Towards Generalist Graph-Level Anomaly Detection

Junjun Pan[1], **Yixin Liu**[1]*, **Yu Zheng**[1], **Fuyi Li**[2], **Alan Wee-Chung Liew**[1], **Shirui Pan**[1]
[1]School of Information and Communication Technology, Griffith University, Australia
[2]South Australian immunoGENomics Cancer Institute (SAiGENCI), Adelaide University, Australia
junjun.pan@griffithuni.edu.au, fuyi.li@adelaide.edu.au
{yixin.liu, yu.zheng, a.liew, s.pan}@griffith.edu.au

Abstract

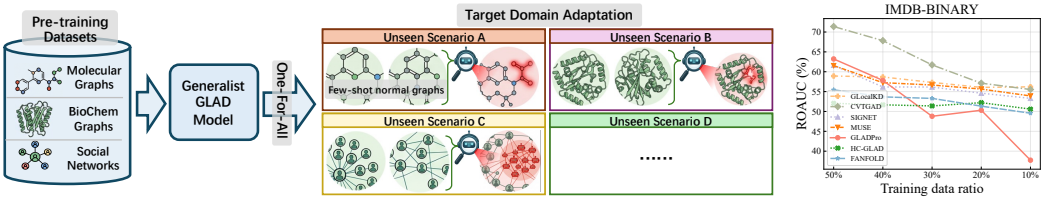
Graph-level anomaly detection (GLAD) aims to identify graphs that deviate significantly from the majority and plays a crucial role in applications such as molecular analysis and fraud detection. While recent advances have achieved competitive performance, existing approaches mainly follow a dataset-specific paradigm, requiring substantial in-domain data and training cost when applied to new application scenarios. This limits their scalability and practicality in real-world applications where labeled data is scarce and diverse domains are continuously emerging. In this paper, we investigate the problem of generalist GLAD, which seeks to learn a single unified model capable of detecting anomalous graphs across multiple domains with minimal supervision on the target data. This problem introduces key challenges, including learning transferable patterns from heterogeneous graph datasets and effective adaptation under limited supervision. To address these challenges, we propose GenGLAD, a generalist GLAD approach that leverages discriminative subgraph extraction as the core mechanism to learn transferable anomaly knowledge and enable efficient adaptation to new domains. Specifically, GenGLAD employs a graph neural network-based discriminative subgraph extraction model to learn the anomaly-discriminative substructures that distinguish anomalous and normal graphs across diverse domains. During the test-time adaptation phase, multi-dimensional discrepancies between the extracted subgraph and multi-level contextual information are calculated to characterize anomalies from multiple perspectives. For efficient adaptation, a lightweight scoring module is introduced to automatically decide the most informative signals, improving robustness against misleading patterns. Extensive experiments on diverse benchmark datasets demonstrate that the proposed approach achieves strong performance, superior generalization ability, and improved data efficiency compared to existing methods.

1 Introduction

Graphs are a ubiquitous data structure for modeling discrete objects and their relationships, which makes them widely used in fields such as chemistry, biology, and social network analysis [46, 56]. Building on this foundation, graph-level anomaly detection (GLAD) aims to identify graph instances that are significantly different from the majority of graphs [18]. Due to its strong capability for solving various real-world challenges, such as identifying toxic chemicals [2], spotting fraudster networks [25] and safeguarding multi-agent systems [28], GLAD has attracted sustained research attention. Recently, graph neural network (GNN)-based methods have become the dominant approach for the GLAD problem, showcasing strong performance compared to traditional kernel-based methods [18, 25, 43].

*Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Generalist GLAD paradigm. (b) AUROC comparison.

Figure 1: (a) Sketch map the generalist GLAD paradigm, where one model fits multiple new datasets. (b) Performance of unsupervised GLAD methods reduces with the reduction of training set size.

Despite the competitive detection performance they achieved, existing GLAD approaches follow the traditional "one model per dataset" paradigm, requiring large amounts of in-domain data to train a dedicated detection model for each dataset. This learning paradigm brings significant limitations for GLAD applications. On the one hand, deploying a dataset-specific model requires sufficient in-domain data for model training, which limits its practical applicability in domains where data are scarce or difficult to collect. On the other hand, training a dataset-specific model is usually computationally expensive, involving high time and hardware costs, making it inefficient for large-scale databases.

Given the aforementioned limitations, a natural question arises: *Can we train a "one-for-all" GLAD model that can generalize to detecting anomalous graphs across diverse application scenarios, without heavy adaptation cost?* Following the previous success of generalist anomaly detection methods [19, 58], developing a **generalist GLAD** approach is a promising solution to the question. As shown in Figure 1a, a generalist GLAD model can be trained on a large collection of diverse datasets, acquiring strong generalization capabilities for various GLAD tasks. When applied to an arbitrary unseen target dataset, it only requires a few-shot normal samples for minimal-cost adaptation to detect anomalous graphs. As the knowledge learned from large-scale pre-training can be transferred to target datasets, the generalist GLAD paradigm eliminates the need for heavy dataset-specific training, reducing both data requirements and training costs.

Despite the promising potential of the generalist paradigm, it is non-trivial to develop a generalist GLAD approach. While recent efforts such as ARC [19] and AnomalyGFM [34] have promising generalist capabilities for node-level graph anomaly detection, extending such paradigms to GLAD comes with unique challenges. Firstly, **(C1) How to learn generalizable patterns from diverse training datasets?** Different graph-level applications lie in different semantic spaces and different structural priors. As a result, the characteristics of normal and abnormal graphs can vary widely across domains, which makes obtaining generalizable knowledge difficult. For example, in a social network, a fraudster community is usually characterized by dense local structures due to spamming behavior, while in molecular graphs, a small motif that exhibits specific functional groups can determine biotoxicity. In this case, how to develop a generalizable model that can capture transferable patterns across diverse domains remains a fundamental challenge. Moreover, **(C2) How can models effectively adapt to a new test domain under limited supervision?** Even if a generalist model can learn transferable patterns across domains, making reliable prediction on a target dataset still requires leveraging a few-shot set of normal samples, which characterize the specific distributional properties of the dataset. Nevertheless, unsupervised GLAD models typically rely heavily on sufficient data to capture the normal distribution: as shown in Figure 1b, their performance degrades sharply as the amount of training data decreases. In this context, how to effectively exploit the limited few-shot normal samples for efficient adaptation of pre-trained generalist GLAD models to unseen domains still remains an open problem.

To address the aforementioned challenges, we propose GenGLAD, a **Gen**eralist **GLAD** approach based on discriminative subgraph extraction. Our core idea is to learn transferable *anomaly-discriminative subgraphs* (e.g., the substructures in red in Figure 1a) through cross-domain pre-training, and adapt them to unseen domains via discrepancy-based evidence modeling under few-shot supervision. Specifically, to tackle **(C1)**, GenGLAD incorporates a memory-guided discriminative subgraph extraction model as its backbone network, learning to capture shared fundamental knowledge for identifying the most anomaly-discriminative substructures that distinguish normal and anomalous graphs from arbitrary graph samples. To enhance transferable knowledge learning, GenGLAD empowers cross-domain pre-training by storing learned knowledge in a memory bank, and also facilitating task-relevant knowledge retrieval with the help of contextual priors. Moreover,

to address **(C2)**, we construct multi-dimensional discrepancy scores between the discriminative subgraph of each graph sample and the contextual information, using them as prediction evidence. In this way, GenGLAD not only leverages few-shot contextual samples for domain-specific adaptation, but also captures anomaly evidence from fine-grained instance-level cues to coarse-grained cross-domain knowledge. With the discrepancy-based evidence, we can perform efficient adaptation by learning a lightweight scoring module with a linear layer to support anomaly prediction on new datasets. To summarize, this paper makes the following contributions:

- **Problem.** To the best of our knowledge, we are the first to formalize and investigate the generalist GLAD problem, aiming to detect anomalous graphs from various domains with a single pre-trained GLAD model with limited in-domain supervision when adapting to arbitrary unseen datasets.
- **Method.** We propose a novel generalist GLAD method, GenGLAD, which learns transferable anomaly-discriminative substructures through memory-guided cross-domain pre-training and performs efficient few-shot adaptation via discrepancy-based evidence modeling.
- **Experiments.** We conduct extensive experiments to demonstrate the strong performance of GenGLAD, and illustrate its generalizability and efficiency across 10 benchmark datasets from various application domains.

2 Related Work

Graph-Level Anomaly Detection aims to identify anomalous graphs that significantly deviate from the majority of normal graphs in a dataset [24, 43]. Existing methods can be broadly categorized into two groups, namely, reconstruction-based methods and contrastive learning-based methods. The former aim to reconstruct input graphs and detect anomalies based on reconstruction errors, assuming that anomalous graphs are harder to reconstruct [22, 5, 14, 55]. The latter learn graph representations by contrasting multiple views of the data, using consistency or agreement between views to distinguish normal patterns from anomalies [15, 18, 50, 11]. There are also several studies that leverage advanced techniques, such as evolutionary mapping [25], mixture-of-experts [4], and knowledge distillation [23], to address the GLAD problem. Existing GLAD methods usually follow a one-dataset-one-model paradigm, where models are trained using either purely normal data (one-class unsupervised setting [43]) or labeled normal and anomalous data (supervised setting [52, 25, 47]), and evaluated on the same dataset. This paradigm increases the requirement for large amounts of labeled or curated data and limits its broad applicability.

Generalist Anomaly Detection is an emerging paradigm that aims to train a universal anomaly detection model on diverse datasets and generalize it to unseen datasets with minimal supervision and cost [58, 31]. Recently, generalist anomaly detection approaches have achieved significant progress and demonstrated strong performance across various domains, such as industrial image data [58], time-series data [37], relational data [17], and tabular data [16]. In text data, such techniques can be utilized to spot hallucination [33, 6] or fraudsters [30]. Recent visual approaches further enhance generalization through residual feature modeling [51], normal-abnormal guidance [44], and residual-guided normality learning [42]. For graph data, recent studies investigate generalist approaches for node-level anomaly detection [19, 34, 20]. While these methods have shown promising performance, they cannot be directly applied to GLAD due to fundamental differences between node-level and graph-level abnormal patterns: node-level methods typically rely on several assumptions (e.g., the homophily assumption [29]), which may not hold in the graph-level counterpart [32].

A more comprehensive and detailed literature review is given in Appendix A.

3 Problem Statement

Notations. In general, a graph can be presented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, which can also be presented in adjacency matrix form $\mathbf{A} \in \{0,1\}^{n \times n}$, with $\mathbf{A}_{i,j} = 1$ iff $(v_i, v_j) \in \mathcal{E}$. For a node $v \in \mathcal{V}$, $\mathcal{N}(v)$ denotes its one-hop neighborhood and $\deg(v) = |\mathcal{N}(v)|$ its degree.

Conventional GLAD Problem. GLAD aims to identify anomalous graphs from a collection of graphs by learning an anomaly scoring function $f : \mathcal{G} \to \mathbb{R}$ where higher scores correspond to greater degrees of abnormality. The function f is optimized on the training split of target datasets $D = \{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$. The mainstream studies of GLAD usually follow an one-class (sometimes a.k.a.

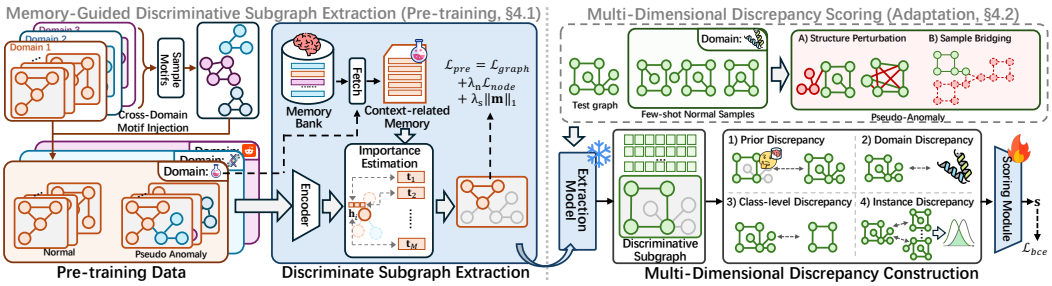


Figure 2: The overall framework of GenGLAD. During the pre-training phase, a discriminative subgraph extraction model learns the transferable patterns for generalist GLAD. During adaptation, multi-dimensional discrepancy is constructed for anomaly scoring on target dataset. Code available at https://github.com/CampanulaBells/Generalist-Graph-Level-Anomaly-Detection.

unsupervised) setting, where the model is trained on a collection of normal graphs and is then applied to detect anomalies in previously unseen graphs. The inductive property makes GLAD challenging, as anomalous patterns are not observed during training.

