal{N}(X_{jk})\right)^2. \tag{4}$$

where $\mathcal{N}(\cdot)$ denotes the normalization. A larger dis_k indicates that the k-th feature exhibits greater variation between connected nodes, which suggests a stronger association with high-frequency graph signals.

To align feature anomaly semantics, we reorder the features based on node-feature disparity. Specifically, rather than joint training, we rank the features of all input graphs in descending order of disparity, thereby achieving anomaly semantic alignment.

Feature Mapping. To unify the feature dimensions across multiple graphs, combined with the varying importance of different features in GAD, we employ weighted feature projection to map features from different dimensions into a common feature space. For a given feature matrix X, we first normalize the features and then apply a fully connected layer for weighted mapping. The feature mapping is defined as follows:

$$Z = map(X) = \mathcal{N}(X) \cdot (w_m dis)^T. \tag{5}$$

where w_m represents the parameters of the mapping function.

3.4 Hierarchical Graph Contrastive Network

To effectively extract anomaly patterns, we employ a hierarchical contrastive learning network for unsupervised training on the source graph. Inspired by GRADATE [24], we first apply view augmentation through graph enhancement techniques. In each view, subgraphs are generated using random walks and paired with target nodes. Subsequently, node-level and subgraph-level contrastive learning are utilized to capture both global and local anomaly patterns. Throughout this process, cross-view subgraph-subgraph and novelly introduced node-node contrasts optimize the model's embeddings across views. A joint-balanced optimization objective is then introduced to guide the training process.

Graph Augmentation. Edge modification [28] is employed to perform view augmentation, helping the model uncover deeper semantic information. Specifically, given the source graph $G_s = (V_s, E_s)$ with edge set E_s including m edges, we construct a second view $\hat{G}_s = (V_s, \hat{E}_s)$ by randomly dropping $\frac{pm}{2}$ edges from the adjacency matrix and adding an equal number of edges, where p represents the proportion (with $p = 0.2$ in our experiments). This approach allows the model to learn more anomaly knowledge without depending on the specific structure of the graph, thus improving its generalization.

To improve scalability on large-scale graphs, we use a random walk with restart strategy, as proposed in previous work [15], to sample subgraphs and construct node pairs targeting specific nodes. Subgraphs G_i and G_i', sampled from the same central node, are considered positive pairs, while subgraphs from different central nodes are treated as negative pairs.

Node-level Anomaly Knowledge Learning. Node-level contrastive learning focuses on the relationships between nodes and their neighboring nodes within each view. In each view, the node representations from its own subgraphs' neighboring nodes form positive pairs, while those from neighboring nodes of subgraphs with different central nodes form negative pairs. As shown in Eq.(2), the GCN layer maps node information from the subgraph into the embedding space. To obtain the neighboring node representations, we employ an MLP to project

the node features into the same embedding space, resulting in the neighboring node representation $u_i = H_i^{(\ell+1)}[1,:]$. Subsequently, following Eq.(1), the target node representation e_i^0 is computed.

As anomaly nodes tend to have lower feature similarity with their neighboring nodes, we leverage a bilinear function to measure the node-level correlation between the target node and its neighboring nodes:

$$c_i^0 = f_b(u_i, e_i) = \sigma\left(u_i W_b e_i^\top\right). \tag{6}$$

where W_b represents the learnable parameter. Given the graph neighborhood heterogeneity, in positive pairs, the target node is expected to have a high correlation with its neighbors, resulting in c_i^0 approaching 1. In contrast, negative pairs exhibit low correlation, causing c_i^0 to approach 0. Therefore, the node-level contrastive loss is calculated as follows:

$$\mathcal{L}_{\mathcal{N}} = -\sum_{i=1}^{n}\left(p_i \log c_i^0 + (1-p_i)\log(1-c_i^0)\right). \tag{7}$$

where p_i is equal to 0 for positive pairs and 1 for negative pairs.

