RESEARCH-ARTICLE

AnomalyGFM: Graph Foundation Model for Zero/Few-shot Anomaly Detection

HEZHE QIAO, Singapore Management University, Singapore City, Singapore

CHAOXI NIU, University of Technology Sydney, Sydney, NSW, Australia

LING CHEN, University of Technology Sydney, Sydney, NSW, Australia

GUANSONG PANG, Singapore Management University, Singapore City, Singapore

Published: 03 August 2025

Citation in BibTeX format

KDD '25: The 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 3 - 7, 2025
Toronto ON, Canada

Conference Sponsors:
SIGMOD
SIGKDD

KDD '25: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (August 2025)
https://doi.org/10.1145/3711896.3736843
ISBN: 9798400714542

AnomalyGFM: Graph Foundation Model for Zero/Few-shot Anomaly Detection

Hezhe Qiao†
School of Computing and Information Systems
Singapore Management University
Singapore
hezheqiao.2022@phdcs.smu.edu.sg

Chaoxi Niu†
Australian Artificial Intelligence Institute
University of Technology Sydney
Sydney, Australia
chaoxi.niu@student.uts.edu.au

Ling Chen
Australian Artificial Intelligence Institute
University of Technology Sydney
Sydney, Australia
ling.chen@uts.edu.au

Guansong Pang*
School of Computing and Information Systems
Singapore Management University
Singapore
gspang@smu.edu.sg

Abstract

Graph anomaly detection (GAD) aims to identify abnormal nodes that differ from the majority of the nodes in a graph, which has been attracting significant attention in recent years. Existing generalist graph models have achieved remarkable success in different graph tasks but struggle to generalize to the GAD task. This limitation arises from their difficulty in learning generalized knowledge for capturing the inherently infrequent, irregular and heterogeneous abnormality patterns in graphs from different domains. To address this challenge, we propose **AnomalyGFM**, a GAD-oriented graph foundation model that supports zero-shot inference and few-shot prompt tuning for GAD in diverse graph datasets. One key insight is that graph-agnostic representations for normal and abnormal classes are required to support effective zero/few-shot GAD across different graphs. Motivated by this, AnomalyGFM is pre-trained to align data-independent, learnable normal and abnormal class prototypes with node representation residuals (*i.e.*, representation deviation of a node from its neighbors). The residual features essentially project the node information into a unified feature space where we can effectively measure the abnormality of nodes from different graphs in a consistent way. This provides a driving force for the learning of graph-agnostic, discriminative prototypes for the normal and abnormal classes, which can be used to enable zero-shot GAD on new graphs, including very large-scale graphs. If there are few-shot labeled normal nodes available in the new graphs, AnomalyGFM can further support prompt tuning to leverage these nodes for better adaptation. Comprehensive experiments on 11 widely-used GAD datasets with real anomalies, covering social networks, finance networks, and co-review networks, demonstrate that AnomalyGFM significantly outperforms state-of-the-art competing methods under both zero- and few-shot GAD settings. Code is available at https://github.com/mala-lab/AnomalyGFM.

*Corresponding author: Guansong Pang, † indicates equal contribution.

KDD '25, Toronto, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736843

CCS Concepts

• **Computing methodologies** → **Machine learning**.

Keywords

Anomaly Detection, Graph Neural Networks, Graph Anomaly Detection, Graph Foundation Models

ACM Reference Format:
Hezhe Qiao†, Chaoxi Niu†, Ling Chen, and Guansong Pang. 2025. AnomalyGFM: Graph Foundation Model for Zero/Few-shot Anomaly Detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736843

1 Introduction

Graph anomaly detection (GAD) aims to identify anomalous nodes within a graph that deviate significantly from others. It has found broad applications in areas such as financial fraud detection, transaction analysis, and social networks [1, 21, 25, 28]. Most existing GAD methods, either supervised or unsupervised methods, follow a paradigm of training and inference on the same graph, making the trained models struggle to generalize to new/unseen graphs due to the distribution shift between the training and testing graphs [6, 19, 27]. Further, this type of methods can become inapplicable in applications where graph data is not accessible during training due to data privacy or other data access issues. Generalist/foundation models for graphs have achieved remarkable progress in tackling these challenges in general tasks such as node classification, graph classification, and link prediction [9, 15, 31, 41, 45], but they are difficult to generalize to the GAD task. The main reason is that the generalized knowledge learned in these generalist models through generic pre-training or tuning methods inclines to the prevalent regular patterns, which differ significantly from the inherently infrequent, irregular, and heterogeneous abnormality patterns across the graphs [16, 18, 28].

Very recently, there have been some efforts, such as ARC [18] and UNPrompt [23], dedicating to generalist models for GAD. Despite the effectiveness on some datasets, these methods fail to provide a universal GAD model that versatilely supports zero-shot inference and sample-efficient tuning in different application scenarios. For

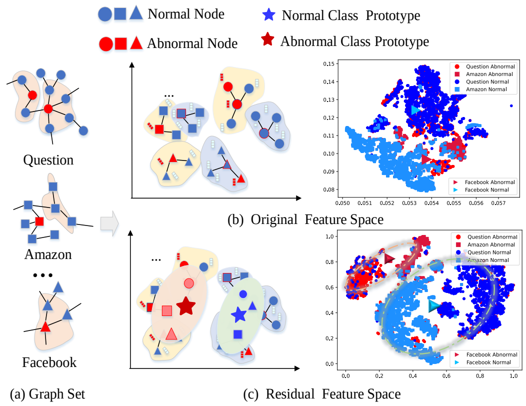

Figure 1: Given a set of graph datasets consisting of normal nodes (blue) and abnormal nodes (red) as in (a), AnomalyGFM aims to learn graph-agnostic prototypes for normal and abnormal nodes in a projected residual feature space. Compared to the original feature space where normal and abnormal patterns are irregularly distributed and differ significantly in different datasets as in (b), the residual feature space learned by AnomalyGFM in (c) is universally effective in separating anomalies from normal nodes in different graphs. In (b) and (c), AnomalyGFM is trained on Facebook [40] and evaluated directly on Amazon [7] and Question [26]; both feature spaces are learned by AnomalyGFM.

instance, UNPrompt [23] is designed exclusively for zero-shot inference, while ARC [18] is limited to few-shot tuning-based GAD within the same domain. Therefore, a foundational model being capable of effectively capturing abnormalities across graphs from different domains and supporting both zero-shot and few-shot GAD is still lacking.

To fill this gap, in this paper, we introduce **AnomalyGFM**, a GAD-oriented graph foundation model (GFM) that is capable of addressing both zero-shot and few-shot scenarios effectively for GAD. One key insight is that to avoid overfitting the pre-training graph data, graph-agnostic representations for normal and abnormal classes are required to support effective zero/few-shot GAD across different graphs. Motivated by this, AnomalyGFM is pre-trained to align data-independent, learnable normal and abnormal class prototypes with *node representation residuals* (*i.e.*, representation deviation of a node from its neighboring nodes). Compared to the original feature space where normal and abnormal patterns are irregularly and heterogeneously distributed in different datasets (see Fig. 1 (b)), the residual features essentially project the node information into a unified feature space where we can effectively measure the abnormality of nodes from different graphs in a consistently identical way, as shown in Fig. 1 (c). This is because normal nodes, regardless of which graphs they are from, are expected to have small residual/deviation from its neighboring nodes, whereas such residuals for abnormal nodes are large [18, 27]. By learning to align such residual features with the learnable normal and abnormal class prototypes, AnomalyGFM distills the residual feature-based anomaly discriminability into the two graph-agnostic prototypes, facilitating strong generalization to GAD in new graphs without any further tuning/training. Furthermore, if there are few-shot labeled normal nodes available in the new graphs, AnomalyGFM can further support graph prompt tuning to leverage these nodes for better adaptation.

To summarize, the contributions of AnomalyGFM are three-fold:

- We propose a new paradigm for GAD, aiming to pre-train a GAD-oriented GFM that achieves strong zero-shot generalization and supports sample-efficient graph prompt tuning for the GAD task.
- We argue that learning graph-agnostic representations for normal and abnormal classes is a key ingredient of a GAD-oriented GFM. To justify this, we propose AnomalyGFM, the very first GAD-oriented GFM with strong zero-shot and few-shot generalization abilities. It is pre-trained to learn discriminative, graph-agnostic class prototypes with normal and abnormal residual features, and it supports few-shot graph prompt tuning for better adaptation.
- A comprehensive benchmark on both zero-shot and few-shot settings using 11 real-world GAD datasets is established, on which i) AnomalyGFM performs significantly better than the state-of-the-art unsupervised, supervised, and generalist GAD methods, and ii) AnomalyGFM can scale up to very large graphs.

2 Related Work

2.1 Graph Anomaly Detection

Existing GAD methods can be generally classified into unsupervised and supervised methods [21, 28], depending on the assumption regarding the presence of normal and anomaly labels. The fully supervised approaches have recently achieved remarkable progress by leveraging both labeled normal and abnormal nodes [11, 17, 33, 35]. These methods are primarily designed to improve message aggregation in graph neural networks (GNNs) by reducing heterophilic edges from both spectral and spatial perspectives, effectively mitigating the over-smoothing problem in GAD [28, 32]. Fully supervised methods are effective in GAD as they treat the task as an imbalanced classification problem. However, the requirement for both normal and abnormal labels significantly hinders their applications to scenarios where labeled nodes are difficult to obtain.