Generalist GLAD Problem. In this paper, we study the *generalist GLAD* problem, which aims to develop a generalist GLAD model capable of detecting anomalous graphs across diverse application scenarios, with only lightweight fine-tuning on few-shot normal samples from target dataset. Formally, we define $\mathcal{T}^{\text{train}} = \{\mathcal{D}_1^{\text{train}}, \ldots, \mathcal{D}_N^{\text{train}}\}$ as a collection of training datasets, where each dataset $\mathcal{D}_i^{\text{train}} = \{G_{i,j}^{\text{train}}\}_{j=1}^{N_i}$ consists of N_i normal graphs from a particular application scenario. The goal is to train a generalist GLAD model f on $\mathcal{T}^{\text{train}}$ such that it can effectively detect anomalies in unseen test datasets $\mathcal{T}^{\text{test}}$, where $\mathcal{D}_i^{\text{test}} \in \mathcal{T}^{\text{test}}$. During inference, we adopt the commonly used *normal few-shot* setting [19, 34, 54, 58]. Specifically, for each test dataset $\mathcal{D}_i^{\text{test}}$, only a small number ($n_k$) of normal graphs are available for lightweight adaptation. The model f is then required to assign anomaly scores to the remaining graphs in the dataset. In this paper, we assume that all graphs are unattributed to cover a wide range of real-world scenarios, because ❶ data from several domains do not come with node or edge features [43], and ❷ node features may play a less critical role in GLAD [48, 3].

4 Method

In this section, we introduce GenGLAD, a generalist GLAD approach that can identify anomalous graph samples on arbitrary testing datasets with few-shot adaptation. As shown in Figure 2, the pre-trained model in GenGLAD learns to *extract discriminative subgraphs* for anomaly discrimination, with a *memory bank* to capture transferable knowledge and a *cross-domain motif injection strategy* to generate pseudo-anomalies (Sec. 4.1). During adaptation phase, GenGLAD leverages few-shot normal samples to compute multi-dimensional discrepancy for anomaly detection, with a lightweight scoring module for final prediction (Sec. 4.2).

4.1 Memory-Guided Discriminative Subgraph Extraction

To establish a generalist model, a fundamental challenge is to capture generalizable patterns across diverse graph domains for normal-anomaly discrimination. However, such patterns can vary significantly across domains due to differences in graph structures and distributions, making it difficult to identify a specific type of pattern as transferable knowledge. To address this challenge, instead of relying on a specific type of pattern, we propose a generalizable graph neural network (GNN) model that learns to **extract the most discriminative substructures** from arbitrary graph samples across diverse domains, serving as the transferable backbone of GenGLAD (Sec. 4.1.1). The discriminative subgraph can serve as key clues for distinguishing normal and abnormal samples, facilitating downstream discrimination. To capture domain-specific knowledge during subgraph extraction, we introduce a memory bank to store transferable patterns, supporting subgraph extraction with domain-relevant information (Sec. 4.1.2). To optimize the model using normal-only data, we propose a cross-domain motif injection strategy to generate pseudo-anomalous samples for training, together with a tailored objective function to guide the learning process (Sec. 4.1.3).

4.1.1 GNN-based Subgraph Extractor

While anomaly patterns across different domains can be diverse, the abnormality of anomalous graphs is usually characterized by a few key substructures [18, 50]. For example, in molecular graphs, abnormality (e.g., chemical instability) can often be attributed to specific functional groups, while in social networks, anomalous behavior may be associated with unusual local irregular interaction motifs. Motivated by this, we propose a GNN backbone model that identifies the most discriminative subgraphs from various graph samples, learning the transferable patterns for GLAD across diverse domains. This capability is generalizable to unseen graphs for anomaly discrimination [25].

Structural Attribute Construction. As we focus on a more general unattributed setting, the first step is to construct a unified and semantically rich node attributes as the input features for the GNN. To this end, we incorporate structural features as node attributes. Specifically, for each graph $\mathcal{G}$, node attributes are constructed by concatenating one-hot encodings of node degrees with random-walk diffusion-based structural encodings [10], i.e, $\mathbf{X} = \mathbf{X}^{\text{deg}} \oplus \mathbf{X}^{\text{RW}}$. This design not only unifies the dimensionality and semantics of node attributes across datasets but also equips them with informative structural signals, thereby providing a consistent and effective foundation for generalist GLAD.

Subgraph Extraction Network. Given an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with unified node attributes $\mathbf{X}$, we aim to extract a discriminative subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ that preserves anomaly-discriminative information while filtering out irrelevant structures. To achieve this, we first obtain node representations $\mathbf{H}$ via a graph encoder, and then compute the node importance scores $\mathbf{p}$ through an importance estimation module $f_\theta(\cdot)$:

$$\mathbf{H} = \text{GNN}(\mathbf{X}, \mathbf{A}), \quad \mathbf{p} = f_\theta(\mathbf{H}). \tag{1}$$

To extract the task-relevant subgraph while obtaining a differentiable node selection, we then apply Gumbel-Softmax with temperature τ to transform the importance scores into a binary mask:

$$\mathbf{m} = \text{GumbelSoftmax}(\mathbf{p}; \tau), \quad \mathbf{m} \in \{0, 1\}^{|\mathcal{V}|}. \tag{2}$$

The resulting mask is used to filter nodes and edges, yielding the extracted subgraph $\mathcal{G}'$. This subgraph focuses the model on the most informative regions of the input graph while ignoring the irrelevant structures, serving as the discriminative evidence for downstream detection.

4.1.2 Importance Estimation via Memory Retrieval

In the subgraph extraction network, it is important to ensure that $f_\theta(\cdot)$ consistently identifies the most discriminative subgraphs. However, what constitutes such discriminative structures is inherently domain-dependent. Due to the diversity of application scenarios, topological patterns may vary significantly across domains, implying that the estimation module $f_\theta(\cdot)$ must incorporate domain-specific knowledge. To improve the domain adaptability of $f_\theta(\cdot)$, we equip it with a memory bank that encodes anomaly-related knowledge accumulated from diverse training domains. Given a new dataset, we can retrieve domain-relevant memory conditioned on its target context, thereby guiding the extraction process.

Specifically, we maintain a set of M trainable memory tokens $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_M\}$, where each token is responsible for identifying a discriminating graph pattern associated with anomaly detection across datasets. Then, given a target dataset $\mathcal{D}$ with available graph samples $\mathcal{S} \subset \mathcal{D}_i$, we summarize a compact context vector that serves as the condition for retrieval: $\mathbf{c} = \frac{1}{\|\mathcal{S}\|} \sum_{\mathcal{G} \in \mathcal{S}} \text{Readout}_{\mathcal{G}}(\mathbf{X})$, with mean pooling operation for readout. Here, $\mathcal{S}$ can be a batch of normal sample during training, or the few-shot supervision during the adaptation stage. The obtained context vector $\mathbf{c}$ is then used to adaptively retrieve relevant knowledge from the memory bank. In detail, we choose the tokens with top-K relevance by computing cosine similarity between $\mathbf{c}$ and memory tokens, resulting in the retrieved token set $\tilde{\mathbf{T}}$ which encodes the domain-specific knowledge. Finally, the node importance is estimated by measuring the alignment between node embeddings and the retrieved memory tokens:

$$\mathbf{p}_i = \max_{\mathbf{t} \in \tilde{\mathbf{T}}} \cos(\mathbf{h}_u, \mathbf{t}), \tag{3}$$

where $\mathbf{h}_u$ denotes the encoded embedding of node u in $\mathcal{G}$, and $\cos(\cdot, \cdot)$ is the cosine similarity. By obtaining the maximum similarity score, we capture the most significant aligned pattern, ensuring the extracted subgraph is discriminative. In this way, the memory bank supplies transferable knowledge, while context-conditioned retrieval ensures domain-specific subgraph extraction.

4.1.3 Model Training

To train a generalist model for GLAD, a straightforward but effective learning objective is teaching it to distinguish between anomalous and normal samples. However, annotating abnormal graph samples demands heavy domain expertise, making the training data for GLAD usually under a one-class assumption, i.e., only normal samples are available [18]. To optimize the generalist model in such an unsupervised pre-training setting, we generate pseudo-anomalies via a *cross-domain motif injection* strategy, and leverage them to guide model training with a bi-level loss function.

Cross-Domain Motif Injection. To generate pseudo-anomalies for training datasets from diverse domains, we propose a cross-domain motif injection strategy that leverages motifs from one domain to perturb graphs in another. The detailed procedure can be found at Appendix B. Concretely, given a batch of graphs sampled from a training dataset $\mathcal{D}_i^{\text{train}} \in \mathcal{T}^{\text{train}}$, we treat these graphs as normal samples for the corresponding graph application, allocated with pseudo label $\tilde{y} = 0$. Pseudo-anomalies ($\tilde{y} = 1$) are then constructed by incorporating cross-dataset substructure, where each normal graph is augmented with a motif randomly sampled from graphs drawn from other datasets. As such subgraphs originate from different data distributions, they can disrupt the inherent structural patterns of the original graph, naturally introducing anomalous signals. To merge the graph and the motif, we link them with additional edges to form a single connected graph. To further provide fine-grained supervision signals, we assign a node-level pseudo-label vector $\tilde{\mathbf{y}}^{(n)} \in \{0,1\}^{|\tilde{\mathcal{V}}_b|}$ for each graph, where each entry indicates whether a node is injected from a motif: $\tilde{\mathbf{y}}_u^{(n)} = 1$ if node u belongs to any injected motif, and $\tilde{\mathbf{y}}_u^{(n)} = 0$ otherwise.

Bi-Level Training Loss. To ensure the model can distinguish anomalies across diverse scenarios, we train the model by a pseudo-label-guided bi-level loss. We first formulate a **graph-level objective** that encourages the extracted subgraph $\mathcal{G}'$ to be semantically consistent with domain-specific knowledge while deviating from anomalous ones. Given that the memory tokens encode prototypical discriminative patterns, we compute the similarity between the graph-level representation $\mathbf{z}_{\mathcal{G}'} = \text{Readout}_{\mathcal{G}'}(\mathbf{H})$ and the aggregated in-domain memory token $\mathbf{t_c} = \text{Mean}(\mathbf{T}')$, and optimize it with a binary cross-entropy (BCE) objective:

$$\mathcal{L}_{\text{graph}} = \text{BCE}\big(\sigma(1 - \cos(\mathbf{z}_{\mathcal{G}'}, \mathbf{t_c})),\, \tilde{y}\big), \tag{4}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity and $\sigma(\cdot)$ is a sigmoid function. This objective encourages the extracted in-domain knowledge to be aligned with normal graphs while diverging from anomalies, thereby promoting cross-domain generalization.

To provide node-level supervisions, we leverage the injected motifs to form **node-level objective**. Specifically, we utilize the labeled injected nodes as hard negatives and penalize false positive predictions, encouraging the model to exclude them from the learned discriminative subgraph:

$$\mathcal{L}_{\text{node}} = -\mathbb{E}_{i \sim \tilde{\mathbf{y}}_i^{(n)}=1}[\log(1 - m_i)]\,. \tag{5}$$

Finally, we integrate the graph- and node-level objectives into a unified pre-training loss:

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{graph}} + \lambda_{\text{n}}\mathcal{L}_{\text{node}} + \lambda_{\text{s}}\|\mathbf{m}\|_1, \tag{6}$$

where the sparsity regularization term $\|\mathbf{m}\|_1$ encourages the model to focus on compact and informative substructures, while λ_{n} and λ_{s} are balance hyperparameters. Overall, the training objective utilizes the informative pseudo-labels to guide the learning of GenGLAD with both coarse-grained and fine-grained losses, grounding robust generalization across diverse graph application domains.

4.2 Multi-Dimensional Discrepancy Scoring for Few-Shot Adaptation

Although the pre-trained backbone model has the capability to capture informative subgraphs that are useful for anomaly discrimination, simply applying it to unseen datasets may not always generalize well due to domain shift and variations in abnormality patterns. To achieve effective target-dataset adaptation, a key idea is to fully exploit the knowledge encoded in the pre-trained model and the memory bank, together with the few-shot samples that reflects the characteristics of the target domain. To this end, we construct multi-dimensional discrepancy signals that characterize the deviation of the discriminative subgraph of an incoming sample from multiple perspectives, including prior knowledge, domain-level context, and feature semantics. To adaptively leverage these discrepancy

signals for anomaly detection, we further learn a lightweight scoring module with the few-shot normal samples, ensuring reliable predictions are made based on the discrepancy-based evidence.

Multi-Dimensional Discrepancy Construction. While the extracted subgraph captures key characteristics of each graph sample, it is not straightforward to characterize its position within the normal distribution of the target dataset for anomaly detection. To achieve this, we model anomaly evidence as discrepancy scores across multiple semantic spaces. We construct a set of complementary discrepancy signals for each incoming graph, providing a holistic characterization of it. Specifically, we construct multi-dimensional discrepancy measures from four perspectives.