Correspondingly, the node-level correlation in the another view, denoted as $\hat{c}_i^0$, and the node-level contrastive loss $\hat{\mathcal{L}}_{\mathcal{N}}$ can be computed analogously.

Subgraph-level Anomaly Knowledge Learning. Importantly, a new GCN layer operates independently at the subgraph level from the GCN at the node level and does not share the weight parameters. The Readout function is then applied to aggregate the node features within the subgraph G_i, computing its representation as follows:

$$s_i = \text{Readout}(Z_i) = \frac{1}{n_i}\sum_{j=1}^{n_i}(Z_i)_j. \tag{8}$$

where Z_i represents the feature representations of all nodes in G_i, n_i is the number of nodes in G_i, and $(Z_i)_j$ is the feature of the j-th node in G_i.

Using an MLP to map the node features into the same embedding space as the subgraph representation, we obtain the target node representation e_i^1, as defined in Eq.(1). Similarly, the subgraph-level correlation c_i^1 between the target node representation e_i^1 and the subgraph representation z_i can be calculated from the bilinear function. The optimization of subgraph-level contrast is as follows:

$$\mathcal{L}_{\mathcal{S}} = -\sum_{i=1}^{n}\left(p_i \log(c_i^1) + (1-p_i)\log(1-c_i^1)\right). \tag{9}$$

Likewise, subgraph-level correlation $\hat{c}_i^1$ and loss $\hat{\mathcal{L}}_{\mathcal{S}}$ can also be computed for another view.

View-level Anomaly Knowledge Learning. Building on the advantages of graph augmentation techniques, view-level contrastive learning considers node-node and subgraph-subgraph contrastive learning across different views.

For cross-view node-node contrast, the node forms a positive pair with the neighboring node representations from its own subgraph in another view and forms a negative pair with neighboring node representations from subgraphs centered around different nodes in the two views. Based on prior work [29], we design the following loss function:

$$\mathcal{L}_{\mathcal{NN}} = -\sum_{i=1}^{n} \log \left(\frac{\exp(u_i \cdot \hat{u}_i)}{\exp(u_i \cdot u_j) + \exp(u_i \cdot \hat{u}_j)} \right). \tag{10}$$

where u_i and $\hat{u}_i$ are the neighboring node representations of node v_i in the two views, while u_j and $\hat{u}_j$ are those of another node v_j in the two views.

For cross-view subgraph-level contrast, a target node v_i forms positive pairs with its own subgraph in another view, and negative pairs with subgraphs centered around different nodes in both views. The loss function is:

$$\mathcal{L}_{\mathcal{SS}} = -\sum_{i=1}^{n} \log \left(\frac{\exp(z_i \cdot \hat{z}_i)}{\exp(z_i \cdot z_j) + \exp(z_i \cdot \hat{z}_j)} \right). \tag{11}$$

where z_i and $\hat{z}_i$ represent the subgraph representations of node v_i in the two views, while z_j and $\hat{z}_j$ represent those of another node v_j in the two views.

Joint-balanced Optimization. During the pre-training phase, we propose a joint-balanced optimization objective to integrate information from different contrastive learning. To effectively balance node-level, subgraph-level, and view-level information, we introduce trade-off parameters that facilitate this process:

$$\begin{aligned} \mathcal{L}'_{\mathcal{N}} &= \alpha \mathcal{L}_{\mathcal{N}} + (1-\alpha)\hat{\mathcal{L}}_{\mathcal{N}} \\ \mathcal{L}'_{\mathcal{S}} &= \alpha \mathcal{L}_{\mathcal{S}} + (1-\alpha)\hat{\mathcal{L}}_{\mathcal{S}} \\ \mathcal{L}_{\mathcal{CR}} &= \beta \mathcal{L}_{\mathcal{NN}} + (1-\beta)\mathcal{L}_{\mathcal{SS}}. \end{aligned} \tag{12}$$

where $\alpha \in (0,1)$ is used to balance the two views, and $\beta \in (0,1)$ is used to balance node and subgraph representations.