Unsupervised methods, which typically assume the absence of both normal and abnormal labels, have garnered significant attention due to their more practical setting assumption on data labels. They generally incorporate some conventional techniques such as reconstruction [6], one-class classification [27, 36, 43], contrastive learning [19, 24], and adversarial learning [5] into graph learning to capture the normal patterns within the graph and then assign an anomaly score for each node based on its deviation from the normal patterns. However, these methods still follow the paradigm of training and inference on the same graph, making them struggle to generalize to new/unseen graphs due to the distribution shift between training and testing set.

2.2 Foundation Models for GAD

Generalist models have recently achieved significant progress on non-graph data by leveraging large pre-trained models to facilitate generalized pattern recognition in diverse downstream tasks, such as generalist anomaly detection on images [42, 44]. However, applying these methods to graph data remains highly challenging due to the absence of such pre-trained models [20, 31, 38] on graph data. The main challenge in designing a GFM is capturing invariance across diverse graphs while mapping them into a shared representation space to enable positive transfer between training and inference [2, 10, 22]. Currently, most GFM models use prompt learning to enable the knowledge transfer across graphs for general graph tasks [9, 20]. However, they still struggle to generalize to the GAD task due to the inherently infrequent, irregular, and heterogeneous abnormality patterns in GAD datasets [18, 23, 28]. Therefore, some recent efforts attempt to devise the GFMs for GAD [18, 23]. ARC [18] is one of such methods, a fine-tuning GFMs for GAD, enabling a "one-for-all" GAD model through a fine-tuning mechanism. UNPrompt [23] is the first zero-shot generalist GAD method that learns generalized neighborhood prompts, allowing latent node attribute predictability to serve as a generalized anomaly measure. Different from ARC and UNPrompt, which address the GAD-oriented GFMs partially under few-shot or zero-shot settings, AnomalyGFM can effectively capture abnormalities across graphs from different domains while supporting both zero-shot inference and few-shot prompt fine-tuning/inference. This offers more versatile abilities for a GFM-based GAD approach.

3 The Proposed AnomalyGFM

3.1 Preliminaries

Notation and Conventional GAD. An attributed graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ represents the node set and $\mathcal{E}$ represents the edge set. The graph can also be denoted as $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times \tilde{d}}$ is the attribute of the nodes and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacent matrix of the graph with $\mathbf{A}_{ij} = 0$ representing there is no edge between node v_i and node v_j and $A_{ij} = 1$ otherwise. In GAD, we denote the normal node set as $\mathcal{V}_n$ and abnormal node set $\mathcal{V}_a$, corresponding to the anomaly labels $\mathbf{y} \in \{0, 1\}^N$ with ($y_i = 0$ denotes a normal node v_i and $y_i = 1$ denotes abnormal). Typically, the number of normal nodes is significantly larger than the abnormal nodes in the GAD datasets, *i.e.*, $|\mathcal{V}_n| \gg |\mathcal{V}_a|$. The conventional GAD is formulated as learning a scoring function $f : \mathcal{G} \to \mathbb{R}$ and making inference on the same $\mathcal{G}$, in which f is expected to assign every node an anomaly score such that $f(v) < f(v'), \forall v \in \mathcal{V}_n, v' \in \mathcal{V}_a$.

Zero-shot GAD. Different from the conventional GAD that trains and inferences on the same graph, zero-shot GAD aims to train a generalist model on $\mathcal{G}_{Train}$ and then apply the learned model to the new test graphs $\mathcal{G}_{Test}$ without any further tuning/retraining.

Few-shot GAD. In the few-shot GAD setting, the model is fine-tuned on $\mathcal{G}_{Test}$ with small labeled nodes and then applied to the rest of the unlabelled nodes in the test graph $\mathcal{G}_{Test}$. We denote the labeled node in the test graph as $\mathcal{V}_l^{Test}$ while the unlabeled nodes in the test graph are denoted as $\mathcal{V}_u^{Test} = \mathcal{V}^{Test} \setminus \mathcal{V}_l^{Test}$.

Feature Unification. Due to the feature dimension difference across the graphs, we need to align the node features/attributes into a shared feature space to ease the feature heterogeneity among the graphs. Following the previous studies [18, 23], we employ singular value decomposition (SVD), a simple, efficient dimensionality reduction technique that can approximately preserve the distance relationship of the data by projecting high-dimensional data into a low-dimensional subspace for unification, For each dataset $\mathbf{X}^{(i)}$ with dimension $d^{(i)}$, it can be formulated as:

$$\mathbf{X}^{(i)} \in \mathbb{R}^{N^{(i)}\times d^{(i)}} \xrightarrow[\text{Projection}]{\text{Feature}} \tilde{\mathbf{X}}^{(i)} \in \mathbb{R}^{N^{(i)}\times d'}, \tag{1}$$

where d' is the common dimensionalities and $N^{(i)}$ is the number of node for graph $\mathcal{G}_i$

GNN for Representation Learning. Due to its simplicity and effectiveness, GNN is employed to learn the representation of each graph which can be formulated as follows,

$$\mathbf{H}^{(\ell)} = \text{GNN}\left(\mathbf{A}, \mathbf{H}^{(\ell-1)}; \mathbf{W}^{(\ell)}\right), \tag{2}$$

where $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N\times h^{(l)}}$, $\mathbf{H}^{(\ell-1)} \in \mathbb{R}^{N\times h^{(l-1)}}$ are the representations of all N nodes in the (ℓ)-th layer and $(\ell-1)$-th layer, respectively, $h^{(l)}$ is the dimension of (ℓ)-th layer, $\mathbf{W}^{(\ell)}$ are the learnable parameters, and $\mathbf{H}^{(0)} = \tilde{\mathbf{X}}$. $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N\}$ is a set of representations of N nodes in the last GNN layer. In this paper, we adopt a 2-layer GCN to model the graph due to its efficient and simple architecture.

3.2 Approach Overview

AnomalyGFM aims to distill discriminative node residual features in a unified feature space into two graph-agnostic prototypes for the normal and abnormal classes, facilitating the establishment of a foundation model that supports effective zero/few-shot GAD across graphs. Specifically, AnomalyGFM consists of pre-training based on the prototype alignment, few-shot graph prompt fine-tuning, and zero-shot/few-shot inference, as shown in Fig. 2. In the pre-training step, we aim to align the data-independent and learnable normal class and abnormal class prototypes with the nodes' residual features to achieve the distillation of graph-agnostic discriminative class-level representations. In the few-shot graph prompt fine-tuning, the labeled normal nodes' residual features are utilized to guide a better adaptation of the normal class prototype. In the inference, we devise zero-shot inference on new graphs, including subgraph-based inference for very large-scale graphs, and few-shot inference, both of which compute anomaly scores by measuring the similarity between the residual features of nodes and the two class prototypes.

3.3 Pre-training via Prototype Alignment in a Unified Feature Space

Node Representation Residual. Node representation residual can be seen as the deviation at the representation level between the target node and its neighboring nodes. The deviation in the representation between the target node and its neighbors is expected to be small for normal nodes, regardless of the graph domain, while the deviation for abnormal nodes is relatively large. Therefore, the node representation residual offers a consistent method to effectively capture heterogeneous abnormalities across graphs. Assume

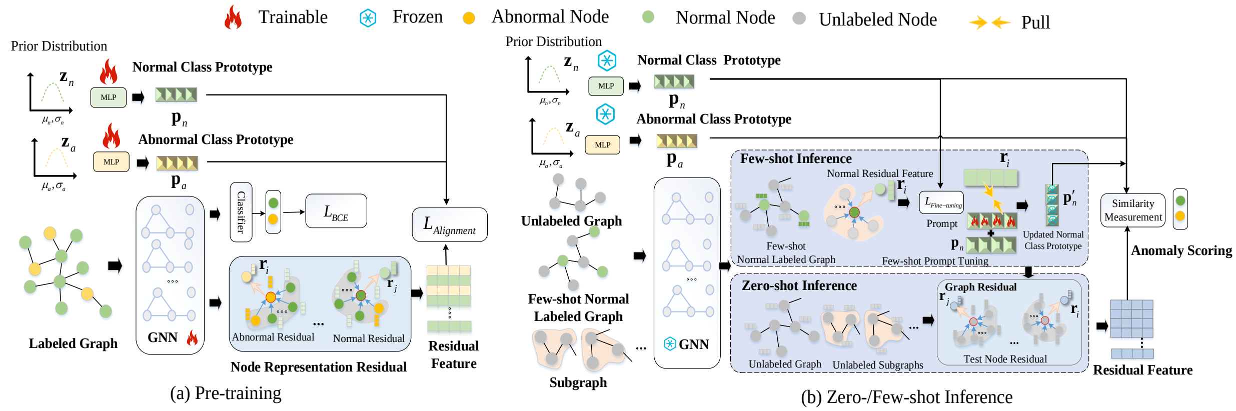

Figure 2: Overview of AnomalyGFM. (a) During the pre-training, AnomalyGFM aims to align two learnable class-level prototypes with the representation residuals of normal and abnormal nodes to learn discriminative graph-agnostic prototypes on an auxiliary labeled graph. (b) During inference, the anomaly score is determined in a consistent way using similarity between node representation residual and prototypes where all learnable parameters are frozen. Given a target graph with few-shot labeled normal nodes, these nodes are used to better adapt the pre-trained normal class prototypes to the target graph through a one-class prompt tuning method.

the node representation outputted by the GNNs as $\mathbf{h}_i \in \mathbb{R}^d$, where d is the dimension of representation, the residual feature $\mathbf{r}_i \in \mathbb{R}^d$ for the node v_i is formally defined as follows:

$$\mathbf{r}_i = \mathbf{h}_i - \frac{1}{|\mathcal{N}(v_i)|} \sum_{v_j \in \mathcal{N}(v_i)} \mathbf{h}_j, \tag{3}$$

where $\mathcal{N}(v_i)$ represents the neighbor set of node v_i. As a graph-agnostic representation, the node representation residual will be assigned to each node as its new feature in the unified feature space.