1. **Prior Discrepancy**, which measures the deviation of the discriminative subgraph embedding $\mathbf{z}_{\mathcal{G}'}$ from the pre-trained semantic prior, characterized by its similarity to the pre-trained prototype $\mathbf{t_c}$. This perspective captures how well the current graph aligns with the general knowledge learned during pre-training, serving as a basic indicator for anomaly detection.
2. **Domain Discrepancy**, which measures the deviation of $\mathbf{z}_{\mathcal{G}'}$ from the domain-level context, characterized by its distance to the global domain prototype $\mathbf{c}$. This perspective reflects the extent to which the current graph deviates from the overall data distribution, providing a coarse-grained signal of distributional shift and potential anomaly.
3. **Class-level Discrepancy**, which evaluates the mismatch between $\mathbf{z}_{\mathcal{G}'}$ and the class prototype $\mathbf{q}$ calculated as the average representations of few-shot normal samples. This perspective characterizes how well the current graph adheres to the semantic pattern of the normal class, offering a more targeted cue for distinguishing abnormal instances.
4. **Instance-level Discrepancy**, which captures the inconsistency between the similarity pattern of the query graph to support instances and the intrinsic relational structure within the support set. This perspective reveals whether the current graph maintains the fine-grained relational regularities of normal data, yielding a distribution-aware signal beyond point-wise comparisons.

Notably, the choice of similarity/distance metrics (e.g., cosine and L2) can influence the characterization of discrepancies; therefore, we adopt multiple metrics for certain discrepancies to improve coverage. A formal formulation of the discrepancy scores is provided in Appendix C.

Lightweight Scoring Module. In practice, different datasets may exhibit varying dependence on different discrepancy scores for anomaly detection, due to differences in data distributions. To account for this, we learn a lightweight scoring module, which adopts a linear combination of discrepancy scores: $s = \sigma(\mathbf{w}^\top \mathbf{s})$, where s is the anomaly score of a query graph, $\mathbf{s}$ is the vector composed of multiple discrepancy scores, and σ is a Sigmoid function. $\mathbf{w} \in \mathbb{R}^{k\times 1}$ is the only parameter that is learnable during the adaptation phase, while the rest of the model remains frozen.

To optimize $\mathbf{w}$, a straightforward approach is to use a binary cross-entropy loss with binary normal/anomalous labeled samples. Nevertheless, the few-shot set only contains normal instances, leading to an urgent need for a mechanism to distinguish pseudo-anomalies from normal samples. As large-scale training datasets are not available at this stage, we employ two simple yet effective strategies to distinguish pseudo-anomalies from few-shot normal samples (details of pseudo-anomaly construction is given in Appendix D): ❶ **Structure perturbation**, which constructs outlier graphs by randomly shuffling graph structures to destroy meaningful relational patterns, and ❷ **Sample bridging**, which generates outlier graphs by merging normal graphs to simulate over-amplified in-domain structural patterns. With the original samples as normal instances and pseudo-anomalies as negative instances, we can optimize $\mathbf{w}$ with only a few gradient steps during adaptation phase. Algorithm description and complexity analysis of GenGLAD are in Appendices E and F.

5 Experiments

5.1 Experimental Setup

Datasets. We train both the baseline approaches and GenGLAD on one set of graph datasets and evaluate them on a separate set of datasets. For a comprehensive evaluation, we utilize real-world GLAD datasets drawn from diverse domains, including molecular graphs, biochemical graphs, and social networks. Following [19], we train the models on the largest dataset within each domain and evaluate them on the remaining datasets. Specifically, the training datasets include NCI1, DD, and COLLAB, while the testing datasets consist of molecular graphs: AIDS, BZR, COX2, DHFR, MUTAG, PTC_MR; biochemical graphs: ENZYMES, PROTEINS; and social networks: IMDB-BINARY, and REDDIT-BINARY. We also denote the collection of molecular and chemical graphs as science graphs. For more detailed information, please refer to the Appendix G.

Table 1: Anomaly detection performance in terms of AUROC (in percent, mean±std). Highlighted are the results ranked **first**, second, and third. OOM indicates out-of-memory.

Method	AIDS	BZR	COX2	DHFR	MUTAG	PTC_MR	ENZYMES	PROTEINS	IMDB-B	REDDIT-B
Pre-training+Few-shot Fine-tuning										
GLocalKD	50.67±0.12	50.29±0.04	52.73±0.07	56.90±0.11	79.55±0.06	49.34±0.03	50.12±0.06	49.90±0.06	67.82±0.23	67.82±0.23
CVTGAD	40.65±0.00	45.96±0.01	52.00±0.03	55.81±0.00	89.55±0.01	51.83±0.01	45.31±0.00	40.36±0.71	OOM	OOM
SIGNET	90.59±0.00	21.46±0.02	53.56±0.01	53.56±0.01	82.47±0.02	53.19±0.02	47.28±0.00	57.19±0.01	50.66±0.00	50.66±0.00
MUSE	77.30±1.73	52.77±0.42	45.27±0.27	57.24±0.88	84.05±1.67	47.32±0.33	52.48±0.47	52.57±1.25	67.84±1.06	67.84±1.06
GLADPro	50.00±0.00	47.39±1.72	48.10±0.69	53.04±1.84	50.00±0.00	53.33±3.61	53.13±0.63	38.51±1.36	64.16±7.36	64.16±7.36
HC-GLAD	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
FANFOLD	73.54±0.00	50.36±0.01	37.94±0.02	58.18±0.00	87.89±0.02	41.70±0.01	49.50±0.00	49.41±0.00	56.64±0.00	18.51±0.00
Few-shot Fine-tuning Only										
GLocalKD	57.71±2.91	58.91±4.54	52.14±2.39	63.34±1.54	89.54±0.23	48.83±0.99	47.60±2.08	53.05±2.89	59.79±1.63	72.37±1.56
CVTGAD	69.49±7.78	43.30±3.80	43.36±3.96	46.24±1.47	79.41±8.93	43.29±2.79	54.50±2.90	69.81±0.86	55.51±4.79	OOM
SIGNET	68.21±5.49	44.60±3.52	44.38±3.02	35.21±3.12	68.86±14.45	50.93±0.55	41.28±2.02	55.48±1.78	45.97±2.37	52.36±1.95
MUSE	84.56±5.82	47.32±0.87	41.80±1.35	61.74±1.91	85.77±6.01	50.76±0.47	49.50±0.33	61.81±0.17	58.18±2.38	73.85±1.82
GLADPro	70.21±13.92	56.24±3.37	43.62±3.80	43.03±3.72	82.96±8.10	55.73±3.05	53.98±1.45	43.44±4.09	54.01±2.89	64.29±6.07
HC-GLAD	3.64±0.55	37.50±3.32	46.16±1.56	54.99±2.66	84.69±2.79	48.94±0.70	63.03±4.45	37.85±3.67	61.85±5.11	OOM
FANFOLD	60.95±18.89	44.92±1.04	41.80±3.84	60.28±2.35	74.00±20.09	47.04±3.60	52.15±3.95	58.24±3.61	51.54±3.97	23.24±2.92
Generalist										
GenGLAD	95.84±0.07	63.89±1.18	62.92±0.30	74.96±0.42	90.75±0.17	60.58±0.47	54.37±1.74	73.63±0.16	74.11±0.25	73.63±0.20

Table 2: Performance (in AUROC) with varying pre-training datasets $\mathcal{T}^{\text{train}}$.

Training Datasets	AIDS	BZR	COX2	DHFR	MUTAG	PTC_MR	ENZYMES	PROTEINS	IMDB-B	REDDIT-B
NCI1	87.56±2.68	56.64±0.76	57.72±1.27	70.92±0.60	75.19±3.45	55.83±1.46	52.70±0.45	63.26±1.45	56.14±1.20	64.06±0.16
DD	88.33±4.26	56.98±1.25	57.89±1.39	70.87±0.13	73.74±2.93	55.83±1.46	52.82±0.26	63.20±1.40	55.80±1.09	63.46±1.04
COLLAB	87.88±0.50	57.48±1.67	57.74±1.76	71.09±0.25	76.25±1.69	55.64±1.43	52.52±0.32	63.30±1.15	55.15±1.38	65.44±0.48
NCI1+DD	89.72±2.71	58.18±1.52	58.23±1.53	71.10±0.50	80.08±3.99	57.70±1.42	53.19±0.54	65.37±1.09	66.77±1.22	70.96±1.09
NCI1+COLLAB	91.62±3.38	59.92±1.10	58.22±1.46	72.51±0.89	82.55±2.64	57.63±1.40	53.49±0.14	67.04±0.86	66.62±1.76	69.22±1.28
DD+COLLAB	91.99±2.94	59.99±1.06	58.55±1.32	72.85±0.80	83.09±2.62	57.32±1.21	53.76±0.21	67.33±1.13	68.59±1.44	65.34±0.40
All	95.84±0.07	63.89±1.18	62.92±0.30	74.96±0.42	90.75±0.17	60.58±0.47	54.37±1.74	73.63±0.16	74.11±0.25	73.63±0.20

Baselines. We compare GenGLAD with several unsupervised GLAD baselines, including contrastive methods such as GLocalKD [23], GLADPro [50], CVTGAD [15], SIGNET [18], and HC-GLAD [11]; reconstruction-based methods such as FANFOLD [5] and MUSE [14].

Evaluation and Implementation. Following [43], we adopt AUROC and AUPRC as evaluation metrics for GLAD. We report the mean AUROC/AUPRC along with standard deviations over five independent runs. We train GenGLAD jointly on all datasets in $\mathcal{T}^{\text{train}}$ and evaluate it on each dataset in $\mathcal{T}^{\text{test}}$ with few-shot tuning, where $n = 10$ normal graphs are provided as supervision by default. For all baselines, we consider two settings: "Pre-training+Few-shot Fine-tuning" and "Few-shot Fine-tuning Only". In the latter, baselines are trained directly using only the few-shot normal supervision to avoid potential performance degradation caused by negative transfer. We utilize random search to determine the optimal hyperparameters for both the baselines and GenGLAD (Details in Appendix G). Since our goal is to train generalist GLAD models, we do not perform dataset-specific hyperparameter tuning, but instead use the same set of hyperparameters for all testing datasets.

5.2 Experimental Results

Performance Comparison.

Table 1 presents a comparison between GenGLAD and baseline methods in terms of AUROC. Results for AUPRC are provided in Appendix H.1. We make the following observations: ❶ GenGLAD demonstrates strong anomaly detection capability in the generalist GLAD setting, despite using only few-shot fine-tuning on normal samples. Specifically, GenGLAD achieves state-of-the-art performance on 9 out of 10 datasets, while still obtaining competitive performance on the ENZYMES dataset. Moreover, GenGLAD demonstrates significant improvements compared to the best baselines on AIDS, REDDIT, COX2, and DHFR. ❷ Pre-training typically does not lead to performance improvements for GLAD baselines, highlighting the limited generalizability of conventional GLAD methods. ❸ With a limited amount of training data, baseline methods perform poorly. Notably, in some cases, they achieve an AUROC below 0.5 (worse than random guessing). This highlights that conventional GLAD methods are data-hungry, making them difficult to apply in emerging GLAD scenarios where training data is scarce.

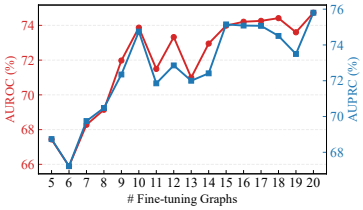


Figure 3: Sensitivity of #shot.

Effectiveness of Data Scale. To investigate the impact of data scale on the performance of GenGLAD, we vary $\mathcal{T}^{\text{train}}$. From Table 2, we observe that the performance of GenGLAD consistently

improves as more pre-training datasets are incorporated, demonstrating its ability to learn transferable patterns across diverse domains. With the help of memory retrieval, the model can further retrieve domain-relevant knowledge, mitigating the negative transfer issue. Moreover, we report the average performance on social networks (Figure 3, more results are in Appendix H.2) with varying number of few-shot samples for adaptation. We observe that increasing the number of fine-tuning graphs consistently improves performance, which highlights the effectiveness of our method in leveraging additional adaptation data when available.