To leverage the advantages of hierarchical contrast, the overall joint objective function during pre-training is defined as follows:

$$\mathcal{L}_{\text{pretrain}} = \beta \mathcal{L}'_{\mathcal{N}} + (1-\beta)\mathcal{L}'_{\mathcal{S}} + \mathcal{L}_{\mathcal{CR}}. \tag{13}$$

Through the above steps, we obtain a pre-trained GAD model g_θ via unsupervised learning on the source graph G_s.

3.5 Structure-Aware Transfer Learning

During fine-tuning, structure differences between the source and target graphs hinder the transferability of pre-trained GAD models. To bridge this gap, we propose a structure-aware transfer learning strategy which comprises two key components: (1) LoRA-based fine-tuning, which alleviates structural differences between graphs, and (2) structure-enhanced regularization, which exploits graph neighborhood heterogeneity to enhance adaptation in label-scarce scenarios.

LoRA-based Fine-tuning. During fine-tuning, we freeze the weights of the pre-trained model g_θ while adding a lightweight, trainable GCN layer with the same architecture to capture structure information from the target graph. This setup allows the pre-trained model to retain structural knowledge from the source graph while the newly added module effectively integrates structural patterns from the target graph. Moreover, LoRA significantly reduces the number of parameters updated during fine-tuning, mitigating potential issues such as overfitting and catastrophic forgetting.

For each GCN layer at the node and subgraph levels with weight matrix W, LoRA introduces an additional GCN layer with parameter matrix ΔW. The hidden representation at $l+1$-th layer is defined as follows:

$$h_i^{(\ell+1)} = \sigma\left(h_i^{(\ell)} W^{(\ell)}\right) + \sigma'\left(h_i^{(\ell)} \Delta W^{(\ell)}\right). \tag{14}$$

where $\Delta W^{(\ell)} = W_B^l W_A^l$, $\sigma\prime$ represents add nonlinear transformation. $W_B^l \in \mathbb{R}^{d_l \times r}$, $W_A^l \in \mathbb{R}^{r \times d_{l+1}}$, and the rank $r \ll \min(d_l, d_{l+1})$.

To enhance the transfer of graph-structured knowledge, we incorporate contrastive learning into each newly added GCN layer. Specifically, the representation h_i of the same node in the original GCN, and its counterpart h_i' in the newly added GCN layer are treated as a positive pair, while representations of different nodes across the two GCN layers are considered negative pairs. The fine-tuning contrastive loss is thus defined as:

$$\mathcal{L} = -\sum_{i=1}^{n} \log\left(\frac{\exp(h_i \cdot h_i')}{\exp(h_i \cdot h_j) + \exp(h_i \cdot h_j')}\right). \tag{15}$$

Based on the above equations, we can obtain the contrastive losses $\mathcal{L}_{\mathcal{CLN}}$, $\hat{\mathcal{L}}_{\mathcal{CLN}}$, $\mathcal{L}_{\mathcal{CLS}}$ and $\hat{\mathcal{L}}_{\mathcal{CLS}}$ for node-level and subgraph-level fine-tuning in both views. Considering the integrated optimization of multi-level contrastive learning information, the fine-tuning contrastive loss is as follows:

$$\begin{aligned} \mathcal{L}'_{\mathcal{CLN}} &= \alpha \mathcal{L}_{\mathcal{CLN}} + (1-\alpha)\hat{\mathcal{L}}_{\mathcal{CLN}} \\ \mathcal{L}'_{\mathcal{CLS}} &= \alpha \mathcal{L}_{\mathcal{CLS}} + (1-\alpha)\hat{\mathcal{L}}_{\mathcal{CLS}} \\ \mathcal{L}_{\mathcal{CL}} &= \beta \mathcal{L}'_{\mathcal{CLN}} + (1-\beta)\mathcal{L}'_{\mathcal{CLS}}. \end{aligned} \tag{16}$$

where α and β take the same values as optimization parameters in pre-training.