Graph-agnostic Prototype Alignment. To exploit the anomaly discriminability in node representation residuals, we learn two data-independent learnable prototypes for the two classes that align with the normal and abnormal residual features, respectively. The alignment essentially distills the residual feature-based anomaly discriminability into the two graph-agnostic prototypes, facilitating a strong generalization to GAD in new graphs. Formally, we integrate two learnable mapping functions $\mathbf{p}_n = \Phi(\mathbf{z}_n; \Theta_n)$, where $\mathbf{z}_n, \mathbf{p}_n \in \mathbb{R}^T$ and $\mathbf{p}_a = \Phi(\mathbf{z}_a; \Theta_a)$, where $\mathbf{z}_a, \mathbf{p}_a \in \mathbb{R}^T$ to map their initializations (*e.g.*, prototypes initialized with vectors drawn from a prior distribution such as Gaussian distribution) into the prototype space to serve as normal class prototype $\mathbf{p}_n$ and abnormal class prototype $\mathbf{p}_a$, respectively. Both Θ_n and Θ_a are the learnable parameters, and T is the dimension of the learnable prototype, which is set to match the dimensionality of the node representation d. The Gaussian prior distributions $\mathbf{z}_n$ and $\mathbf{z}_a$ are uniformly defined as $\mathcal{N}(\mu, \sigma)$ for the normal and abnormal classes, where μ is the mean of the distribution and σ is its covariance.

The pre-training of AnomalyGFM is performed by aligning the two learnable prototypes with the abnormal residual features and normal residual features respectively using an alignment loss function $L_{Alignment}$ defined as follows:

$$L_{Alignment} = \sum_{i=1}^{|\mathcal{V}|} I_{y_{i=0}} \|\mathbf{r}_i - \mathbf{p}_n\|_2^2 + I_{y_{i=1}} \|\mathbf{r}_i - \mathbf{p}_a\|_2^2, \tag{4}$$

where $I_{y_{i=0}}$ is an indicator function, which takes the value of 1 when the condition is satisfied. This distills the graph-agnostic discriminability in the residual feature space into the two prototypes. To enhance the anomaly discriminability in the node representation residual, we utilize a standard binary classification loss L_{BCE} to learn discriminative representations for the normal and abnormal nodes in the original node representation space:

$$L_{BCE} = \sum_{i=1}^{|\mathcal{V}|} y_i \log(p_i) + (1 - y_i) \log(1 - p_i), \tag{5}$$

where $p_i = f_\theta(\mathbf{h}_i)$ and $f_\theta(\cdot)$ is a Multi-Layer Perceptron (MLP) classifier with training parameters θ, which maps the representation to the probability for a node being an anomaly. Finally, we jointly optimize the $L_{Alignment}$ and L_{BCE} via the following total loss :

$$L_{total} = L_{BCE} + \alpha L_{Alignment}, \tag{6}$$

where α is a hyperparameter to control the importance of the proposed alignment module. Note that the pre-training is performed on auxiliary graph datasets rather than target graph datasets.

3.4 Few-shot Prompt Tuning

To enable the few-shot prompt tuning in the scenarios where few-shot labeled normal nodes are available in the test/target graph, we utilize these limited nodes to have a prompt tuning of AnomalyGFM in the prototype learning module while keeping the GNN parameters frozen.

Normal Prototype-based Prompting. To fully leverage the labeled normal nodes during the fine-tuning step, we introduce a small learnable prompt into the normal class prototype to better align it with the normal node representation residuals on the test graph during a prompt fine-tuning step. In addition, we also add an adaptation layer that offers more learning flexibility to the tuning. They are designed to refine the pre-trained normal class prototype $\mathbf{p}_n$ to a new normal prototype $\mathbf{p}'_n$, ensuring a better alignment with the normal class residual feature in the target graphs. Formally, the new normal prototype $\mathbf{p}'_{\mathbf{n}}$ is defined as:

$$\mathbf{p}'_{\mathbf{n}} = \mathbf{p}_n + g(\mathbf{p}_n; \phi) + \Psi_{\mathbf{p}_n}, \tag{7}$$

where $\Psi_{\mathbf{p}_n} \in \mathbb{R}^d$ is the learnable prompt while $g(\mathbf{p}_n; \phi)$ is the added adaptation layer with parameters ϕ, both of which are of the same dimensionality as $\mathbf{p}_n$. The learned prompts and the adapted prototype complements the pre-trained normal prototype in the target graph, so we fuse these three vectors to obtain the updated normal class prototype $\mathbf{p}'_{\mathbf{n}}$.

One-class Prompt Tuning. During the prompt tuning, the original learnable parameters of AnomalyGFM are all frozen. The few-shot labeled normal nodes are solely utilized to learn the new parameters ϕ and Ψ for refining the normal class prototype. Specifically, given some labeled normal node $\mathcal{V}_l^{Test}$ in the test graph, we first calculate the residual feature $\mathbf{r}_i$ of the labeled normal node $v_i \in \mathcal{V}_l^{Test}$ using Eq. 3. Then the graph prompt is optimized using the following one-class tuning loss L_{pt}, aiming to minimize the Euclidean distance between the residual features of labeled normal node $\mathbf{r}_i$ and the new trainable normal class prototype $\mathbf{p}'_{\mathbf{n}}$ composed by the prompt and the pre-trained normal prototype.

$$L_{pt} = \sum_{i=1}^{|\mathcal{V}_l^{Test}|} \left\| \mathbf{r}_i - \mathbf{p}'_{\mathbf{n}} \right\|_2^2, \tag{8}$$

where the residual feature $\mathbf{r}_i$ is fixed since the learnable parameters in GNN are frozen while $\mathbf{p}'_n$ is adjusted based on the newly added trainable parameters.

3.5 GAD Using AnomalyGFM

Zero-shot Inference. To build a foundation model for GAD with zero-shot learning capabilities, we directly apply AnomalyGFM pre-trained on auxiliary graphs to the test graph $\mathcal{G}_{test}$, meaning that Θ_n and Θ_a in $\Phi(\mathbf{z}_n; \Theta_n)$ and $\Phi(\mathbf{z}_a; \Theta_a)$ and the trainable parameters $\mathbf{W}$ in GNN are all frozen. This process allows us to obtain the residual features for each node in the test graph, the graph-agonistic normal prototype $\mathbf{p}_n$ and abnormal prototype $\mathbf{p}_a$. The anomaly score is determined by computing the similarity of the target node's representations residual with the two prototypes $\mathbf{p}_a$ and $\mathbf{p}_n$. Formally, the anomaly score s_i for the node v_i is defined as,

$$s_i = \exp\left(\mathbf{r}_i^{\mathrm{T}} \mathbf{p}_a\right) + \beta \exp\left(-\mathbf{r}_i^{\mathrm{T}} \mathbf{p}_n\right), \tag{9}$$

where β is the hyperparameter that controls the weight of two parts. Note that the anomaly scoring calculation consists of the direct similarity with $\mathbf{p}_a$ and the inverse similarity with $\mathbf{p}_n$.

Few-shot Inference. In the few-shot inference, the original parameters in GNN, mapping functions, and newly added parameters for refining the normal class prototype are frozen. The anomaly score of $\mathcal{V}_u^{Test}$ is calculated based on the prompt-tuning-based updated normal class prototype $\mathbf{p}'_{\mathbf{n}}$ and the pre-trained abnormal class prototype $\mathbf{p}_a$, which is defined as:

$$s_i = \exp\left(\mathbf{r}_i^{\mathrm{T}} \mathbf{p}_a\right) + \beta \exp\left(-\mathbf{r}_i^{\mathrm{T}} \mathbf{p}'_{\mathbf{n}}\right). \tag{10}$$

Inference on Very Large-scale Graphs. Most existing GAD methods typically require loading all nodes of the graph, which often leads to poor scalability on large graphs during inference. By learning the graph-agnostic prototypes, AnomalyGFM can generalize to very large-scale graphs through a subgraph-based inference approach. AnonalyGFM can effectively infer the anomaly score without considering the entire graph structure, eliminating the bottleneck of loading the full graph for GAD inference. This subgraph inference is also desired in privacy-sensitive settings where we do not want to reveal the entire graph structure to the detection models. Specifically, AnomalyGFM operates in a way that given a test node v_i and its subgraph $\mathcal{S}(v_i)$ with size s are given, it first computes its residual feature $\mathbf{r}_i$ as:

$$\mathbf{r}_i = \mathbf{h}_i - \frac{1}{|\mathcal{S}(v_i)|} \sum_{v_j \in \mathcal{S}(v_i)} \mathbf{h}_j, \tag{11}$$

where $\mathcal{S}(v_i)$ is extracted using a random walk starting with the target node v_i. Then, AnomalyGFM infers the anomaly score using the similarity of the residual feature to the normal and abnormal class prototypes as defined in Eq. (9).