Ablation Study. To verify the effectiveness of each component of GenGLAD, we conduct ablation studies by removing/modifying each design, with the results on selected datasets in Table 3 (full results are in Appendix H.3). Specifically, for the pre-training phase, we construct variants by removing $\mathcal{L}$node and the ℓ_1 regularization on $\mathbf{m}$. We find that both components contribute to overall performance. In particular, $\mathcal{L}$node has a larger impact, highlighting the importance of informative node-level pseudo anomaly supervision compared to unsupervised regularization. Moreover, we investigate the effectiveness of each discrepancy measure in the fine-tuning stage and the scoring module. Generally, removing any of the discrepancy measures degrades the performance in most cases. Among them, the domain and class discrepancy scores have the most significant impact, as their removal can lead to performance close to or even below random guessing. To assess the necessity of the scoring module, we replace it with simple score aggregation strategies, including mean, max, and min pooling. As illustrated, these replacements significantly degrade performance and can even lead to AUROC scores below 0.5. This suggests that the model fails to distinguish between normal and anomaly-indicative cues even with normal graph supervisions, indicating the importance of our adaptation framework.

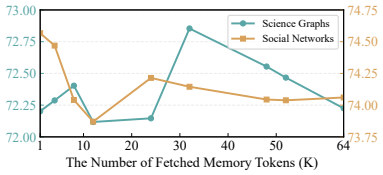
Table 3: Ablation study.

Variant	DHFR	PROTEINS	IMDB-B
GenGLAD	74.96	73.63	74.11
Pre-training			
w/o $\mathcal{L}^{\text{node}}$	70.98	66.03	58.03
w/o $\|\mathbf{m}\|_1$	71.30	68.53	60.06
Discrepancy Measures			
w/o Prior	73.05	71.66	71.15
w/o Domain	73.77	63.76	51.65
w/o Cls.	45.63	60.20	63.72
w/o Inst.	74.25	73.75	72.96
Scoring Module			
MEAN	28.92	41.54	42.00
MAX	40.32	61.94	30.12
MIN	29.56	38.05	43.57

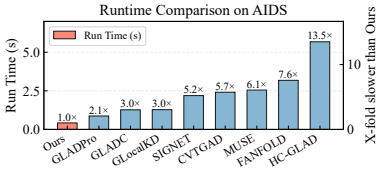
Effectiveness of #Retrieved Tokens. To verify the effectiveness of retrieved memory tokens, we study the effect of retrieved memory token number K on performance (AUROC) in Figure 4a. Overall, both groups of datasets exhibit a consistent trend: after a certain threshold, increasing K leads to performance degradation, as less relevant tokens may introduce noisy or unhelpful information for the target task. Moreover, scientific graphs and social networks exhibit different sensitivities to K, where scientific graphs prefer larger K for diverse motifs, while social networks favor smaller K.

Efficiency Analysis. To investigate the efficiency of GenGLAD, we compare fine-tuning time on the AIDS dataset. As shown in Figure 4b, GenGLAD outperforms all baselines in terms of efficiency. Compared with our method, the fastest baseline, GLADPro, is still 2.1× slower. A comprehensive runtime breakdown is further discussed in Appendix H.5.

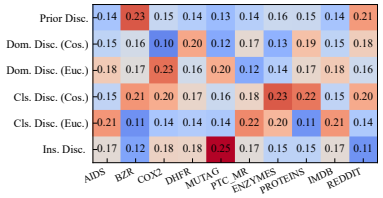
Visualization. To investigate how the pre-trained GenGLAD adapts to target applications, we visualize the weights of the scoring module. As shown in Figure 4c, GenGLAD learns distinct weighting patterns across different test domains. This indicates that the scoring module successfully captures domain-specific dependencies, enabling effective yet efficient few-shot adaptation. We also visualized the importance scores of individual memory tokens in Appendix H.4.



(a) Sensitivity to # mem. tokens.



(b) Time comparison.



(c) Visualization results.

Figure 4: Analysis results.

6 Conclusion

In this paper, we take a pioneering step toward addressing the generalist GLAD problem, which aims to detect anomalous graphs across diverse emerging applications under a one-for-all paradigm with only a few-shot normal graph supervision for adaptation. To this end, we propose GenGLAD, a discriminative subgraph extraction-based generalist GLAD approach capable of identifying anomalous graphs with limited in-domain supervision. Extensive experiments on real-world GLAD datasets from various domains demonstrate the effectiveness, generalizability, and efficiency of GenGLAD. The main **limitation** of GenGLAD is that it still requires few-shot supervision for adaptation and therefore cannot yet achieve fully zero-shot generalist GLAD. **Future work** may explore incorporating expert knowledge or external knowledge bases to enable zero-shot GLAD across diverse domains.

Acknowledgments and Disclosure of Funding

The work of S. Pan was partially supported by the Australian Research Council (ARC) under Grant Nos. DP240101547 and FT210100097. The work of Y. Liu was partially supported by the ARC under Grant No. DE260101172. F. Li was supported by the Australian National Health and Medical Research Council Investigator grants [2041439].

References

[1] Lemar Abdi, Amaan Valiuddin, Francisco Caetano, Christiaan Viviers, and Fons Van Der Sommen. Zero-shot image anomaly detection using generative foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3554–3563, 2025.

[2] Charu C Aggarwal and Haixun Wang. Graph data management and mining: A survey of algorithms and applications. In *Managing and mining graph data*, pages 13–68. Springer, 2010.

[3] Jinyu Cai, Yunhe Zhang, Zhoumin Lu, Wenzhong Guo, and See-Kiong Ng. Towards effective federated graph anomaly detection via self-boosted knowledge distillation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5537–5546, 2024.

[4] Jinyu Cai, Yunhe Zhang, Pengyang Wang, and See-Kiong Ng. Moegad: A mixture-of-experts framework with pseudo-anomaly generation for graph-level anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[5] Rui Cao, Shijie Xue, Jindong Li, Qi Wang, and Yi Chang. Fanfold: Graph normalizing flows-driven asymmetric network for unsupervised graph-level anomaly detection. *arXiv preprint arXiv:2407.00383*, 2024.

[6] Baiyu Chen, Wilson Wongso, Xiaoqian Hu, Yue Tan, and Flora D Salim. Multi-stage verification-centric framework for mitigating hallucination in multi-modal rag. In *2025 KDD Cup Workshop for Multimodal Retrieval Augmented Generation*, 2026.

[7] Xuhai Chen, Yue Han, and Jiangning Zhang. A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 (vand) workshop challenge tracks 1 &2. *1st Place on Zero-shot AD and 4th Place on Few-shot AD*, 2305:17382, 2023.

[8] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991.

[9] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

[10] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

[11] Yali Fu, Jindong Li, Jiahong Liu, Qianli Xing, Qi Wang, and Irwin King. Hc-glad: dual hyperbolic contrastive learning for unsupervised graph-level anomaly detection. *Neural Networks*, page 109009, 2026.

[12] Gastón García González, Pedro Casas, Emilio Martínez, and Alicia Fernández. Towards foundation auto-encoders for time-series anomaly detection. *arXiv preprint arXiv:2507.01875*, 2025.

[13] Christoph Helma, Ross D. King, Stefan Kramer, and Ashwin Srinivasan. The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17(1):107–108, 2001.

[14] Sunwoo Kim, Soo Yong Lee, Fanchen Bu, Shinhwan Kang, Kyungho Kim, Jaemin Yoo, and Kijung Shin. Rethinking reconstruction-based graph-level anomaly detection: limitations and a simple remedy. *Advances in Neural Information Processing Systems*, 37:95931–95962, 2024.

[15] Jindong Li, Qianli Xing, Qi Wang, and Yi Chang. Cvtgad: Simplified transformer with cross-view attention for unsupervised graph-level anomaly detection. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 185–200. Springer, 2023.

[16] Shiyuan Li, Yixin Liu, Yu Zheng, Xiaofeng Cao, Shirui Pan, and Heng Tao Shen. Towards one-for-all anomaly detection for tabular data. *arXiv preprint arXiv:2603.14407*, 2026.

[17] Shiyuan Li, Yunfeng Zhao, Yue Tan, Qingfeng Chen, Yixin Liu, and Shirui Pan. Towards anomaly detection on relational data. *arXiv preprint arXiv:2606.18621*, 2026.

[18] Yixin Liu, Kaize Ding, Qinghua Lu, Fuyi Li, Leo Yu Zhang, and Shirui Pan. Towards self-interpretable graph-level anomaly detection. *Advances in Neural Information Processing Systems*, 36:8975–8987, 2023.

[19] Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan. Arc: A generalist graph anomaly detector with in-context learning. *Advances in Neural Information Processing Systems*, 37:50772–50804, 2024.

[20] Yujing Liu, Yixin Liu, Yu Zheng, Alan Wee-Chung Liew, Xiaofeng Cao, and Shirui Pan. Rethinking feature alignment in generalist graph anomaly detection: A relational fingerprint-based approach. In *International Conference on Machine Learning*, 2026.

[21] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[22] Xuexiong Luo, Jia Wu, Jian Yang, Shan Xue, Hao Peng, Chuan Zhou, Hongyang Chen, Zhao Li, and Quan Z Sheng. Deep graph level anomaly detection with contrastive learning. *Scientific Reports*, 12(1):19867, 2022.

[23] Rongrong Ma, Guansong Pang, Ling Chen, and Anton Van Den Hengel. Deep graph-level anomaly detection by glocal knowledge distillation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 704–714, 2022.

[24] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE transactions on knowledge and data engineering*, 35(12):12012–12038, 2021.

[25] Xiaoxiao Ma, Jia Wu, Jian Yang, and Quan Z Sheng. Towards graph-level anomaly detection via deep evolutionary mapping. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 1631–1642, 2023.

[26] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

[27] Chaoxi Niu, Guansong Pang, and Ling Chen. Graph-level anomaly detection via hierarchical memory networks. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 201–218. Springer, 2023.

[28] Junjun Pan, Yixin Liu, Rui Miao, Kaize Ding, Yu Zheng, Quoc Viet Hung Nguyen, Alan Wee-Chung Liew, and Shirui Pan. Explainable and fine-grained safeguarding of llm multi-agent systems via bi-level graph anomaly detection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30491–30506, 2026.

[29] Junjun Pan, Yixin Liu, Xin Zheng, Yizhen Zheng, Alan Wee-Chung Liew, Fuyi Li, and Shirui Pan. A label-free heterophily-guided approach for unsupervised graph fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12443–12451, 2025.

[30] Junjun Pan, Yixin Liu, Yu Zheng, Lianhua Chi, Alan Wee-Chung Liew, and Shirui Pan. Camera: Adapting to semantic camouflage in unsupervised text-attributed graph fraud detection. In *International Joint Conference on Artificial Intelligence*, 2026.

[31] Junjun Pan, Yixin Liu, Chuan Zhou, Fei Xiong, Alan Wee-Chung Liew, and Shirui Pan. Correcting false alarms from unseen: Adapting graph anomaly detectors at test time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24691–24699, 2026.

[32] Junjun Pan, Yu Zheng, Yue Tan, and Yixin Liu. A Survey of Generalization of Graph Anomaly Detection: From Transfer Learning to Foundation Models . In *2025 IEEE International Conference on Knowledge Graph (ICKG)*, pages 316–323, Los Alamitos, CA, USA, November 2025. IEEE Computer Society.

[33] Yanyu Qian, Yue Tan, Yixin Liu, Wang Yu, and Shirui Pan. Dynhd: Hallucination detection for diffusion large language models via denoising dynamics deviation learning. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026.

[34] Hezhe Qiao, Chaoxi Niu, Ling Chen, and Guansong Pang. 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*, pages 2326–2337, 2025.

[35] Chen Qiu, Marius Kloft, Stephan Mandt, and Maja Rudolph. Raising the bar in graph-level anomaly detection. In *International Joint Conference on Artificial Intelligence*, 2022.

[36] Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl_1):D431–D433, 2004.

[37] Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *The Thirteenth International Conference on Learning Representations*, 2025.

[38] Jeffrey J Sutherland, Lee A O'brien, and Donald F Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915, 2003.

[39] Yue Tan, Guodong Long, Jing Jiang, and Chengqi Zhang. Influence-oriented personalized federated learning. In *IEEE International Conference on Data Mining*, 2026.

[40] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.

[41] Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.

[42] Xiaolei Wang, Yuexin Wang, Tianhong Dai, Huihui Bai, Yao Zhao, and Jimin Xiao. Hunting normality from query sample via residual learning for generalist anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 43093–43102, 2026.

[43] Yili Wang, Yixin Liu, Xu Shen, Chenyu Li, Kaize Ding, Rui Miao, Ying Wang, Shirui Pan, and Xin Wang. Unifying unsupervised graph-level anomaly detection and out-of-distribution detection: A benchmark. In *International Conference on Learning Representations*, 2025.

[44] Yuexin Wang, Xiaolei Wang, Yizheng Gong, and Jimin Xiao. Normal-abnormal guided generalist anomaly detection. *Advances in Neural Information Processing Systems*, 38:62560–62585, 2026.