Structure-enhanced Regularization. In GAD, anomaly nodes exhibit dissimilarity with their neighboring node features, whereas normal nodes are more similar to their neighbors. To this end, we leverage the principle of structure heterogeneity to enhance transferability in scenarios with limited labels.

Building on the correlation calculation in Eq.(6), we introduce a structure-enhanced regularization objective. Normal nodes demonstrate high correlation with their neighbors/subgraphs, while anomaly nodes show low correlation. The

regularization objectives at the node-level and subgraph-level are as follows:

$$\begin{aligned}\mathcal{L}_{RN} &= -\sum_i y_i \log(1-c_i^0) + (1-y_i)\log(c_i^0) \\ \mathcal{L}_{RS} &= -\sum_i y_i \log(1-c_i^1) + (1-y_i)\log(c_i^1).\end{aligned} \tag{17}$$

Here, y represents the label information in the target graph. The regularization objectives for the other view can be defined as $\hat{\mathcal{L}}_{RN}$ and $\hat{\mathcal{L}}_{RS}$. Similar to Eq.(16), we introduce the trade-off parameter to derive the final structure-enhanced optimization objective $\mathcal{L}_R$.

Fine-tuning Objective Optimization. During the fine-tuning phase, we employ multi-task learning to jointly optimize multiple objective functions. The overall objective function is defined as follows:

$$\mathcal{L}_{\text{finetune}} = \lambda_1 \mathcal{L}_{\mathcal{R}} + \lambda_2 \mathcal{L}_{\mathcal{CL}} + \lambda_3 \mathcal{L}_{\mathcal{CR}}. \tag{18}$$

where λ_i represents the importance of each objective function, which is set to 1 in our experiments.

3.6 Anomaly Detection

In anomaly detection, normal nodes exhibit high similarity with their own subgraph and neighbor node representations, while showing low similarity with the subgraph and neighboring node representations of other nodes. On the other hand, anomaly nodes are dissimilar to both their own subgraph and the subgraphs and neighboring nodes of other nodes. Thus, we define the anomaly score using the correlation as follows:

$$c_i = c^n - c^p. \tag{19}$$

where c^n represents the correlation in negative pair and c^p represents the correlation in positive pair. Leveraging the trade-off parameter in Eq.(12), we integrate node-level, subgraph-level, and view-level anomaly information, with the anomaly score further represented as:

$$\begin{aligned}c_i^{node} &= \alpha c_i^0 + (1-\alpha)\hat{c}_i^0 \\ c_i^{sub} &= \alpha c_i^1 + (1-\alpha)\hat{c}_i^1 \\ C_i &= \beta c_i^{node} + (1-\beta) c_i^{sub}.\end{aligned} \tag{20}$$

As a single-round detection may not always capture the relevant semantics, we perform multi-round anomaly detection and compute the average across these rounds as the final detection result.

Table 1. Statistics of datasets including the number of nodes and edges, the node feature dimension, the ratio of anomaly nodes in graph.

	Nodes	**Edges**	**Features**	**Anomaly**
Questions	48,921	153,540	301	3.00%
T-Finance	39,357	21,222,543	10	4.60%
Weibo	8,405	407,963	400	10.30%
Reddit	10,984	168,016	64	3.30%
Tolokers	11,758	519,000	10	21.80%

4 Experimental Evaluation

4.1 Experiments Settings

Datasets. For pre-training, we use the Questions [30] dataset which focuses on social media as source graph. For comprehensive evaluations, we consider four large-scale real-world datasets as target graphs that span a variety of domains, including finance (T-Finance [21]), crowd-sourcing (Toloker [30]), and social media (Weibo, Reddit) [31]. The statistics of datasets are provided in Table 1.