Table 1: Key statistics of GAD datasets used in our experiments. Ano. indicates the number (rate) of anomalies in the graph and Sim. indicates the global average edge similarity.

Dataset	Domain	#Nodes	#Edges	#Feat.	Ano.	Sim.
Facebook	Social Networks	1,081	55,104	576	25(2.31%)	0.690
Reddit	Social Networks	10,984	168,016	64	366(3.33%)	0.997
Amazon	Co-review	10,244	175,608	25	693(6.66%)	0.645
Disney	Co-purchase	124	335	28	6(4.8%)	0.804
Amazon-all	Co-review	11,944	4,398,392	25	821(6.87%)	0.645
YelpChi-all	Co-review	45,941	3,846,979	32	6,674(14.52%)	0.905
Tolokers	Work Collaboration	11,758	519,000	10	2,566(21.8%)	0.814
Question	Social Networks	48,921	153,540	301	1,460(2.98%)	0.679
Elliptic	Bitcoin Transaction	46,564	73,248	93	4,545 (9.8%)	0.356
T-Finance	Transaction Record	39,357	21,222,543	10	1,803(4.6%)	0.107
T-Social	Social Friendship	5,781,065	73,105,508	10	174,280(3.0%)	0.307

4 Experiments

4.1 Experimental Setup

Datasets. We utilize 11 benchmark datasets from social networks, finance, and co-review domains for evaluation. These datasets are summarized in Table 1, with additional details provided in Appendix A. The social network datasets consist of Facebook [40], Reddit [14], Question [26] and T-Social [33]. The financial datasets include T-Finance [33] and Elliptic [37]. For the co-review category, we use Amazon [7], Amazon-all [7], and YelpChi-all [7]. Additionally, the Disney [29] and Tolokers [26] datasets, derived from co-purchase and work collaboration networks, are also included.

Zero-shot Competing Methods. For the zero-shot setting, we evaluate AnomalyGFM against the state-of-the-art approaches from three main categories as follows (1) unsupervised methods: AnomalyDAE [8], CoLA [19], TAM [27], and GADAM [4]; (2) supervised methods: GCN [13], GAT [34], BWGNN [33], GHRN [11], and XGBGraph [32]; and (3) general GFM methods, including GraphPrompt [20] for general graph tasks, and zero-shot GAD UNPrompt [23].

Few-shot Competing Methods. In few-shot setting, three generalist methods are icluded, *i.e.*GPPT [30], GraphPrompt [20] and ARC [18].

Table 2: AUROC and AUPRC results for zero-shot GAD on nine real-world datasets, with the models trained on Facebook only. For each dataset, the best performance per column within each metric is boldfaced, with the second-best underlined. Avg. denotes the average performance. P-values are from one-tailed Wilcoxon signed rank tests.

Metric	Method	Dataset: Reddit	Amazon	Disney	Aamzon-all	YelpChi-all	Tolokers	Question	Elliptic	T-Finance	Avg.	p-value
AUROC	Unsupervised Methods											
	AnomalyDAE (ICASSP'20)	0.5016	0.5818	0.4853	0.7228	0.5002	0.5948	0.4311	0.4197	0.2324	0.4966	0.007
	CoLA (TNNLS'21)	0.4623	0.4580	0.4696	0.4091	0.4879	0.4501	0.4945	0.5572	0.4889	0.4752	0.003
	TAM (NeurIPS'23)	0.5725	0.4720	0.4773	0.7543	0.4216	0.5351	0.5119	0.3282	0.2990	0.4857	0.003
	GADAM (ICLR'24)	0.4532	0.6646	0.4288	0.5959	0.4829	0.4832	**0.5594**	0.3922	0.1382	0.4664	0.007
	Supervised Methods											
	GCN (ICLR'17)	0.5645	0.5988	0.5000	0.7195	0.5486	0.5319	0.5161	**0.7640**	0.2345	0.5531	0.039
	GAT (ICLR'18)	0.5000	0.4981	0.5175	0.5005	0.4802	0.5030	0.4577	0.6588	0.5072	0.5136	0.007
	BWGNN (ICML'22)	0.5208	0.4769	0.6073	0.3648	0.5282	0.4877	0.4404	0.5843	0.5457	0.5062	0.003
	GHRN (WebConf'23)	0.5253	0.4560	0.5336	0.3382	0.5125	0.4860	0.4535	0.5400	0.5324	0.4863	0.003
	XGBGraph (NeurIPS'23)	0.4601	0.4179	0.6692	0.7950	0.4945	0.5462	0.5095	0.4274	0.3402	0.5177	0.003
	Generalist Methods											
	GraphPrompt (WebConf'23)	0.4677	0.4904	0.5192	0.3215	0.4976	0.4779	0.4204	0.3221	0.5405	0.4508	0.003
	UNPrompt (Arxiv'24)	0.5337	0.7525	0.6412	0.7962	0.5558	**0.6853**	0.4757	0.5901	0.2318	0.5847	0.074
	AnomalyGFM	**0.5974**	**0.8417**	**0.6751**	**0.9032**	**0.5791**	0.5843	0.5280	0.6195	**0.5614**	**0.6544**	/
AUPRC	Unsupervised Methods											
	AnomalyDAE (ICASSP'20)	0.0327	0.0833	0.0566	0.1921	0.1484	0.1876	0.0241	0.0798	0.0274	0.0924	0.003
	CoLA (TNNLS'21)	0.0391	0.0669	0.0701	0.0861	0.1466	0.0848	0.0292	0.0998	0.0430	0.0739	0.007
	TAM (NeurIPS'23)	0.0413	0.0666	0.0628	0.1736	0.1240	0.0970	0.0307	0.0697	0.0332	0.0776	0.007
	GADAM (ICLR'24)	0.0293	0.1562	0.0651	0.1595	0.1371	0.1001	0.0395	0.0733	0.0261	0.0873	0.003
	Supervised Methods											
	GCN (ICLR'17)	**0.0439**	0.0891	0.0484	0.1536	0.1735	0.1060	0.0387	**0.1963**	0.0274	0.0974	0.074
	GAT (ICLR'18)	0.0329	0.0688	0.0530	0.0696	0.1400	0.0822	0.0259	0.1366	0.0463	0.0728	0.003
	BWGNN (ICML'22)	0.0389	0.0652	0.0624	0.0586	0.1605	0.1030	0.0257	0.1158	0.0479	0.0753	0.007
	GHRN (WebConf'23)	0.0407	0.0633	0.0519	0.0569	0.1505	0.0957	0.0259	0.1148	0.0457	0.0717	0.007
	XGBGraph (NeurIPS'23)	0.0330	0.0536	0.1215	0.2307	0.1449	0.1256	0.0306	0.0816	**0.0754**	0.0996	0.027
	Generalist Methods											
	GraphPrompt (WebConf'23)	0.0334	0.0661	0.0610	0.0666	0.1542	0.2070	0.0266	0.0664	0.0492	0.0811	0.003
	UNPrompt (Arxiv'24)	0.0351	0.1602	0.1236	0.2430	0.1810	0.2219	0.0348	0.1278	0.0279	0.1283	0.003
	AnomalyGFM	0.0387	**0.5790**	**0.1242**	**0.6820**	**0.1819**	**0.2749**	**0.0397**	0.1371	0.0593	**0.2352**	/

Evaluation Metrics. Following the previous studies [18, 27, 33], two widely-used metrics, AUROC and AUPRC, are used to evaluate the performance of all methods. For both metrics, a higher value denotes better performance. Moreover, for each method, we report the average performance for 3 independent runs with different random seeds. One-tailed Wilcoxon signed rank test is performed to evaluate the statistical significance of the performance.

Implementation Details. For all competing methods and AnomalyGFM, the code is implemented with Pytorch 1.11 , PyG 2.1.0, DGL 1.0.1, and Python 3.8.19. All experiments are conducted on a Linux server with an Intel CPU (Intel Xeon Gold 6346 3.1GHz) and an Nvidia RTX3090 GPU with 24 GB GPU memory. The pre-train model is optimized using the Adam optimizer [12] with 300 epochs and a learning rate of $1e-4$ by default. The dimension of hidden space for representation and the prototype size are uniformly set to 300. The parameter α in the loss function is set to one by default. The setting of β is determined based on the prior we gain from the global average similarity information provided in Table 1 (see Sec. 4.6 for details). The μ and σ of the Gaussian distribution for both normal and abnormal class prototypes are set to zero and one by default. The size of the subgraph in subgraph inference is set to five by default. The implementation of competing methods is based on their publicly available official source code.