[45] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[46] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[47] Dong Xiangyu, Zhang Xingyi, and Wang Sibo. Rayleigh quotient graph neural networks for graph-level anomaly detection. In *ICLR*, 2024.

[48] Chunjing Xiao, Shikang Pang, Wenxin Tai, Yanlong Huang, Goce Trajcevski, and Fan Zhou. Motif-consistent counterfactuals with adversarial refinement for graph-level anomaly detection. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3518–3526, 2024.

[49] Pinar Yanardag and Svn VN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015.

[50] Zhenyu Yang, Ge Zhang, Jia Wu, Jian Yang, Shan Xue, Amin Beheshti, Hao Peng, and Quan Z Sheng. Global interpretable graph-level anomaly detection via prototype. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3586–3597, 2025.

[51] Xincheng Yao, Zixin Chen, Chao Gao, Guangtao Zhai, and Chongyang Zhang. Resad: A simple framework for class generalizable anomaly detection. *Advances in Neural Information Processing Systems*, 37:125287–125311, 2024.

[52] Ge Zhang, Zhenyu Yang, Jia Wu, Jian Yang, Shan Xue, Hao Peng, Jianlin Su, Chuan Zhou, Quan Z Sheng, Leman Akoglu, et al. Dual-discriminative graph neural network for imbalanced graph-level anomaly detection. *Advances in Neural Information Processing Systems*, 35:24144–24157, 2022.

[53] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

[54] Xiong Zhang, Hong Peng, Changlong Fu, Xin Jin, Yun Yang, and Cheng Xie. Ta-ggad: Testing-time adaptive graph model for generalist graph anomaly detection. *arXiv preprint arXiv:2603.09349*, 2026.

[55] Yunfeng Zhao, Yixin Liu, Qingfeng Chen, Shiyuan Li, Yue Tan, and Shirui Pan. Fedcigar: A personalized reconstruction approach for federated graph-level anomaly detection. In *International Joint Conference on Artificial Intelligence*, 2026.

[56] Xin Zheng, Yixin Liu, Zhifeng Bao, Meng Fang, Xia Hu, Alan Wee-Chung Liew, and Shirui Pan. Towards data-centric graph machine learning: Review and outlook. *arXiv preprint arXiv:2309.10979*, 2023.

[57] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. In *The Twelfth International Conference on Learning Representations*, 2024.

[58] Jiawen Zhu and Guansong Pang. 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*, pages 17826–17836, 2024.

A Detailed Related Work

A.1 Graph-Level Anomaly Detection

Graph-level anomaly detection (GLAD) focuses on identifying abnormal graphs that significantly deviate from the dominant graph patterns [52, 27, 38]. In many real-world applications, anomaly annotations are scarce or unavailable, making one-class learning the mainstream setting for GLAD. Under this setting, models are trained solely on normal graphs, while anomalous graphs are identified during inference based on their deviations from learned normal patterns (also referred to as the unsupervised setting in prior studies).

Recently, GNN-based approaches have become the prevailing solutions for GLAD and achieved promising performance. These methods can be broadly categorized into two groups, namely, reconstruction-based methods and contrastive learning-based methods. The reconstruction-based methods aim to reconstruct input graphs and detect anomalies based on reconstruction errors, assuming that anomalous graphs are harder to reconstruct [22, 5, 14]. For example, GLADC [22] takes autoencoder-based reconstruction as an important supervision signal for GLAD. MUSE [14] introduces a two-stage strategy that combines graph reconstruction with reconstruction-error modeling for anomaly identification. FedCIGAR [55] employs a personalized reconstruction approach for Federated learning [39].

The contrastive learning-based methods learn graph representations by contrasting multiple views of the data, using consistency or agreement between views to distinguish normal patterns from anomalies [15, 18, 50, 11]. For instance, SIGNET [18] measures graph abnormality by maximizing the mutual information between the original graph view and its corresponding dual hypergraph representation. CVTGAD [15] further introduces a transformer-based cross-view attention mechanism to directly capture the inter-relationship between augmented graph views for anomaly detection.

There are also several studies that leverage advanced techniques, such as evolutionary mapping [25], mixture-of-experts [4], and knowledge distillation [23], to address the GLAD problem. Specifically, GmapAD [25] adopts an evolutionary search strategy to identify representative nodes and detects anomalies according to the similarity between graphs and the selected representatives. OCGTL [35] improves graph-level anomaly detection by integrating self-supervised transformation learning to alleviate issues such as hypersphere collapse and performance flip in deep one-class learning.

Existing GLAD methods usually follow a one-dataset-one-model paradigm, where models are trained using either purely normal data (one-class unsupervised setting [43]) or labeled normal and anomalous data (supervised setting [52, 25, 47]), and evaluated on the same dataset. This paradigm increases the requirement for large amounts of labeled or curated data and limits its broad applicability.

A.2 Generalist Anomaly Detection

Recently, anomaly detection research has increasingly shifted toward generalist paradigms, where a unified pre-trained model is expected to generalize across different datasets and application scenarios. Instead of retraining separate models for each domain, these methods usually rely on pretrained foundation architectures together with lightweight test-time adaptation strategies such as prompts, contextual examples, or conditional modules.

In computer vision, many generalist approaches are built upon vision-language models (VLMs). For example, WinCLIP reformulates industrial anomaly detection by matching image features with compositional textual prompts at multiple granularities in CLIP, supporting zero-shot and few-shot anomaly recognition without additional task-specific optimization [7, 57]. Besides VLM-based approaches, generative foundation models have also been explored. Diffusion-based methods reuse denoising processes as generic perceptual priors and identify anomalies according to reconstruction or score discrepancies [1]. To further reduce dependence on manually designed prompts, InCTRL introduces an in-context residual learning framework, where query samples are compared with a few normal examples during inference for adaptive anomaly estimation [58]. Recent works further enhance generalization through residual feature modeling [51], normal-abnormal guidance [44], and residual-guided normality learning [42].

For time-series anomaly detection, recent studies also investigate reusable pretrained detectors across domains. FAE pretrains a variational autoencoder-based generative model on large-scale time-series

corpora and combines it with dilated convolutions to model complex temporal dependencies, aiming to support zero-shot anomaly detection on unseen datasets [12]. DADA further improves cross-domain transferability by introducing adaptive bottlenecks and dual adversarial decoders to handle domain divergence and diverse anomaly patterns [37].

Generalist anomaly detection has also been explored in graph learning, but they are all focus on node-level anomaly detection. ARC adopts an in-context adaptation strategy that extracts dataset-specific regularities from a few normal nodes at inference time, combining feature alignment, residual encoding, and cross-attention scoring for dynamic adaptation [19]. Similarly, AnomalyGFM develops a graph foundation model by aligning learnable normal and abnormal prototypes with residual neighborhood representations, enabling both zero-shot inference and few-shot adaptation on unseen graph datasets [34]. Recently, REFI-GAD introduces a fingerprint-based approach to boost the performance [20]. However, due to the fundamental differences between node-level and graph-level anomaly detection, these node-level approaches cannot be directly applied to GLAD.

Overall, existing generalist anomaly detection studies suggest that strong transferability mainly relies on robust pretrained representations together with effective test-time adaptation mechanisms, rather than dataset-specific retraining. Following this idea, our method targets a new research direction, i.e., generalist model for GLAD.

B Pseudo-anomalies Construction for Pre-training

Training generalist models typically depends on large volumes of labeled data [21]. However, annotating graph-structured data demands substantial domain expertise, making it infeasible in practice [18]. In this section, we provide a detailed version of how we construct $\mathcal{T}^{\text{train}}$ via **cross-domain motif injection**. Specifically, given a batch of graphs $\mathcal{B} = \{\mathcal{G}_1^{\mathcal{B}}, \dots, \mathcal{G}_B^{\mathcal{B}}\}$ sampled from a dataset $\mathcal{D}_{\text{curr}}^{\text{train}} \in \mathcal{T}^{\text{train}}$, we treat these graphs as inliers for the corresponding graph application. Pseudo-anomalies are then constructed by incorporating cross-dataset information, where each normal graph is augmented with motifs sampled from graphs drawn from other datasets to introduce anomalous information. Formally, for each graph $\mathcal{G}_i^{\mathcal{B}} \in \mathcal{B}$, we build a corresponding outlier graph $\bar{\mathcal{G}}_i^{\mathcal{B}}$ as a pseudo-anomaly. Let $\{\mathcal{M}_1, \dots, \mathcal{M}_K\}$ denote the set of motifs sampled via random walks from graphs drawn from $\mathcal{T}^{\text{train}} \setminus \{\mathcal{D}_{\text{curr}}^{\text{train}}\}$, where each motif is defined as $\mathcal{M}_k = (\mathcal{V}_k^{\text{motif}}, \mathcal{E}_k^{\text{motif}}, \mathbf{X}_k^{\text{motif}})$. We then introduce these motifs to $\mathcal{G}_i^{\mathcal{B}}$ by connecting a random node with the root node of the motif, resulting in the pseudo anomaly graph:

$$\bar{\mathcal{G}}_i^{\mathcal{B}} = (\mathcal{V}_i^{\mathcal{B}} \cup \mathcal{V}_1^{\text{motif}} \cup \dots, \mathcal{E}_i^{\mathcal{B}} \cup \mathcal{E}_1^{\text{motif}} \cup \dots \cup \mathcal{E}^{\text{join}}, \mathbf{X}_i^{\mathcal{B}} \oplus \mathbf{X}_1^{\text{motif}} \oplus \dots), \tag{7}$$

where $\mathcal{E}^{\text{join}}$ contains the edges that join the root node of motifs and the graph. Moreover, we denote the mask of injected nodes (i.e., nodes in out-domain motifs) as $\tilde{\mathbf{m}}_i \in \{0,1\}^{|\tilde{\mathcal{V}}_i^{\mathcal{B}}|}$, where

$$(\tilde{\mathbf{m}}_i)_j = \begin{cases} 1, & \text{if } v_j \in \bigcup \mathcal{V}_k^{\text{motif}}, \\ 0, & \text{otherwise.} \end{cases}$$

The cross-domain motif injection provides a simple yet effective way to construct negative samples. In many graph applications, the distinction between normal and anomalous instances is often determined by discriminative subgraphs that encapsulate the graph's key information [18]. (e.g., functional groups in molecular graphs that determine chemical properties.) Injecting out-of-domain motifs perturbs the semantic consistency of inlier graphs, thereby yielding realistic pseudo-anomalies. Moreover, it also offers a fine-grained supervisory signal with injected out-of-domain motifs to guide the training of the generalist GLAD model.

C Discrepancy Design

We construct a set of complementary discrepancy signals for each incoming graph, providing a holistic characterization of the incoming graph.

Prior Discrepancy. We first utilize the pre-trained model to obtain the subgraph G' on the test graph. Hence, the prior discrepancy is defined as:

$$s^{\text{prior}} = -\cos(\mathbf{z}_{\mathcal{G}'}, \mathbf{t}_{\mathbf{c}}), \tag{8}$$

Algorithm 1 GenGLAD: Cross-domain Motif Injection for Constructing Pseudo-anomaly

Input: Training task collection $\mathcal{T}$, batch of normal graph $\mathcal{B} \subset \mathcal{D}_{\text{curr}}^{\text{train}} \in \mathcal{T}$, number of injected motifs K, random walk length l.
Output: A batch of pseudo-anomalies $\bar{\mathcal{B}}$ with injected node masks.

$\bar{\mathcal{B}} \leftarrow \emptyset$
for graph $\mathcal{G}_i^{\mathcal{B}} = (\mathcal{V}_i^{\mathcal{B}}, \mathcal{E}_i^{\mathcal{B}}, \mathbf{X}_i^{\mathcal{B}}) \in \mathcal{B}_i$ **do**
  Sample motifs $\{\mathcal{M}_1, ..., \mathcal{M}_K\}$ from $\mathcal{T}^{\text{train}} \setminus \{\mathcal{D}_{\text{curr}}^{\text{train}}\}$ with random walk of length l.
  $\bar{\mathcal{G}}_i^{\mathcal{B}} \leftarrow \mathcal{G}_i^{\mathcal{B}}$.
  for $k = 1, \ldots, K$ **do**
    Construct the joining edge e_k^{join} by connecting the root node of $\mathcal{M}_k$ with a randomly selected nodes in $\mathcal{V}_i^{\mathcal{B}}$.
    Attach $\mathcal{M}_k$ to current graph: $\bar{\mathcal{G}}_i^{\mathcal{B}} \leftarrow (\bar{\mathcal{V}}_i^{\mathcal{B}} \cup \mathcal{V}_k^{\text{motif}}, \bar{\mathcal{E}}_i^{\mathcal{B}} \cup \mathcal{E}_k^{\text{motif}} \cup \{e_k^{\text{join}}\}, \bar{\mathbf{X}}_i^{\mathcal{B}} \oplus \mathbf{X}_k^{\text{motif}})$.
  end for
  Obtain the injected node mask $\tilde{\mathbf{m}}_i \in \{0, 1\}$.
  $\bar{\mathcal{B}} \leftarrow \bar{\mathcal{B}} \cup \{(\bar{\mathcal{G}}_i^{\mathcal{B}}, \tilde{\mathbf{m}}_i)\}$
end for
return $\bar{\mathcal{B}}$.

which estimates the difficulty of estimating the discriminative subgraph with the knowledge learned during pre-training.