Baselines. We compare HCT with eight methods. For training-from-scratch methods, we choose two conventional GNNs, GIN [32] and GraphSAGE [33], along with AMNET [8] and BWGNN [21], which are specifically designed for the GAD task. For cross-graph transfer, we include three SOTA methods GCC [15]+fine-tuning, GraphControl [18], GraphLoRA [13], and the GAD-specific baselines ARC [5]. Notably, we add a classifier to GCC during fine-tuning, as it is originally an unsupervised contrastive learning model.

Metrics and Evaluation. We introduce two main metrics that match those of previous empirical studies [4, 5], including the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Area Under the Prevision Recall Curve (AUPRC). For all metrics, anomalies are considered as the positive class, and higher scores indicate better model performance. Experiments are conducted in two distinct settings: public and 10-shot. The public setting assumes sufficient labels, with 10% of the target graph dataset randomly sampled for training. The 10-shot setting represents a low-label scenario, where only 10 labeled instances per class are available in the target graph. In both cases, 80% of the target graph dataset is used for testing. For all methods, we report the average AUROC/AUPRC with standard deviations over 5 trials.

Settings. In HCT, both GCN consist of a single layer with ReLU activation. The subgraph size is fixed at 4, and both node and subgraph features are projected into a 64-dimensional hidden space. The model is trained for up to 400 epochs, followed by 100 rounds anomaly score calculation. Our implementation builds on prior work [24], with all experiments conducted on a single A800 GPU.

Table 2. Comparison of GAD performance in AUROC (%, mean±std), where highlighted results indicate the first and second rankings. OM indicates 'Out of Memory' in our experimental settings.

	T-Finance		Reddit		Weibo		tolokers	
Model	Public	10-shot	Public	10-shot	Public	10-shot	Public	10-shot
GIN	76.70±9.26	69.91±4.14	53.87±0.88	52.71±2.76	82.83±2.78	63.99±2.90	52.85±0.43	55.51±1.41
GraphSAGE	57.61±6.67	59.93±5.05	46.69±1.98	44.58±2.14	21.01±5.61	16.01±3.87	58.88±0.93	54.49±2.87
AMNet	83.38±1.82	80.51±3.95	50.68±0.21	57.56±0.54	80.52±1.51	71.50±2.79	59.31±0.35	53.01±1.96
BWGNN	83.57±2.84	79.92±4.31	50.90±2.83	55.58±3.90	67.37±2.70	59.68±1.31	60.25±0.18	56.03±2.65
GCC+finetuning	50.34±0.26	47.13±9.34	50.02±0.07	51.04±2.61	84.05±3.86	77.91±5.25	50.97±2.18	49.96±0.25
GraphControl	85.71±1.59	71.27±0.58	50.06±0.13	54.45±1.17	90.05±0.56	75.63±2.69	60.18±2.47	53.92±2.37
GraphLoRA	OM	OM	50.00±2.18	54.18±0.64	77.30±2.95	69.51±0.68	50.36±0.43	49.58±0.01
ARC	75.25±0.69	68.84±4.75	**58.90±0.29**	**57.58±1.99**	88.45±0.30	77.43±0.21	48.31±0.75	49.64±2.96
HCT	**88.99±0.23**	**81.17±3.01**	55.39±0.49	54.99±1.12	**91.63±0.30**	**80.65±3.07**	**61.61±0.73**	**59.81±0.69**

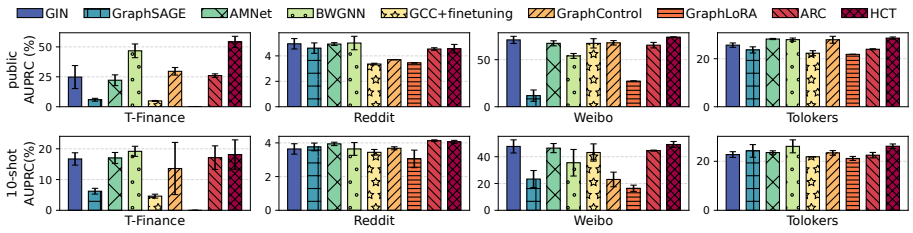

Fig. 2. GAD performance in terms of AUPRC.