4.2 Zero-shot GAD Performance

The AUROC and AUPRC results under the zero-shot setting are shown in Table 2 where the model is trained on Facebook and inference is made on other datasets without any further tuning. Similar results can be found when the models are trained on the Amazon dataset in Appendix B.2. From the table, we observe (1) AnomalyGFM consistently outperforms the three types of competing methods, including unsupervised methods, supervised methods, and generalist methods on most datasets. Specifically, AnomalyGFM achieves the best AUROC performance on six datasets and the best AUPRC performance on the five datasets, with up to 11% improvement in AUROC and 44% improvement in AUPRC when compared to the best-competing method. The main reason is that AnomalyGFM performs inference based on two discriminative graph-agonistic prototypes distilled from the node residual feature, supporting strong generalization across the datasets. (2) Unsupervised methods often fail to produce consistently good results across the datasets due to the fact that each of their model designs is devised in a way to focus on some specific abnormal patterns for the training graphs, making them struggle to generalize to the new graphs. (3) Most supervised methods perform well in zero-shot inference, particularly simple approaches like GCN, which achieves top performance on the Elliptic dataset. This can be

Table 3: Results on the nine real-world GAD datasets under 1/5/10-shot with models tuned on the few-shot data.

Metric	Setting	Method	Reddit	Amazon	Disney	Aamzon-all	YelpChi-all	Tolokers	Question	Elliptic	T-Finance	Avg.	p-value
AUROC	1-shot	GPPT (KDD'22)	0.5000	0.5303	0.4997	0.5010	0.5000	0.5061	0.4921	0.6162	0.3647	0.5011	0.003
		GraphPrompt (WebConf'23)	0.4216	0.4882	0.4223	0.2631	0.4811	0.5328	0.4086	0.6001	0.4000	0.4464	0.004
		ARC (NeurIPS'24)	0.4899	0.4571	0.3578	0.4570	0.4910	0.4667	**0.5865**	0.2904	0.2484	0.4272	0.008
		AnomalyGFM	**0.5922**	**0.8531**	**0.6649**	**0.8972**	**0.5872**	**0.5898**	0.5303	**0.6199**	**0.5916**	**0.6584**	/
	5-shot	GPPT (KDD'22)	0.5000	0.5098	0.5000	0.5051	0.5000	0.5181	0.4959	0.5736	0.2609	0.4848	0.003
		GraphPrompt (WebConf'23)	0.4406	0.4900	0.6497	0.4726	0.5359	0.5381	0.4069	0.6012	0.4069	0.5046	0.003
		ARC (NeurIPS'24)	0.4720	0.4458	0.4435	0.4473	0.5112	0.4746	**0.5906**	0.2714	0.2168	0.4303	0.007
		AnomalyGFM	**0.6023**	**0.8600**	**0.6613**	**0.9011**	**0.5951**	**0.6095**	0.5426	**0.6119**	**0.6248**	**0.6676**	/
	10-shot	GPPT (KDD'22)	0.5000	0.5087	0.4769	0.5023	0.5000	0.4971	0.5047	0.4212	0.5539	0.4961	0.003
		GraphPrompt (WebConf'23)	0.4321	0.4906	0.6314	0.7167	0.5367	0.5329	0.3826	0.6221	0.4260	0.5301	0.007
		ARC (NeurIPS'24)	0.4867	0.4323	0.4769	0.4467	0.5145	0.4786	**0.5901**	0.2644	0.2298	0.4355	0.003
		AnomalyGFM	**0.6252**	**0.8649**	**0.6649**	**0.9215**	**0.6064**	**0.6140**	0.5611	**0.6303**	**0.6283**	**0.6796**	/
AUPRC	1-shot	GPPT (KDD'22)	0.0333	0.0766	0.0488	0.0687	0.1453	0.2204	0.0294	0.1239	0.0432	0.0877	0.003
		GraphPrompt (WebConf'23)	0.0283	0.0680	0.0486	0.0426	0.1113	0.2321	**0.0448**	0.1108	0.0302	0.0796	0.012
		ARC (NeurIPS'24)	0.0332	0.0581	0.0453	0.0590	0.1402	0.2122	0.0468	0.0701	0.0277	0.0769	0.011
		AnomalyGFM	**0.0398**	**0.5801**	**0.1223**	**0.6921**	**0.1852**	**0.2786**	0.0332	**0.1401**	**0.0601**	**0.2368**	/
	5-shot	GPPT (KDD'22)	0.0333	0.0692	0.0504	0.0716	0.1453	0.2265	0.0297	0.1127	0.0365	0.0861	0.004
		GraphPrompt (WebConf'23)	0.0285	0.0681	0.0892	0.0600	0.1661	**0.2957**	0.0327	0.1416	0.0360	0.1019	0.009
		ARC (NeurIPS'24)	0.0312	0.0571	0.0546	0.0572	0.1464	0.2150	**0.0471**	0.0726	0.0267	0.0786	0.011
		AnomalyGFM	**0.0401**	**0.5831**	**0.1257**	**0.6985**	**0.1918**	0.2866	0.0336	**0.1437**	**0.0622**	**0.2405**	/
	10-shot	GPPT (KDD'22)	0.0334	0.0691	0.0526	0.0698	0.1453	0.2178	0.0301	0.0905	0.0511	0.0844	0.004
		GraphPrompt (WebConf'23)	0.0278	0.0681	0.0848	0.1427	0.1649	**0.2922**	0.0263	0.1421	0.0382	0.1096	0.007
		ARC (NeurIPS'24)	0.0327	0.0557	0.0743	0.0583	0.1491	0.2168	**0.0463**	0.0677	0.0272	0.0809	0.011
		AnomalyGFM	**0.0444**	**0.5895**	**0.1399**	**0.7124**	**0.1990**	0.2897	0.0346	**0.1570**	**0.0644**	**0.2478**	/

attributed to their simple model design, avoiding heavy overfitting to the training graph. However, the performance still underperforms AnomalyGFM, demonstrating that the node representation residual is more generalized and our prototype alignment module provides a strong add-on to the binary loss function in our method. (4) Generalist methods, such as GraphPrompt, which are designed for general node/graph classification, struggle to perform well on GAD. This is due to the irregular nature of the abnormalities in different GAD datasets, hindering the applicability of the regular patterns/knowledge these generalist method learned from other tasks to GAD. AnomalyGFM outperforms UNPrompt on all datasets except Tolokers, which can be attributed to the node representation residual-based class prototypes offers more generalization power than the latent attribute predictability used in UNPrompt. Further, our statistical significance test results show that the superiority of AnomalyGFM over all competing methods is significant at the 90%/95% confidence level in both metrics.

4.3 Few-shot GAD Performance

To demonstrate the effectiveness in the few-shot scenarios, the comparison of AnomalyGFM to GraphPrompt [20] and ARC [18] is done under 1/5/10-shot. The AUROC and AUPRC results are shown in Table 3. In the few-shot GAD scenario, AnomalyGFM outperforms these two generalist methods on most datasets in both AUROC and AUPRC. GraphPrompt continues to underperform AnomalyGFM since the presence of only one class of samples renders its fine-tuning ineffective, leading to suboptimal performance. ARC also leverages residual features and is specifically designed for few-shot GAD datasets achieving the best performance on the Question and the second performance on most datasets. Compared to ARC, AnomalyGFM achieves better sample-efficiency tuning due to the learning of class-level prototypes that are agnostic to different domains of graphs, enhancing the generalization across the graph, thereby yielding the best performance on the eight datasets. Tuning the abnormal prototype is also possible when a few labeled anomalies are available and the results are provided in App. B.1.

Table 4: Subgraph inference on the very large-scale graphs. '/' indicates that the model cannot handle the dataset.

Metric	Method	T-Finance	T-Social
AUROC	*Unsupervised Methods*		
	TAM (NeurIPS'23)	0.2990	/
	GADAM (ICLR'24)	0.1382	0.5155
	Supervised Methods		
	BWGNN (ICML'22)	0.5457	0.4964
	GHRN (WebConf'23)	0.5324	0.4934
	XGBGraph (NeurIPS'23)	0.3402	0.4602
	AnomalyGFM	**0.7852**	**0.5991**
AUPRC	*Unsupervised Methods*		
	TAM (NeurIPS'23)	0.0332	/
	GADAM (ICLR'24)	0.0261	0.0285
	Supervised Methods		
	BWGNN (ICML'22)	0.0479	0.0301
	GHRN (WebConf'23)	0.0457	0.0303
	XGBGraph (NeurIPS'23)	0.0754	0.0305
	AnomalyGFM	**0.1059**	**0.0398**

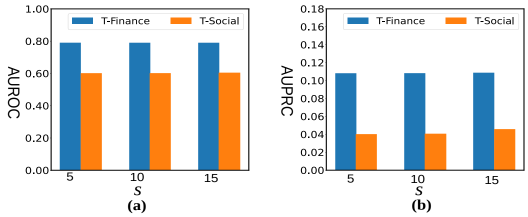

Figure 3: AnomalyGFM performance w.r.t subgraph size s.