Domain Discrepancy. We quantify how much the graph deviates from the domain-level context. Let $\mathbf{x}_{\mathcal{G}'} = \text{Readout}_{\mathcal{G}'}(\mathbf{X})$ denote the readout attribute on the discriminative subgraph. To capture complementary geometry, we consider both cosine and Euclidean distances to get discrepancy in both angular and scale differences:

$$s_{\text{cos}}^{\text{domain}} = -\cos(\mathbf{x}_{\mathcal{G}'}, \mathbf{c}), \quad s_{\text{euc}}^{\text{domain}} = \|\mathbf{x}_{\mathcal{G}'} - \mathbf{c}\|_2. \tag{9}$$

Class-level Discrepancy. We further measure deviation from the class prototype $\mathbf{q}$ estimated from few-shot normal graphs:

$$\mathbf{q} = \frac{1}{|\mathcal{T}^{\text{test}}|} \sum_{\mathcal{G} \in \mathcal{T}^{\text{test}}} \text{Readout}_{\mathcal{G}'}(\mathbf{H}), \tag{10}$$

$$s_{\text{cos}}^{\text{class}} = -\cos(\mathbf{z}_{\mathcal{G}'}, \mathbf{q}), \quad s_{\text{euc}}^{\text{class}} = \|\mathbf{z}_{\mathcal{G}'} - \mathbf{q}\|_2. \tag{11}$$

Instance-level Discrepancy. Beyond prototype-level statistics, we capture fine-grained relational structure via similarity distributions. Let $\mathcal{D}_{\mathcal{S}}$ denote the pairwise similarity distribution among support graphs, and $\mathcal{D}_G$ denote the similarity distribution between the query graph and each support instance:

$$\mathcal{D}_G = \{\cos(\mathbf{z}_{\mathcal{G}'}, \mathbf{z}_{\mathcal{G}_s})\}_{s=1}^K. \tag{12}$$

We then define the instance-level discrepancy as the 1-D Wasserstein distance:

$$s^{\text{ins}} = W_1(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_G), \tag{13}$$

where the Wasserstein distance captures distributional discrepancies beyond point-wise similarity.

D Pseudo-anomalies Construction for Adaptation

In this section, we provide a detailed version of how we construct pseudo-anomalies from few-shot normal data supervision. Since the few-shot supervision only contains normal instances, directly learning the $\mathbf{w}$ for the scoring function is infeasible. However, as all normal instances come from the same domain, it is unsuitable apply cross-domain motif injection. While reusing the motif sampling during training can be a potential bandit, this may raise potential data leakage and privacy concerns.

To address the aforementioned concerns, we introduce an alternative pseudo-anomaly construction way that effectively utilizes the few-shot normal graphs to generate semantically corrupted graphs and graphs with over-strong patterns, thereby supervising the contrastive learning objective with target domain-specific insights in a data-efficient manner. Specifically, to simulate anomalous

graphs with out-of-domain characteristics, we build pseudo-anomaly graphs (negative outliners) by **structure perturbation**, i.e., applying structural corruption to normal graphs by shuffling edges, thereby destroying meaningful relational structures while introducing out-of-domain structure patterns. Formally, given a normal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, we generate a corrupted variant by replacing its edge set with randomly sampled node pairs. Specifically, we construct a new edge set $\tilde{\mathcal{E}}$ by sampling without replacement from the set of all possible unordered node pairs:

$$\tilde{\mathcal{E}} \sim \text{Uniform}\Big(\{(u, v) \mid u, v \in \mathcal{V},\, u \neq v\}\Big), \quad |\tilde{\mathcal{E}}| = |\mathcal{E}|. \tag{14}$$

The resulting pseudo-anomalous graph is defined as:

$$\tilde{\mathcal{G}}_{\text{shuf}}^{\text{anomaly}} = (\mathcal{V}, \tilde{\mathcal{E}}, \mathbf{X}). \tag{15}$$

In addition to structural corruption, real-world anomalies may also arise from over-expressed or exaggerated in-domain patterns [14], where certain structures appear with abnormally high intensity or frequency. For example, in molecular graphs, excessive presence of certain functional groups may lead to toxicity; in social networks, while active users are typically normal, abnormally high interaction frequency can indicate spammers. Motivated by this, we conduct **sample bridging** to construct positive outliers by amplifying patterns observed in few-shot normal samples.

Specifically, given two normal graphs $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathbf{X}_i)$ and $\mathcal{G}_j = (\mathcal{V}_j, \mathcal{E}_j, \mathbf{X}_j)$, we generate a bridged graph by merging them and introducing additional cross-graph connections. The resulting graph is defined as:

$$\tilde{\mathcal{G}}_{\text{bridge}}^{\text{anomaly}} = (\mathcal{V}_i \cup \mathcal{V}_j, \mathcal{E}_i \cup \mathcal{E}_j \cup \mathcal{E}_{\text{bridge}}, \mathbf{X}_i \oplus \mathbf{X}_j), \tag{16}$$

where $\mathcal{E}_{\text{bridge}}$ is the randomly sampled edge set to connect two graphs together. By bridging graphs together, this construction amplifies structural patterns beyond their normal range, thereby guilding GenGLAD to spot any over-strong in-domain patterns.

We then utilize the constructed pseudo-anomalies with the few-shot supervisions to optimize the loss function:

$$\mathcal{L}_{\text{adapt}} = \text{BCE}\big(\sigma(s), y_{\text{pseudo}}^{\text{adapt}}\big). \tag{17}$$

while keeping the encoder and memory bank frozen. This preserves the transferable knowledge learned during pre-training and avoids overfitting under few-shot supervision. Overall, structure perturbation and sampling bridging enable GenGLAD to learn target-application specific knowledge for GLAD, thereby complementing the transferable knowledge learned during the pre-training stage with cross-domain motif injection.

E Algorithm

The procedure of pre-training and few-shots adaptation GenGLAD is summarized in Algorithm 2 and Algorithm 3, respectively.

F Complexity Analysis

We analyze the computational complexity of each stage in GenGLAD. Let N denote the total number of graphs in $\mathcal{T}^{\text{train}}$, V and E the average numbers of nodes and edges in a graph, respectively, M the number of memory tokens, and K the number of retrieved memory. We also denote the number of few-shot normal graphs as N^{adapt}.

During pre-training, the time complexity of training set construction is $\mathcal{O}(N(V+E))$, where motif caching is employed to reduce computational overhead. Since the random-walk-sampled motifs are relatively small, the overall complexity is dominated by the graph merging operation, which primarily incurs copying overhead. When it comes to pre-training, for each graph, the GNN encoder requires $\mathcal{O}(E)$ for message passing. The subsequent subgraph sampling costs $\mathcal{O}(KV + E)$. In addition, it also takes $\mathcal{O}(M)$ to obtain the domain-related memory tokens for each batch. Overall, during pre-training, GenGLAD requires $\mathcal{O}(N(KV + E))$ time per training epoch, in addition to a one-time training set construction cost of $\mathcal{O}(N(V + E))$ and a per-batch overhead of $\mathcal{O}(M)$. As K and M are small constants, our method maintains good training efficiency and **scales comparably to training a vanilla GCN** [53].

Algorithm 2 GenGLAD: Generalist Pre-training

Input: Training task collection $\mathcal{T}^{\text{train}} = \{\mathcal{D}_1^{\text{train}}, \ldots, \mathcal{D}_N^{\text{train}}\}$, training epochs $E^{\text{pre-train}}$, batch size B, top-K retrieval number, Gumbel temperature τ, learning rate lr.
Output: Pre-trained GenGLAD.

Randomly initialize the parameters of GenGLAD.
$\mathcal{P} \leftarrow \{(\mathcal{D}_i^{\text{train}} : \mathcal{M}^{\text{pool}})\}_{i=1}^{N}$ (Pre-sampling motifs for each dataset to avoid computation overhead during cross-domain motif injection)
for $epoch = 1, \ldots, E^{\text{pre-train}}$ **do**
  $\mathcal{B}^{\text{all}} \leftarrow$(randomly split input datasets into batches of size B.)
  for batch $\mathcal{B} \in \mathcal{B}^{\text{all}}$ **do**
    $\mathcal{L}_{\text{graph}} \leftarrow 0$
    $\mathcal{L}_{\text{node}} \leftarrow 0$
    $\mathcal{L}_{\text{reg}} \leftarrow 0$
    Retrieve top-K relevant memory tokens $\tilde{\mathbf{T}}$ from $\mathbf{T}$ according to cosine similarity with $\mathbf{c}$.
    Obtain pseudo-anomalies $\bar{\mathcal{B}}$ following Algorithm 1 with the cached motifs $\mathcal{P}$
    $\tilde{\mathcal{B}}_i \leftarrow \mathcal{B}_i \cup \bar{\mathcal{B}}_i$.
    $\mathbf{c} \leftarrow \frac{1}{|\mathcal{B}_i|} \sum_{\mathcal{G} \in \mathcal{B}_i} \text{Readout}_{\mathcal{G}}(\mathbf{X})$.
    $\mathbf{t}_{\mathbf{c}} \leftarrow \text{Mean}(\tilde{\mathbf{T}})$.
    for graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{\mathbf{X}}) \in \tilde{\mathcal{B}}_i$ **do**
      $\mathbf{H} \leftarrow \text{GNN}(\mathbf{X}, \mathbf{A})$.
      Obtain $\mathcal{G}'$, $\mathbf{m}$ following Section 4.1.1.
      $\mathbf{z}_{\mathcal{G}'} \leftarrow \text{Readout}_{\mathcal{G}'}(\mathbf{H})$.
      $\mathcal{L}_{\text{graph}} \leftarrow \mathcal{L}_{\text{graph}} + \text{BCE}\Big(\sigma(1 - \cos(\mathbf{z}_{\mathcal{G}'}, \mathbf{t}_{\mathbf{c}})), \tilde{y}\Big)$.
      $\mathcal{L}_{\text{node}} \leftarrow \mathcal{L}_{\text{node}} - \mathbb{E}_{u \sim \tilde{\mathbf{y}}_u^{(n)}=1}[\log(1 - m_u)]$.
      $\mathcal{L}_{\text{reg}} \leftarrow \mathcal{L}_{\text{reg}} + \|\mathbf{m}\|_1$.
    end for
    $\mathcal{L}_{\text{pre}} \leftarrow \mathcal{L}_{\text{graph}} + \lambda_n \mathcal{L}_{\text{node}} + \lambda_s \mathcal{L}_{\text{reg}}$.
    Back-propagate $\mathcal{L}_{\text{pre}}$ to update the parameters of GenGLAD with learning rate lr.
  end for
end for
return pre-trained GenGLAD model.

During adaptation, pseudo-anomalies are constructed from N^{adapt} few-shot normal graphs through structure perturbation and sample bridging. Structure perturbation requires $\mathcal{O}(E)$, while graph bridging costs $\mathcal{O}(V+E)$. Overall, each round of pseudo-anomaly construction requires $\mathcal{O}(N^{\text{adapt}}(V+E))$ as we obtain a balanced number of normal and pseudo-anomaly during training. As inference only involves memory-guided subgraph extraction and discrepancy scoring, resulting in a complexity of $\mathcal{O}(KV+E)$. To summarize, each adaptation round incurs a time complexity of $\mathcal{O}(N^{\text{adapt}}(KV+E))$, in addition to an $\mathcal{O}(M)$ overhead for obtaining the domain-related memory tokens at the beginning of training. With the help of light-weight module design, the adaptation of GenGLAD is extremely efficient as demonstrated in Figure 4b.

G Detailed Experimental Setup

G.1 Description of Datasets

In this section, we present additional details of all graph datasets used in our experiments. Table 5 reports key statistics, including the number of graphs as well as the average number of nodes and edges. All datasets are sourced from TUDataset [26], a publicly available graph benchmark repository. We follow UB-GOLD [43] to configure the GLAD experimental settings. Specifically, the detailed descriptions for the datasets are given as follows:

NCI1 [41]: A dataset of chemical compounds screened for anti-cancer activity, where each graph corresponds to a molecular structure. It is widely used for evaluating graph learning methods in anti-cancer drug discovery tasks.