4.2 Main Results

We evaluate the GAD performance by comparing HCT with eight baselines. Table 2 shows the comparsion results of AUROC in both settings, while Figure 2 illustrates the AUPRC comparison. The observations are as follows:

Overall Performance. HCT demonstrates strong anomaly detection capability in the transferable GAD scenario across various datasets. Specifically, HCT achieves state-of-the-art results on three out of four datasets and approaches the optimal performance on the remaining one. Compared to the best-performing baseline, GraphControl, HCT improves AUROC by 2.91% and AUPRC by 8.05% in the public setting. In the 10-shot setting, it surpasses the strongest baseline, AMNet, by 3.51% in AUROC and 1.63% in AUPRC.

Effectiveness of Cross-graph Transferability. HCT presents robust stability in cross-domain transfer, even when dealing with disparate graph domains. Compared to training-from-scratch methods, HCT leverages anomaly knowledge from the source graph to enhance performance on the target graph, achieving over a 10% AUROC improvement on the Reddit dataset. Moreover, we observe that transfer learning baselines show minimal improvements or even adverse effects when fine-tuned on specific datasets. In contrast, HCT not only enhances performance on datasets from similar domains but also improves AUROC by 1.85% in the public setting and 7.89% in the 10-shot setting on the cross-domain T-finance and Tolokers datasets, surpassing other transfer learning methods.

Table 3. Ablation study with AUROC(%, mean±std) in public and 10-shot settings.

	T-Finance		Reddit		Weibo		tolokers	
Model	Public	10-shot	Public	10-shot	Public	10-shot	Public	10-shot
HCT	**88.99**±0.23	**81.17**±3.01	**55.39**±0.49	**54.99**±1.12	**91.63**±±0.03	**80.65**±3.07	**61.61**±0.73	**59.81**±0.69
w/o dfs	85.16±0.12	65.19±2.01	52.75±0.99	49.59±4.30	43.29±8.90	78.32±0.72	60.08±1.32	49.07±3.76
w/o view	83.20±0.39	59.26±8.66	51.21±0.44	50.74±1.75	62.72±4.78	65.15±3.79	59.78±0.44	59.23±0.87
w/o aug	80.01±0.10	28.27±6.48	53.39±0.68	50.58±1.60	14.16±0.88	66.01±8.57	61.02±0.32	59.20±0.94
w/ node	54.51±0.36	25.60±10.39	53.88±0.79	49.79±3.30	16.12±0.42	62.80±6.96	61.46±0.28	58.72±1.36
w/ subgraph	83.19±0.17	37.35±9.35	52.17±0.80	51.12±3.71	22.78±8.90	66.53±4.56	59.71±0.11	59.08±0.83
w/ finetuning	88.23±2.75	81.05±2.04	55.04±0.36	50.09±0.82	89.40±0.65	72.62±9.15	60.49±0.10	59.69±2.02
w/o ser	60.99±5.21	53.36±2.51	43.75±1.48	46.89±5.33	66.24±1.29	65.02±5.16	56.12±1.42	49.89±3.97

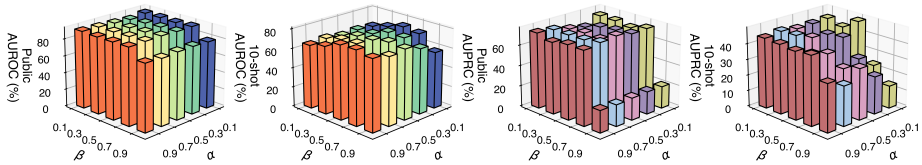

Fig. 3. Sensitivity analysis for the trade-off parameters α and β on Weibo.