4.4 Scale up to Very Large Graphs

We choose two large-scale graphs T-Finance (with a large number of edges) and T-Social (with a large number of nodes) to evaluate the generalization of AnomalyGFM on very large graphs. The results are shown in Table 4. We can observe that AnomalyGFM outperforms the unsupervised methods and supervised methods on these datasets. This is mainly because the learned data-independent prototypes can effectively measure the abnormality of node representation residual, even in the presence of subgraphs. The deviations of the target node from the contextual node are still preserved in the subgraph, thereby enabling effective subgraph-based inference.

We also evaluate the sensitivity of AnomalyGFM with subgraph sizes and the results are shown in Fig. 3. The performance of AnomalyGFM remains stable with varying subgraph sizes, demonstrating its robust generalization capability in modeling large graphs.

4.5 Ablation Study

To verify the effectiveness of each component of AnomalyGFM, we designed three variants including (i) **BCE** which replaces the similarity measurement with the predicted probability from the BCE loss function for anomaly scoring. (ii) **BCE Residual (BCE-R)** which firstly removes the alignment loss and applies the BCE loss on the residual features to differentiate normal and abnormal nodes. Then, the predicted probability is used as the anomaly score. (iii) **Feature Alignment (FA)** which replaces the alignment on the residual feature with the alignment on the original feature. The results are shown in Table 5. **FA** achieves the highest AUROC on the Elliptic and T-Finance datasets, suggesting that learning prompts from the original features can also be effective as the original feature also maintains anomaly discriminability but they aren't in the unified feature space leading to sub-optimal performance. **BCE** does not perform well except on the small dataset Disney. It directly uses the predicted probability as the anomaly score, which makes it struggle to generalize to new graphs due to the differences between the training and test sets. **BCE-R** does not achieve good performance compared to **BCE**. The reason is that although the residual features are discriminative, many samples located on the decision boundaries after being mapped to a unified feature space may not be effectively scored by the supervised classifier. In contrast, aligning the learnable prototype with the average node representation residual and using a similarity measurement for scoring can effectively discriminate the normal and abnormal nodes.

4.6 Hyperparameter Sensitivity Analysis

We evaluate the sensitivity of AnomalyGFM w.r.t the size of the prototype T, hyperparameter α, and commonality dimension d'. Additional sensitivity analyses are in Appendix B.3.

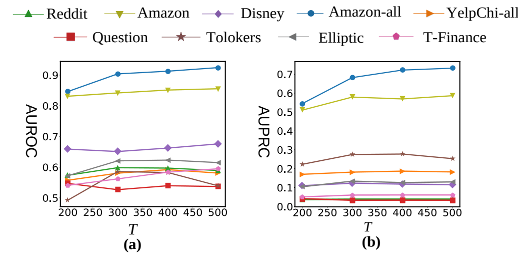

Figure 4: AnomalyGFM performance w.r.t prototype size T.

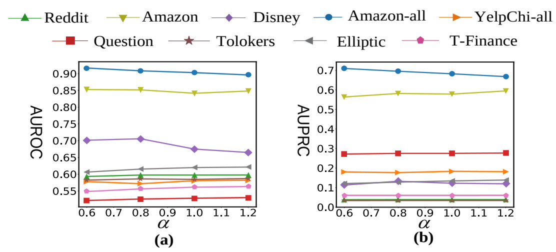

Figure 5: AnomalyGFM performance w.r.t α.

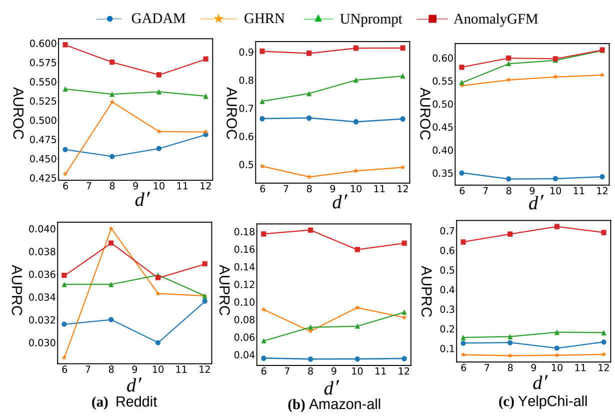

Figure 6: AUROC and AUPRC of AnomalyGFM w.r.t d'

AUROC and AUPRC results of AnomalyGFM w.r.t. prototype size T. As shown in Fig. 4, AnomalyGFM generally achieves better performance with increasing prototype size. This suggests that a larger prototype size provides more information and enhances the modeling capability of the foundation model.

AUROC and AUPRC results of AnomalyGFM w.r.t. α. We vary the hyperparameter α in the range of [0.6, 1.2] with the interval of 0.1 and the results are shown in Fig. 5. We can see that AnomalyGFM remains stable with different α on most datasets in terms of both metrics, demonstrating the robustness of AnomalyGFM against this hyperparameter.

Sensitivity of AnomalyGFM w.r.t common dimensionality d'. To further demonstrate AnomalyGFM's generalization, we evaluate AnomalyGFM under different d'. The results are shown in Fig. 6. AnomalyGFM outperforms all the competing methods under different dimensions. The main reason is that the deviation between connected nodes is still preserved with different d' which can be efficiently measured by the graph-agnostic prototypes.

Table 5: Ablation study results for AnomalyGFM and its variants.

Metric	Method	Reddit	Amazon	Disney	Amazon-all	YelpChi-all	Tolokers	Question	Elliptic	T-Finance	Avg.
AUROC	FA	0.5697	0.5680	0.6073	0.5097	0.5211	0.5594	0.5326	**0.7765**	**0.6674**	0.5901
	BCE	0.5445	0.4308	**0.7542**	0.5545	0.5232	0.5187	0.4593	0.2581	0.6581	0.5223
	BCE-R	0.5108	0.5314	0.4887	0.5777	0.4590	0.4788	0.4601	0.5414	0.5380	0.5095
	AnomalyGFM	**0.5974**	**0.8417**	0.6751	**0.9032**	**0.5791**	**0.5843**	**0.5380**	0.6195	0.5614	**0.6555**
AUPRC	FA	**0.0400**	0.0776	0.0960	0.0684	0.1555	0.2478	0.0356	0.1199	**0.1495**	0.1100
	BCE	0.0394	0.0544	**0.2254**	0.0718	0.1594	0.2264	0.0272	0.0287	0.1389	0.1079
	BCE-R	0.0314	0.0677	0.0685	0.0787	0.1294	0.2122	0.0305	0.0658	0.0971	0.0868
	AnomalyGFM	0.0387	**0.5790**	0.1242	**0.6820**	**0.1819**	**0.2749**	**0.0397**	**0.1371**	0.0593	**0.2352**

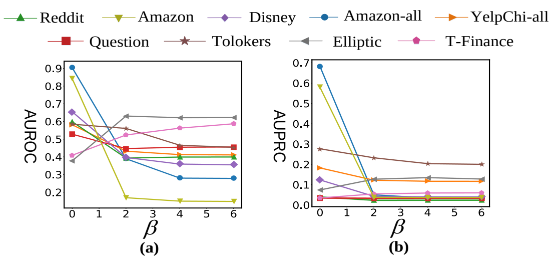

Figure 7: AUROC and AUPRC of AnomalyGFM w.r.t β.

Impact of β in anomaly scoring. The performance under different β is shown in Fig. 7. As shown in Table 9 in App. B.3, the set of β values are determined based on the prior we gain from the global average similarity. Specifically, the hyperparameter in anomaly scoring β is set to zero by default for graphs with high global similarity such as Amazon and Yelp, and four for those with low global similarity such as Ellitipic and T-Finance, see the detailed global similarity information in Table 1. In the few-shot scoring, β is set to 0.5 by default for graphs with high global similarity. The main reason for these settings is that for graphs with high global average similarity, where node features are relatively similar, the deviations in local neighborhood would be small, and the fluctuations in scoring based on the normal prototype may affect the overall anomaly score. Therefore, we solely use the abnormal class prototype or assign a smaller weight for the normal class prototype in scoring. For nodes with less similar features, we use a mix of normal and abnormal prototypes for scoring. In the few-shot setting, the labeled normal nodes help refine the normal prototypes, allowing the anomaly scores to be computed using both normal and abnormal prototypes.

Table 6: Training and inference time (seconds) comparison.