Algorithm 3 GenGLAD: Few-shot Adaptation

Input: Pre-trained GenGLAD, few-shot normal set $\mathcal{S} \subset \mathcal{D}_i^{\text{test}}$, adaptation epochs E^{adapt}, learning rate lr.
Output: Anomaly scores for graphs in $\mathcal{D}_i^{\text{test}}$.

Randomly initialize scoring parameter $\mathbf{w}$.
Retrieve top-K relevant memory tokens $\tilde{\mathbf{T}}$ from $\mathbf{T}$ according to cosine similarity with $\mathbf{c}$.
Construct the pseudo-anomaly set $\bar{\mathcal{S}}$ $\mathcal{B}^{\text{adapt}}$ using following Appendix D.
$\mathbf{c} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{\mathcal{G} \in \mathcal{S}} \text{Readout}_{\mathcal{G}}(\mathbf{X})$.
$\mathbf{q} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{\mathcal{G} \in \mathcal{S}} \mathbf{z}_{\mathcal{G}'}$.
for $epoch = 1, \ldots, E^{\text{adapt}}$ **do**
  $\mathcal{L}_{\text{adapt}} \leftarrow 0$
  Obtain $\bar{\mathcal{S}}$ following Algorithm 1 and form fine-tuning batch $\tilde{\mathcal{S}}_i \leftarrow \mathcal{S}_i \cup \bar{\mathcal{S}}_i$.
  for graph $\tilde{\mathcal{G}}_i \in \tilde{\mathcal{S}}$ **do**
    Compute the four discrepancies following Section 4.2.
    $\mathcal{L}_{\text{adapt}} \leftarrow \mathcal{L}_{\text{adapt}} + \text{BCE}(\sigma(\mathbf{w}^\top \mathbf{s}), y_i^{\text{adapt}})$.
  end for
  Update scoring parameter $\mathbf{w}$ by back-propagating $\mathcal{L}_{\text{adapt}}$ with learning rate lr.
end for
for query graph $\mathcal{G} \in \mathcal{D}_i^{\text{test}}$ **do**
  Compute the four discrepancies following Section 4.2.
  $s_i \leftarrow \sigma(\mathbf{w}^\top \mathbf{s})$.
end for
return anomaly scores $\mathbf{s}$.

Table 4: Statistics of the datasets.

Datasets	# Graphs	Avg. Nodes	Avg. Edges
Molecular Graphs			
NCI1	4110	29.87	64.60
AIDS	2000	15.59	47.98
BZR	405	35.75	76.72
COX2	467	41.22	86.89
DHFR	756	42.43	89.09
MUTAG	188	17.93	57.52
PTC_MR	344	14.29	29.38
Biochemistry Graphs			
DD	1178	284.32	1715.63
ENZYMES	600	32.63	124.27
PROTEINS	1113	39.06	145.63
Social Networks			
COLLAB	5000	74.49	4914.43
IMDB-BINARY	1000	19.77	212.84
REDDIT-BINARY	2000	429.63	1425.13

AIDS [26]: A collection of molecular graphs representing chemical compounds tested for antiviral activity against HIV, commonly used in antiviral drug discovery research.

BZR [45]: A molecular dataset of benzodiazepine receptor ligands, where nodes denote atoms and edges denote chemical bonds, designed for studying molecular interactions with benzodiazepine receptors.

COX2 [26]: A dataset consisting of cyclooxygenase-2 inhibitor molecules represented as graphs, used for analyzing the structural properties of anti-inflammatory compounds.

DHFR [26]: A collection of molecular graphs for compounds that inhibit the dihydrofolate reductase (DHFR) enzyme, important for research in cancer therapy and antibacterial drug development.

MUTAG [8]: A benchmark dataset containing 188 mutagenic aromatic and heteroaromatic nitro compounds. Each graph represents a molecule, and the classification task determines whether the compound exhibits mutagenic effects on the bacterium *Salmonella typhimurium*.

PTC-MR [13]: A subset of the Predictive Toxicology Challenge (PTC) dataset, where molecular compounds are labeled according to their carcinogenicity. The suffix "MR" indicates experiments conducted on male rats.

DD [40]: A protein-protein interaction dataset in which graphs capture the complex relationships among proteins, supporting the study of biological processes and cellular functions.

ENZYMES [36]: A dataset of protein tertiary structures, where each graph represents an enzyme with nodes corresponding to secondary structure elements and edges indicating spatial adjacency relationships.

PROTEINS [9]: A protein graph dataset containing structural and interaction information, where nodes represent amino acids and edges capture interactions between them, facilitating the analysis of protein functions and biological interactions.

COLLAB [49]: A scientific collaboration network dataset in which nodes correspond to researchers and edges indicate co-authorship relationships across multiple research domains.

IMDB-BINARY [49]: A movie collaboration network dataset where nodes represent actors and edges denote co-appearances in films, enabling the analysis of collaboration patterns in the film industry.

REDDIT-BINARY [49]: A dataset of Reddit discussion threads represented as graphs, where nodes correspond to users and edges capture interactions between them, reflecting the structural dynamics of online discussions.

As the native node and edge attributes may be unavailable or semantically incompatible across domains, we construct structure attributes, which not only addresses that issue but also unify the dimension. Such formulation can provides strong discriminative information for graph-level tasks from graph topology. Specifically, we follow the commonly used structure-based attribute construction, which consists of two components: the degree encoding $\mathbf{X}^{\mathrm{deg}}$ and the random-walk positional encoding $\mathbf{X}^{\mathrm{RW}}$. We concatenate them to obtain the structural embedding. The degree encoding is a one-hot representation of each node's degree, designed to characterize its local connectivity. We use a fixed dimensionality of 32, with degrees exceeding the supported range clipped to the final bin. To capture higher-order structural information, we additionally employ random-walk positional encoding that denotes as the probability that a k-step random walk returns to the starting point. By concatenating them together, we obtain a 64-dimensional structural embedding.

G.2 Details of Implementation

Hyper-parameters. We tune the key hyper-parameters of GenGLAD via random search over predefined grids. Specifically, the search space includes $\lambda_n \in \{0.0, 0.1, 0.5, 1.0, 1.5, 2.0\}$, $\lambda_s \in \{0.0, 0.01, 0.05, 0.1, 0.15, 0.2\}$, and the pre-training learning rate $\mathrm{lr}^{\text{pre-train}} \in \{1 \times 10^{-4}, 1 \times 10^{-3}\}$. For the remaining hyper-parameters, we set the number of memory tokens M to 128 and the number of retrieved in-domain memory tokens K to 12. For structural encoding, we set the dimensionality of both $\mathbf{X}^{\text{deg}}$ and $\mathbf{X}^{\text{RW}}$ to 32, yielding a 64-dimensional concatenated structure representation. The temperature parameter τ for Gumbel-Softmax and the hidden embedding dimension are set to 0.5 and 64, respectively. We pre-train the model for 30 epochs. For adaptation, we use a $\mathrm{lr}^{\text{adapt}} = 1 \times 10^{-4}$ and train for 10 epochs. We utilize the Adam optimizer for both pre-training and fine-tuning.

Implementation Pipeline. We employ a fixed set of hyperparameters to construct a generalist GLAD model across all datasets. First, all methods (including baselines and GenGLAD) are trained on the training set $\mathcal{T}^{\text{train}}$. Then, each method is adapted separately to the corresponding dataset using $\mathcal{T}^{\text{test}}$. For adaptation and few-shot fine-tuning, we ensure that the few-shot supervision is identical across all baselines and GenGLAD to guarantee a fair comparison. During pretraining, the memory tokens are randomly initialized and jointly optimized with the model through backpropagation, while in finetuning they are frozen to avoid overfitting.

Evaluation Metrics. Following standard practice in GAD and GLAD [24, 43], we evaluate performance using two widely adopted and complementary metrics: the area under the receiver

Table 5: Anomaly detection performance in terms of AUPRC (in percent, mean±std). Highlighted are the results ranked **first**, second, and third.

Method	AIDS	BZR	COX2	DHFR	MUTAG	PTC_MR	ENZYMES	PROTEINS	IMDB-B	REDDIT-B
Pre-training+Few-shot Fine-tuning										
GLocalKD	82.47±0.06	26.54±0.06	25.83±0.09	69.52±0.13	87.22±0.05	44.04±0.03	81.69±0.02	53.43±0.05	50.21±0.05	69.93±0.13
CVTGAD	76.10±0.00	21.15±0.03	23.30±0.01	67.41±0.00	94.71±0.00	48.36±0.02	82.58±0.00	30.82±0.00	42.19±0.49	OOM
SIGNET	95.64±0.00	13.55±0.02	25.13±0.02	67.25±0.01	82.78±0.03	51.26±0.19	81.72±0.00	40.01±0.00	52.33±0.01	52.30±0.00
MUSE	92.67±0.44	24.30±0.59	20.95±0.30	67.87±0.72	86.26±1.96	46.07±0.93	84.65±0.16	39.38±0.37	50.16±1.06	63.30±0.59
GLADPro	80.00±0.00	20.76±0.23	24.14±0.73	64.35±2.85	66.49±0.00	46.49±2.60	84.34±0.22	42.18±2.96	45.47±1.02	63.01±7.06
HC-GLAD	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
FANFOLD	91.63±0.00	21.09±0.01	16.70±0.01	69.28±0.00	94.52±0.01	43.96±0.01	82.90±0.00	41.71±0.01	56.12±0.01	34.01±0.00
Few-shot Fine-tuning Only										
GLocalKD	85.24±1.05	26.83±3.24	23.23±0.97	77.15±0.62	94.93±0.24	43.21±0.81	81.54±0.86	43.71±3.22	55.69±1.28	62.14±1.40
CVTGAD	89.90±3.71	21.82±6.16	20.77±3.09	58.70±3.62	83.76±8.41	41.03±2.03	86.22±1.74	32.98±1.35	53.45±7.46	OOM
SIGNET	86.33±2.17	18.94±3.97	18.45±1.54	52.52±2.12	74.93±11.66	45.08±0.62	79.56±0.61	48.12±3.36	51.78±3.49	48.22±2.48
MUSE	95.23±1.85	19.21±0.39	18.19±0.59	76.14±1.54	89.45±6.26	48.44±1.31	84.10±0.25	41.96±0.64	53.17±2.05	68.27±1.44
GLADPro	86.57±7.03	28.85±7.18	27.28±4.40	55.70±3.70	88.42±8.01	51.52±1.72	85.19±0.45	43.32±5.06	44.64±2.32	68.24±3.85
HC-GLAD	60.55±0.13	18.04±2.13	20.33±0.95	65.31±1.67	92.86±1.09	43.97±1.27	90.35±1.61	32.97±3.45	38.59±4.51	OOM
FANFOLD	85.76±7.21	18.58±0.76	17.90±1.26	71.24±3.06	85.06±13.52	46.54±1.80	84.58±1.49	51.46±4.15	52.80±2.68	35.58±1.00
Generalist										
GenGLAD	98.79±0.01	38.87±3.46	31.64±1.50	82.21±0.10	95.06±0.13	56.05±0.56	85.47±0.71	69.26±0.11	74.72±0.64	74.77±0.76

Table 6: AUROC of GenGLAD and its variants.

Variant	AIDS	BZR	COX2	DHFR	MUTAG	PTC_MR	ENZYMES	PROTEINS	IMDB-B	REDDIT-B
GenGLAD	95.84±0.07	63.89±1.18	62.92±0.30	74.96±0.42	90.75±0.17	60.58±0.47	54.37±1.74	73.63±0.16	74.11±0.25	73.63±0.20
Pre-training										
w/o $\mathcal{L}^{\text{node}}$	91.43±0.73	57.80±0.73	57.32±0.95	70.98±0.94	77.98±0.32	58.20±1.69	54.47±0.60	66.03±0.19	58.03±0.95	62.77±0.32
w/o $\|\mathbf{m}\|_1$	93.82±0.32	59.74±1.37	58.20±0.17	71.30±0.34	86.32±0.43	56.68±0.19	53.70±0.29	68.53±0.37	60.06±2.43	67.07±0.33
Discrepancy Measures										
w/o Prior Disc.	93.75±0.16	60.82±3.90	59.03±1.49	73.05±0.49	88.00±0.59	57.31±1.66	52.54±1.19	71.66±0.37	71.15±0.99	71.38±0.43
w/o Domain Disc.	93.31±3.41	57.82±4.22	46.87±6.66	73.77±0.52	88.58±0.36	46.65±0.67	47.58±2.00	63.76±1.57	51.65±2.44	65.93±1.78
w/o Cls. Disc.	92.60±3.58	52.41±3.05	59.28±0.96	45.63±1.53	87.07±4.98	55.56±0.55	53.58±0.64	60.20±1.48	63.72±2.81	61.92±1.45
w/o Inst. Disc.	95.77±0.13	56.62±2.39	59.03±1.80	74.25±0.51	89.38±2.21	58.84±0.74	53.75±0.80	73.75±0.14	72.96±0.99	72.16±1.69
Scoring Module										
MEAN	6.15±0.06	49.06±0.33	56.86±1.27	28.92±0.69	13.34±0.72	46.90±0.22	46.29±0.21	41.54±0.10	42.00±0.17	50.32±0.06
MAX	27.63±0.18	33.71±0.12	57.56±0.01	40.32±0.05	56.57±0.23	53.89±0.09	44.40±0.08	61.94±0.25	30.12±0.04	67.78±0.18
MIN	8.44±0.25	47.02±0.52	58.34±1.84	29.56±0.87	16.36±1.26	45.19±0.41	46.90±0.53	38.05±0.24	43.57±0.24	45.77±0.25

operating characteristic curve (AUROC) and the area under the precision–recall curve (AUPRC). Higher AUROC and AUPRC values indicate better detection performance. We report the mean AUROC and AUPRC along with standard deviations over five independent runs.