Effectiveness of Heterogeneity Consideration. It is curial to consider heterogeneity in GAD. Compared to GraphLoRA, which assumes homogeneity, HCT demonstrates an average improvement of 10.32% in AUROC and 18.21% in AUPRC in the public setting, and 7.36% in AUROC and 12.69% in AUPRC in the few-shot setting, all under the same LoRA-based fine-tuning conditions. Moreover, general transfer learning methods based on homogeneity assumption are less effective than specialized approaches that account for graph neighborhood heterogeneity in GAD, which is evidenced by the strong AUROC and AUPRC performance of AMNet, BWGNN, and ARC.

4.3 Further Validation and Analysis

Ablation Studies. We evaluate the importance of our purposed modules in HCT, including feature alignment, hierarchical contrastive learning, and structure-enhanced objectives. To this end, several variants are designed: For feature alignment, **w/o dfs** replaces feature alignment with a simple dimensional mapping module, without ranking and weighting mechanism based on node-feature disparity. For hierarchical constrastive learning, **w/ node** and **w/ subgraph** denote using only node-level or subgraph-level contrast, respectively. **w/o aug** represents the performance without graph augmentation, and **w/o view** indicates the exclusion of view-level contrast. To evaluate LoRA-based contrast effectiveness, **w/ finetuning** refers to full fine-tuning on the training dataset for comparison. For objective optimization, **w/o ser** evaluates the impact of removing structure-aware regularization. The results, as shown in Table 3, show that the fully equipped HCT consistently achieves the best performance, thus demonstrating the effectiveness of each component.

Convergence Analysis. We discuss the important trade-off parameters, α and β, involved in our methods. As shown in Figure 3, these parameters effectively enhance GAD performance on the Weibo dataset, demonstrating the effectiveness of view balancing and node-subgraph balancing. Notably, we observe a significant drop in AUROC and AUPRC when $\beta = 0.9$, indicating that subgraph-level information plays a crucial role in capturing graph anomalies.

5 Conclusion

In this paper, we investigate the challenge of cross-graph anomaly detection in GNNs. To address the differences in feature and structure distributions across graphs, we introduce HCT, a novel general GAD framework that enables effective cross-graph transfer on undirected attributed graphs. To achieve this, HCT integrates a disparity-based mapping mechanism for cross-graph feature alignment alongside hierarchical contrastive learning to facilitate anomaly pattern capture and transfer. Additionally, a structure-aware regularization objective is proposed to enhance adaptability in label-scarce scenarios. Extensive experiments on four large-scale real-world datasets confirm the effectiveness and generalizability of HCT, significantly outperforming baselines while maintaining stable transferability across disparate graph domains. In the future, we will continue to explore the graph heterogeneity and efficient detection on large-scale graphs for the task.

Acknowledgments. This study was funded by Joint Innovation Initiative of the Yangtze River Delta Science and Technology Innovation Community (Jiangsu, Zhejiang, Shanghai)(grant number YDZX20223100004022-3).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Duan, M., Zheng, T., Gao, Y., Wang, G., Feng, Z., Wang, X.: Dga-gnn: Dynamic grouping aggregation gnn for fraud detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 11820–11828 (2024)
2. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: Proceedings of the 22nd international conference on World Wide Web. pp. 897–908 (2013)
3. Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., Huang, J.: Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 549–556 (2020)
4. Tang, J., Hua, F., Gao, Z., Zhao, P., Li, J.: Gadbench: Revisiting and benchmarking supervised graph anomaly detection. Advances in Neural Information Processing Systems **36**, 29628–29653 (2023)
5. Liu, Y., Li, S., Zheng, Y., Chen, Q., Zhang, C., Pan, S.: Arc: A generalist graph anomaly detector with in-context learning. In: Advances in Neural Information Processing Systems (2024)

6. Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: Current limitations and effective designs. Advances in neural information processing systems **33**, 7793–7804 (2020)
7. Gao, Y., Wang, X., He, X., Liu, Z., Feng, H., Zhang, Y.: Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In: Proceedings of the ACM web conference 2023. pp. 1528–1538 (2023)
8. Chai, Z., You, S., Yang, Y., Pu, S., Xu, J., Cai, H., Jiang, W.: Can abnormality be detected by graph neural networks? In: IJCAI. pp. 1945–1951 (2022)
9. Ding, K., Li, J., Bhanushali, R., Liu, H.: Deep anomaly detection on attributed networks. In: Proceedings of the 2019 SIAM international conference on data mining. pp. 594–602. SIAM (2019)
10. Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K.T., Chen, Y.P.P.: Generative and contrastive self-supervised learning for graph anomaly detection. IEEE Transactions on Knowledge and Data Engineering **35**(12), 12220–12233 (2021)
11. Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., Karypis, G.: Anomaly detection on attributed networks via contrastive self-supervised learning. IEEE transactions on neural networks and learning systems **33**(6), 2378–2392 (2021)
12. Zhao, H., Chen, A., Sun, X., Cheng, H., Li, J.: All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 4443–4454 (2024)
13. Yang, Z.R., Han, J., Wang, C.D., Liu, H.: Graphlora: Structure-aware contrastive low-rank adaptation for cross-graph transfer learning. arXiv preprint arXiv:2409.16670 (2024)
14. Gui, A., Ye, J., Xiao, H.: G-adapter: Towards structure-aware parameter-efficient transfer learning for graph transformer networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 12226–12234 (2024)
15. Qiu, J., Chen, Q., Dong, Y., Zhang, J., Yang, H., Ding, M., Wang, K., Tang, J.: Gcc: Graph contrastive coding for graph neural network pre-training. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1150–1160 (2020)
16. Qiao, H., Pang, G.: Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. Advances in Neural Information Processing Systems **36**, 49490–49512 (2023)
17. Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., Karypis, G.: Anomaly detection on attributed networks via contrastive self-supervised learning. IEEE transactions on neural networks and learning systems **33**(6), 2378–2392 (2021)
18. Zhu, Y., Wang, Y., Shi, H., Zhang, Z., Jiao, D., Tang, S.: Graphcontrol: Adding conditional control to universal graph pre-trained models for graph domain transfer learning. In: Proceedings of the ACM Web Conference 2024. pp. 539–550 (2024)
19. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)
20. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A comprehensive survey on graph anomaly detection with deep learning. IEEE transactions on knowledge and data engineering **35**(12), 12012–12038 (2021)
21. Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: International conference on machine learning. pp. 21076–21089. PMLR (2022)
22. Huang, T., Pei, Y., Menkovski, V., Pechenizkiy, M.: Hop-count based self-supervised anomaly detection on attributed networks. In: Joint European con-

ference on machine learning and knowledge discovery in databases. pp. 225–241. Springer (2022)
23. Hafidi, H., Ghogho, M., Ciblat, P., Swami, A.: Negative sampling strategies for contrastive self-supervised learning of graph representations. Signal Processing **190**, 108310 (2022)
24. Duan, J., Wang, S., Zhang, P., Zhu, E., Hu, J., Jin, H., Liu, Y., Dong, Z.: Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 7459–7467 (2023)
25. Li, S., Han, X., Bai, J.: Adaptergnn: Parameter-efficient fine-tuning improves generalization in gnns. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 13600–13608 (2024)
26. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International conference on machine learning. pp. 6861–6871. Pmlr (2019)
27. Gao, Y., Wang, X., He, X., Liu, Z., Feng, H., Zhang, Y.: Alleviating structural distribution shift in graph anomaly detection. In: Proceedings of the sixteenth ACM international conference on web search and data mining. pp. 357–365 (2023)
28. Jin, M., Liu, Y., Zheng, Y., Chi, L., Li, Y.F., Pan, S.: Anemone: Graph anomaly detection with multi-scale contrastive learning. In: Proceedings of the 30th ACM international conference on information & knowledge management. pp. 3122–3126
29. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
30. Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., Prokhorenkova, L.: A critical look at the evaluation of gnns under heterophily: Are we really making progress? arXiv preprint arXiv:2302.11640 (2023)
31. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1269–1278 (2019)
32. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
33. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)