Methods	AnomalyDAE	TAM	GAT	BWGNN	UNPrompt	AnomalyGFM
Training Time	86.04	479.70	2.43	4.86	2.08	7.45
Inference Time	170.46	252.52	91.88	98.33	19.47	11.00

5 Time Complexity Analysis

In this section, we analyze the time complexity of AnomalyGFM. We build the GCN as the backbone of AnomalyGFM which takes $O(mdh)$, where m is the number of non-zero elements in matrix $\mathbf{A}$ (the number of edge in graph), d is the dimension of representation, and h is the number of feature maps. The computation of the residual feature takes $O(N^2d)$. The MLP layer mapping the representation to the anomaly score takes $O(Nd)$. The feature mapping of normal class prototype and abnormal class prototype takes $O(2d^2)$. The alignment loss between residual features and prototypes takes $O(2Nd)$. Thus, the overall complexity of the pre-training of AnomalyGFM is $O(mdh + N^2d + 3Nd + 2d^2)$. In the inference, we employ the residual feature's similarity with two prototypes which takes $O(2Nd)$. Therefore, the overall complexity of inference of AnomalyGFM is $O(mdh + N^2d + 2d^2 + 2Nd)$. The complexity of inference on large-scale graph takes $O\left(K\left(mdh + n^2d + 2d^2 + 2nd\right)\right)$ with n as subgraph size and K as the number of subgraphs. It is significantly lower than other methods, as they require loading entire graphs and performing reconstruction or affinity calculations, leading to higher space/time complexity. In Table 6, we report the running time comparison on Facebook and inference time on YelpChi-all. AnomalyGFM requires much less training and inference time compared to TAM and AnomalyDAE, demonstrating the efficiency. For inference, AnomalyGFM is the most efficient method because scoring relies solely on measuring the similarity between node representation residuals and prototypes.

6 Conclusion

In this paper, we build a GAD-oriented graph foundation model, AnoamlyGFM, that can work effectively under both few-shot and zero-shot scenarios. AnomalyGFM is pre-trained to learn discriminative and data-independent prototypes by aligning them with the graph-agnostic node representation residuals. This provides a consistent and identical way for abnormality measurement using the similarity between residual node representation and the learned class prototypes, facilitating the strong generalization in both zero-shot and few-shot inference. Extensive experiments on 11 datasets demonstrated the effectiveness and generalization of AnomalyGFM.

Acknowledgments

This research is supported by A*STAR under its MTC YIRG Grant (No. M24N8c0103), the Ministry of Education, Singapore under its Tier-1 Academic Research Fund (No. 24-SIS-SMU-008), the Lee Kong Chian Fellowship, and ARC under Grant DP240101349.

References

[1] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* 29 (2015), 626–688.

[2] Dominique Beaini, Shenyang Huang, Joao Alex Cunha, Zhiyi Li, Gabriela Moisescu-Pareja, Oleksandr Dymov, Samuel Maddrell-Mander, Callum McLean, Frederik Wenkel, Luis Müller, et al. 2023. Towards foundational models for molecular learning on large-scale multi-task datasets. *arXiv preprint arXiv:2310.04292* (2023).

[3] Bo Chen, Jing Zhang, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang. 2022. Gccad: Graph contrastive coding for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 8037–8051.

[4] Jingyan Chen, Guanghui Zhu, Chunfeng Yuan, and Yihua Huang. 2024. Boosting Graph Anomaly Detection with Adaptive Message Passing. In *The Twelfth International Conference on Learning Representations*.

[5] Kaize Ding, Jundong Li, Nitin Agarwal, and Huan Liu. 2021. Inductive anomaly detection on attributed networks. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 1288–1294.

[6] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM international conference on data mining*. SIAM, 594–602.

[7] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 315–324.

[8] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. 2020. Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5685–5689.

[9] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. 2024. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems* 36 (2024).

[10] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2023. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562* (2023).

[11] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference 2023*. 1528–1538.

[12] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

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

[15] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. 2024. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1725–1735.

[16] Yiqing Lin, Jianheng Tang, Chenyi Zi, H Vicky Zhao, Yuan Yao, and Jia Li. 2024. UniGAD: Unifying Multi-level Graph Anomaly Detection. *arXiv preprint arXiv:2411.06427* (2024).

[17] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*. 3168–3177.

[18] Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan. 2024. ARC: A Generalist Graph Anomaly Detector with In-Context Learning. *arXiv preprint arXiv:2405.16771* (2024).

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

[20] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023*. 417–428.

[21] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering* 35, 12 (2021), 12012–12038.

[22] Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. [n. d.]. Position: Graph Foundation Models Are Already Here. In *Forty-first International Conference on Machine Learning*.

[23] Chaoxi Niu, Hezhe Qiao, Changlu Chen, Ling Chen, and Guansong Pang. 2024. Zero-shot Generalist Graph Anomaly Detection with Unified Neighborhood Prompts. *arXiv preprint arXiv:2410.14886* (2024).

[24] Junjun Pan, Yixin Liu, Yizhen Zheng, and Shirui Pan. 2023. PREM: A Simple Yet Effective Approach for Node-Level Graph Anomaly Detection. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1253–1258.

[25] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)* 54, 2 (2021), 1–38.

[26] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640* (2023).

[27] Hezhe Qiao and Guansong Pang. 2023. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. *Advances in Neural Information Processing Systems* 36 (2023).

[28] Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. 2024. Deep Graph Anomaly Detection: A Survey and New Perspectives. *arXiv preprint arXiv:2409.09957* (2024).

[29] Patricia Iglesias Sánchez, Emmanuel Müller, Fabian Laforet, Fabian Keller, and Klemens Böhm. 2013. Statistical selection of congruent subspaces for mining attributed graphs. In *2013 IEEE 13th international conference on data mining*. IEEE, 647–656.

[30] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 2022. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1717–1727.

[31] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2023. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2120–2131.

[32] Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. 2023. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Advances in Neural Information Processing Systems* 36 (2023), 29628–29653.

[33] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. 2022. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*. PMLR, 21076–21089.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[35] Qizhou Wang, Guansong Pang, Mahsa Salehi, Wray Buntine, and Christopher Leckie. 2023. Open-Set Graph Anomaly Detection via Normal Structure Regularisation. *arXiv preprint arXiv:2311.06835* (2023).

[36] Xuhong Wang, Baihong Jin, Ying Du, Ping Cui, Yingshui Tan, and Yupu Yang. 2021. One-class graph neural networks for anomaly detection in attributed networks. *Neural computing and applications* 33 (2021), 12073–12085.

[37] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591* (2019).

[38] Zhihao Wen, Yuan Fang, Yihan Liu, Yang Guo, and Shuji Hao. 2023. Voucher Abuse Detection with Prompt-based Fine-tuning on Graph Neural Networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4864–4870.

[39] Xiongxiao Xu, Kaize Ding, Canyu Chen, and Kai Shu. 2024. MetaGAD: Meta Representation Adaptation for Few-Shot Graph Anomaly Detection. In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 1–10.

[40] Zhiming Xu, Xiao Huang, Yue Zhao, Yushun Dong, and Jundong Li. 2022. Contrastive attributed network anomaly detection with data augmentation. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 444–457.

[41] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. 2024. 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*. 4443–4454.

[42] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. 2023. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961* (2023).

[43] Shuang Zhou, Qiaoyu Tan, Zhiming Xu, Xiao Huang, and Fu-lai Chung. 2021. Subtractive aggregation for attributed network anomaly detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3672–3676.

[44] Jiawen Zhu and Guansong Pang. 2024. Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17826–17836.

[45] Chenyi Zi, Haihong Zhao, Xiangguo Sun, Yiqing Lin, Hong Cheng, and Jia Li. 2024. ProG: A Graph Prompt Learning Benchmark. *arXiv preprint arXiv:2406.05346* (2024).

A Detailed Dataset Description

A detailed introduction of all the used datasets is given as follows.

- Facebook [40]: It is a social network where nodes represent users and edges indicate relationships between them. Anomalies are the nodes that either connect manually constructed clusters or display attributes differing from those of their contextual neighbors.
- Reddit [14]: It is a user-subreddit graph, capturing one month's worth of posts shared across various subreddits at Reddit. The users who have been banned by the platform are labeled anomalies. The text of each post is transformed into the feature vector and the features of the user and subreddits are the feature summation of the post they have posted.
- Amazon [7]: It includes product reviews under the Musical Instrument category. The users with more than 80% of helpful votes were labeled as begin entities, with the users with less than 20% of helpful votes treated as fraudulent entities.
- Disney [29]: It comes from the Amazon co-purchase network of movies where the attributes are the prices, ratings, number of reviews, etc. The anomalies are labeled manually by majority vote and the ground truths are derived from amazonfail stage information.
- Amazon-all [7]: It includes three types of relations: U-P-U (users reviewing at least one same product), U-S-U (users giving at least one same star rating within one week), and U-V-U (users with top-5% mutual review similarities). Amazon-all is formed by treating the different relations as a single relation following [3, 27].
- YelpChi-all [7]: Similar to Amazon-all, YelpChi-all includes three types of edges: R-U-R (reviews posted by the same user), R-S-R (reviews for the same product with the same star rating), and R-T-R (reviews for the same product posted in the same month). YelpChi-all is formed by treating the different relations as a single relation following [3, 27].
- Tolokers [26]: It is obtained from Toloka crowdsourcing platform where the node represents the person who has participated in the selected project and the edge connects two workers means that they work on the same task. The attributes of the node are the profile and task performance statistics of workers.
- Question [26]: It is collected from the website Yandex Q where the node represents the user and the edge connecting the node represents the user who has answered other's questions during a one year period. The attribute is the mean of embeddings for words in the description of the user.
- Elliptic [37]: It is a Bitcoin transaction network in which the node represents the transactions and the edge is the flow of Bitcoin currency. The Bitcoin transaction is mapped to real-world entities associated with licit categories.
- T-Finance [33]: It is a financial transaction network where the node represents an account and the edge represents two accounts that have transaction. The features of each account are related to some attributes of logging like registration days, logging activities, and interaction frequency, etc. The users are labeled as anomalies when they fall into the categories like fraud money laundering and online gambling.
- T-Social [33]: It is a social network where the node represents the user and the edge indicates the relationship between users for more than three months. The attributes of each account are some logging information like registration days, logging activities, interaction frequency, etc.