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

H Additional Experimental Results

H.1 Main Results in AUPRC

Table 5 reports the anomaly detection performance in terms of AUPRC. Overall, GenGLAD achieves consistently strong performance across diverse datasets and demonstrates clear advantages in the generalist GLAD setting. Specifically, GenGLAD obtains the best performance on 9 out of 10 datasets, while remaining highly competitive on the remaining datasets. Notably, substantial improvements can be observed on datasets such as DHFR, IMDB-B, and REDDIT-B, indicating the effectiveness of GenGLAD in capturing transferable anomaly-related patterns across domains. In contrast, conventional GLAD baselines generally exhibit limited gains from pre-training and often suffer noticeable performance degradation under the few-shot fine-tuning setting. This trend is particularly evident on datasets with limited training samples, where several baselines achieve relatively low AUPRC scores or even unstable performance. These observations further validate that existing GLAD approaches heavily rely on sufficient in-domain training data and struggle to generalize across domains. The strong AUPRC performance of GenGLAD demonstrates that the proposed discriminative subgraph extraction and discrepancy-based adaptation strategy can effectively identify anomalous graphs while maintaining robust cross-domain generalization under limited supervision.

H.2 More Results of Sensitivity to #shot

Table 7: AUPRC of GenGLAD and its variants.

Variant	AIDS	BZR	COX2	DHFR	MUTAG	PTC_MR	ENZYMES	PROTEINS	IMDB-B	REDDIT-B
GenGLAD	98.79±0.01	38.87±3.46	31.64±1.50	82.21±0.10	95.06±0.13	56.05±0.56	85.47±0.71	69.26±0.11	74.72±0.64	74.77±0.76
Pre-training										
w/o $\mathcal{L}^{\text{node}}$	97.76±0.12	30.72±0.14	25.73±0.33	80.16±1.33	87.61±1.58	55.29±1.44	86.69±1.58	63.80±0.48	59.63±0.59	67.24±1.74
w/o $\|\mathbf{m}\|_1$	98.35±0.14	37.22±7.62	25.87±0.34	80.18±0.15	92.94±0.13	54.66±0.11	83.76±0.75	65.35±0.97	55.08±1.31	69.35±1.74
Discrepancy Measures										
w/o Prior Disc.	97.79±0.05	31.92±5.12	27.32±1.23	80.18±0.42	93.77±0.35	54.67±1.42	82.44±0.60	68.04±0.25	71.85±1.11	72.28±0.92
w/o Domain Disc.	97.11±1.06	30.19±8.18	20.34±3.63	81.21±0.35	94.30±0.17	43.49±0.25	81.74±1.89	59.75±1.27	52.02±2.92	70.22±1.95
w/o Cls. Disc.	96.60±1.17	21.32±2.38	25.77±0.47	63.11±1.57	92.71±3.41	53.54±0.27	84.19±0.30	54.69±1.92	62.62±2.21	60.04±1.35
w/o Inst. Disc.	96.78±0.05	23.59±1.52	26.43±1.31	81.78±0.11	93.91±1.56	55.22±0.47	84.60±0.49	69.18±0.08	73.25±1.41	73.69±1.38
Scoring Module										
MEAN	60.97±0.02	19.30±0.18	24.81±0.94	48.40±0.47	48.75±0.28	43.97±0.65	80.82±0.19	35.05±0.06	43.70±0.09	51.79±0.11
MAX	74.09±0.06	15.92±0.02	24.38±0.01	59.26±0.11	81.51±0.16	49.46±0.08	79.82±0.06	49.20±0.55	37.89±0.02	60.52±0.11
MIN	61.64±0.08	18.63±0.09	26.41±2.57	49.04±0.75	49.45±0.55	42.87±0.78	81.24±0.22	33.96±0.35	44.73±0.12	49.42±0.24

Figure 6 presents the performance of GenGLAD with varying numbers of few-shot samples for adaptation on individual datasets, while Figure 5 reports the averaged results on science graph datasets. Overall, GenGLAD demonstrates stable adaptation behavior under different few-shot settings. In most cases, increasing the number of fine-tuning graphs leads to improved AUROC and AUPRC performance, indicating that the proposed adaptation framework can effectively exploit additional normal samples when available.

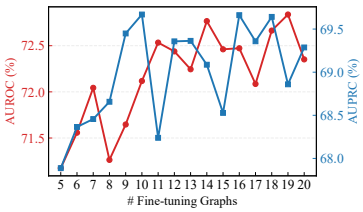


Figure 5: Sensitivity of #shot (Science groups).

Moreover, even under extremely limited supervision, GenGLAD still achieves competitive performance on most datasets, demonstrating the strong transferability of the pre-trained backbone model. We also observe that the sensitivity to the number of few-shot samples varies across datasets. For example, datasets such as IMDB-B and DHFR exhibit relatively consistent performance improvements as more adaptation samples are introduced, while datasets such as BZR and MUTAG show larger fluctuations due to their relatively small scales and higher structural diversity. Nevertheless, the overall trends suggest that GenGLAD maintains robust adaptation capability across diverse domains and does not heavily rely on large amounts of target-domain data.

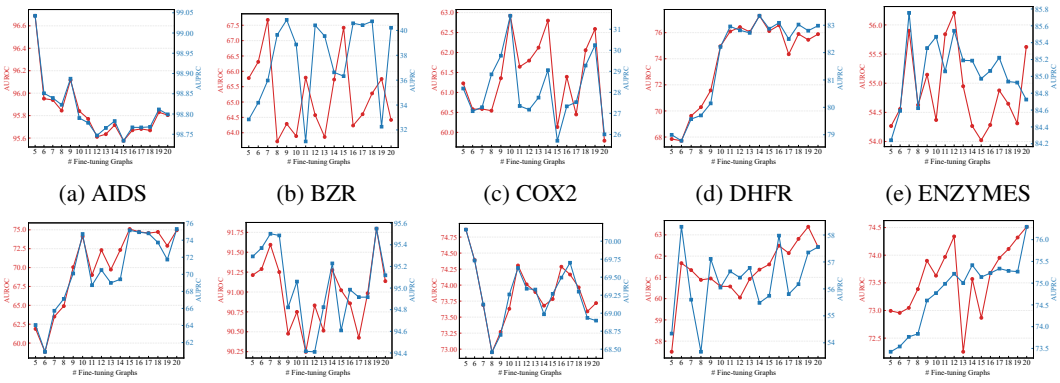
H.3 Full Results of Ablation Study

Full Ablation Results. Tables 6 and 7 present the complete ablation results of GenGLAD in terms of AUROC and AUPRC, respectively. Overall, the observations are consistent with those reported in the main paper, further validating the effectiveness of each component in GenGLAD.

For the pre-training stage, removing either the node-level pseudo anomaly supervision $\mathcal{L}_{\text{node}}$ or the sparsity regularization on $\mathbf{m}$ consistently degrades performance on most datasets. In particular, removing $\mathcal{L}_{\text{node}}$ leads to more noticeable performance drops, demonstrating the importance of fine-grained pseudo anomaly supervision for learning transferable discriminative substructures.

For the adaptation stage, all discrepancy measures contribute positively to the final performance. Among them, the domain discrepancy and class discrepancy generally have the largest impact. Their removal causes substantial degradation on several datasets and even results in near-random performance in certain cases, highlighting the importance of capturing both domain-aware and class-level abnormality evidence. In comparison, the prior discrepancy and instance discrepancy provide complementary gains by refining the anomaly characterization from different perspectives.

We further evaluate the effectiveness of the scoring module by replacing it with simple aggregation strategies, including mean, max, and min pooling. As shown in the tables, these heuristic strategies consistently underperform the learnable scoring module and often lead to unstable results across datasets. In some cases, they even produce performance close to or below random guessing. These observations indicate that different discrepancy signals contribute unequally across datasets, and adaptive weighting is crucial for reliable anomaly detection under few-shot adaptation settings.



(a) AIDS (b) BZR (c) COX2 (d) DHFR (e) ENZYMES

(f) IMDB-B (g) MUTAG (h) PROTEINS (i) PTC_MR (j) REDDIT-B

Figure 6: Performance with varying numbers of few-shot samples for adaptation.

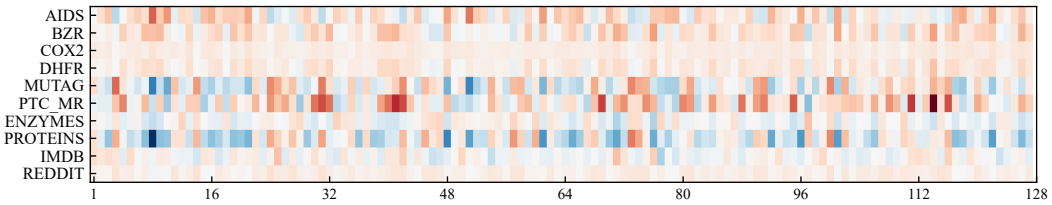


Figure 7: Visualization results of memory token importance.

H.4 Visualization of Memory Retrieval Weight

As illustrated in Figure 7, token activations vary across application domains. With the help of context-guided memory retrieval, GenGLAD can selectively leverage the most relevant pre-trained knowledge for GLAD, thereby improving generalization.

H.5 More Analysis on Efficiency

We further provide a detailed runtime breakdown in Table 8. As the results show, synthesizing the pseudo-graphs requires even less time than a single training epoch, demonstrating the negligible computational overhead of our graph synthesis procedure. Notably, the pretraining dataset, consisting of NCI1, DD, and COLLAB, contains 10,288 graphs, 830,144 nodes, and 26,858,668 edges in total. The reported runtime and memory consumption on this relatively large collection further demonstrate the practical efficiency and scalability of our method.

H.6 Quantitative Evaluation of Discriminative Subgraph

We further provide a quantitative evaluation for the extracted discriminative subgraph on BM-MN and BM-MT. We compare GenGLAD with SIGNET, which is trained under the conventional in-domain setting. As shown in Table 9, despite being pretrained for cross-dataset generalization rather than optimized separately on each target dataset (like SIGNET), GenGLAD achieves interpretation performance close to SIGNET, which is trained on 500 normal graphs on BM-MN and BM-MT. Overall, these results suggest that the pretrained discriminative-subgraph extractor can identify meaningful explanatory structures across diverse anomaly patterns.

Table 8: Runtime breakdown of GenGLAD in second. Pretraining is conducted on NCI1, DD, and COLLAB, while fine-tuning is evaluated on AIDS.

Process	Time (s)
Pretraining: pseudo-anomaly synthesis (motif pool $\mathcal{P}$)	0.85
Pretraining: per epoch	10.86
Pretraining: total	325.93
Fine-tuning: pseudo-anomaly synthesis (pool $\mathcal{S}$)	0.81
Fine-tuning: per epoch	0.04
Fine-tuning: total	0.42

Table 9: Quantitative Evaluation of Discriminative Subgraph. Results in AUROC (%).

Method	BM-MN	BM-MT
SIGNET	76.6	78.4
GenGLAD	72.1	76.7

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our paper aims to introduce and address generalist GLAD, which is consistent with what we state in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We mentioned limitations in the conclusion.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: Does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We report all the hyperparameters in the appendix. The detailed algorithm can also be find in the appendix.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in the supplemental material?

   Answer: [Yes]

   Justification: We use publicly available datasets, and the source code is provided.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all hyperparameters together with range , and details of pre-training and fine-tuning dataset construction in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard deviation error bar is provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include the hardware configuration in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm compliance with the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In the introduction, we highlight that generalist GLAD helps to safeguard diverse graph applications, including new domains with limited in-domain data.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We include citations for all used datasets and baselines we compared.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide documentation for the code in READEME.md.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM is only used for grammar and word choice.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.