B Additional Experimental Results

B.1 Few-shot Tuning on Abnormal Nodes

Additionally, tuning the abnormal prototype $\mathbf{p_a}$ is also feasible when few-shot abnormal node labels are available. Give some labeled abnormal nodes $\mathcal{V}_l^{Test}$, the fine tuning process is formulated as

$$L_{pt} = \sum_{i=1}^{\left|\mathcal{V}_l^{Test}\right|} \left\|\mathbf{r}_i - \mathbf{p}'_\mathbf{a}\right\|_2^2, \tag{12}$$

where $\mathbf{p}'_\mathbf{a} = \mathbf{p}_a + g(\mathbf{p}_a; \phi) + \Psi_{\mathbf{p}_a}$, with a similar formulation applied during the tuning of few-shot normal nodes. To evaluate the performance of AnomalyGFM under few-shot abnormal nodes setting. we include a few labeled abnormal nodes, and treat the unlabeled nodes as normal. These can be combined to serve as the training source. We compare AnomalyGFM with the few-shot GAD method MetaGAD [39] and fully supervised methods BWGNN [33]. We show results in Table 7 for the scenario where few-shot labeled abnormal nodes are available. As shown in Table 7, AnomalyGFM outperforms MetaGAD and BWGNN since they are limited by the scarcity of anomalies on the target graphs, whereas AnomalyGFM has better generalization since it only needs to adapt the pre-trained anomaly prototype with the few anomaly nodes to the target graphs.

Table 7: Comparison under 1/5-shot anomalies

Metric	Setting	Method	Dataset	
			Amazon-all	YelpChi-all
AUROC	1-shot	BWGNN (ICML'22)	0.7441	0.5447
		MetaGAD (DSAA'24)	0.8602	0.5434
		AnomalyGFM	**0.9155**	**0.5941**
	5-shot	BWGNN (ICML'22)	0.8292	0.4759
		MetaGAD (DSAA'24)	0.8558	0.5741
		AnomalyGFM	**0.9246**	**0.6004**
AUPRC	1-shot	BWGNN (ICML'22)	0.2947	0.1703
		MetaGAD (DSAA'24)	**0.7562**	0.1681
		AnomalyGFM	0.6956	**0.1922**
	5-shot	BWGNN (ICML'22)	0.4433	0.1408
		MetaGAD (DSAA'24)	**0.7567**	0.1814
		AnomalyGFM	0.7538	**0.1935**

B.2 Train the Model on Amazon and Evaluate on Other Datasets

To further demonstrate the effectiveness of AnomalyGFM, we pre-train the AnomalyGFM on Amazon and evaluate it on the other datasets. The AUROC and AUPRC results are shown in the Table 8. AnomalyGFM consistently outperforms competing methods from supervised methods, unsupervised methods, and generalist methods across most datasets in both AUROC and AUPRC, demonstrating its strong generalization when pre-trained on a different auxiliary dataset.

Table 8: Comparison of AUROC and AUPRC results on 8 real-world GAD datasets under zero-shot setting with the models trained on Amazon only.

Metric	Method	Facebook	Reddit	Disney	YelpChi-all	Tolokers	Question	Elliptic	T-Finance	Avg.	p-value
AUROC	Unsupervised Methods										
	AnomalyDAE (ICASSP'20)	0.6123	<u>0.5799</u>	0.4938	0.4912	0.5080	0.5401	0.2769	0.2886	0.4738	0.0156
	CoLA (TNNLS'21)	0.5427	0.4962	0.5455	0.4937	0.4680	0.4768	<u>0.6654</u>	0.4108	0.5123	0.0390
	GADAM (ICLR'24)	0.6024	0.4720	0.3966	0.5289	<u>0.5193</u>	**0.5575**	0.2907	0.1463	0.4391	0.0234
	Supervised Methods										
	BWGNN (ICML'22)	0.5441	0.4026	0.4196	0.5841	0.4430	0.5117	0.5471	**0.6196**	0.5089	0.0390
	XGBGraph (NeurIPS'23)	0.4869	0.4869	0.4376	<u>0.5869</u>	0.4710	<u>0.5430</u>	0.5367	<u>0.5950</u>	<u>0.5180</u>	0.0781
	Generalist Methods										
	GraphPrompt (WebConf'23)	0.3093	0.4511	<u>0.7128</u>	0.4994	0.5161	0.4720	**0.6772**	0.4753	0.5141	0.0546
	UNPrompt (Arxiv'24)	<u>0.7917</u>	0.5356	0.6959	0.5448	0.4949	0.4467	0.3558	0.1805	0.5057	0.0078
	AnomalyGFM	**0.8279**	**0.6331**	**0.7135**	**0.5898**	**0.5305**	0.5268	0.6010	0.5784	**0.6251**	/
AUPRC	Unsupervised Methods										
	AnomalyDAE (ICASSP'20)	0.0675	<u>0.0413</u>	0.0583	0.1479	0.2253	0.0361	0.0630	0.0300	0.0836	0.0234
	CoLA (TNNLS'21)	0.0468	0.0327	0.0717	0.1474	0.2038	0.0289	<u>0.1427</u>	0.0367	0.0888	0.0546
	GADAM (ICLR'24)	0.0461	0.0299	0.0732	0.1602	<u>0.2348</u>	0.0424	0.0636	0.0255	0.0844	0.0156
	Supervised Methods										
	BWGNN (ICML'22)	0.0289	0.0263	0.0494	0.1975	0.2073	0.0316	0.0992	**0.0635**	0.0879	0.0390
	XGBGraph (NeurIPS'23)	0.0268	0.0315	0.0541	**0.1994**	0.1982	**0.0426**	0.0991	0.0538	0.0881	0.0546
	Generalist Methods										
	GraphPrompt (WebConf'23)	0.0169	0.0298	<u>0.1157</u>	0.1481	0.2166	<u>0.0425</u>	**0.1590**	0.0464	0.0968	0.1484
	UNPrompt (Arxiv'24)	**0.2291**	0.0340	0.0933	0.1767	0.2294	0.0274	0.0708	0.0258	<u>0.1108</u>	0.1953
	AnomalyGFM	<u>0.1094</u>	**0.0440**	**0.1593**	<u>0.1979</u>	**0.2505**	0.0322	0.1112	<u>0.0604</u>	**0.1206**	/

B.3 Sensitivity Analysis

Impact of β in Anomaly Scoring. As shown in Table 9, the set of β values are determined based on the prior we gain from the global average similarity in Table 1. It is set to 0 by default for graphs with high global similarity and 4 for graphs with low global similarity. In the few-shot scoring, β is set to 0.5 by default for graphs with high global similarity and four for graphs with low global similarity, allowing the anomaly scores to be computed using both normal and abnormal class prototypes, where the normal class prototype is refined using the labeled normal nodes.

Table 9: The value of β in anomaly scoring in both zero-shot and few-shot setting. Sim. indicate the global average edge similarity of the graph.

Setting	**Sim.**>0.5	**Sim.**<=0.5
Zero-shot Setting	0	4
Few-shot Setting	0.5	4

Impact of μ and σ in Gaussian distribution initialization. To evaluate the effectiveness of AnomalyGFM, we set different values for μ and σ during the Gaussian distribution initialization for both the normal class prototype and the abnormal class prototype, see in the Table 10. The performance remains stable when we adjust μ and σ, indicating that AnomalyGFM is not dependent on the initial Gaussian distribution. This is mainly because the supervised loss function provides sufficient supervision, reducing the reliance on the Gaussian distribution.

Table 10: Performance of AnomalyGFM under different initialization hyperparameters for Gaussian distribution

Metric	Initialization	Amazon-all	Tolokers	Question	Elliptic	T-Finance
AUROC	$\mu = 0, \sigma = 1$	0.9032	0.5843	0.5280	0.6195	0.5614
	$\mu = 0, \sigma = 0.5$	0.9229	0.5981	0.5378	0.6068	0.5902
	$\mu = 0.1, \sigma = 0.5$	0.8756	0.6004	0.5437	0.5907	0.5727
	$\mu = 0.2, \sigma = 1$	0.8963	0.5975	0.5429	0.6099	0.5922
AUPRC	$\mu = 0, \sigma = 1$	0.6820	0.2749	0.0335	0.1333	0.0593
	$\mu = 0, \sigma = 0.5$	0.7334	0.2824	0.0345	0.1234	0.0612
	$\mu = 0.1, \sigma = 0.5$	0.6568	0.2870	0.0342	0.1096	0.0596
	$\mu = 0.2, \sigma = 1$	0.6959	0.2853	0.0339	0.1177	0.0604