Case-Based Interpretability in Graph-Level Anomaly Detection via Contrast with Normal Prototypes

Qiuran Zhao, Kai Ming Ting✉, and Xinpeng Li

State Key Laboratory for Novel Software Technology and
School of Artificial Intelligence, Nanjing University, Nanjing 210023, China
zhaoqr@lamda.nju.edu.cn, tingkm@nju.edu.cn, lixp@lamda.nju.edu.cn

Abstract. The task of graph-level anomaly detection (GLAD) aims to identify graphs that deviate significantly from the majority in a dataset. While existing deep learning methods achieve strong detection performance, they operate as black boxes that provide anomaly scores without human-interpretable explanations. Recent explainable GLAD methods either highlight suspicious substructures without referencing normal patterns, or rely on abstract latent vectors as prototypes that cannot be directly inspected by domain experts. We propose **Proto**type-based **G**raph-**L**evel **A**nomaly **D**etection (ProtoGLAD), an intrinsically interpretable framework that draws on core principles of case-based reasoning (CBR) to address these limitations. ProtoGLAD maintains a case base of normal prototype graphs discovered directly from the dataset using a density-adaptive point-set kernel. Given a graph, the framework retrieves its nearest normal prototype, which is a concrete, real graph instance, and determines anomalousness based on the similarity to its closest normal cluster. For each detected anomaly, ProtoGLAD generates a contrastive case-based explanation by explicitly comparing the anomalous graph with its nearest normal prototype at the node level, highlighting which substructures deviate from the normal reference and which align with it. This retrieve-and-contrast paradigm provides transparent, instance-grounded explanation that is lacking in existing GLAD methods. Extensive experiments demonstrate that ProtoGLAD achieves competitive detection performance while providing more human-interpretable case-based explanations compared to state-of-the-art methods.

Keywords: Graph-Level Anomaly Detection · Interpretability · Prototype-Based Explanation · Contrastive Explanation · Similarity-Based Retrieval

1 Introduction

Graphs are data structures widely used to represent complex relationships in various domains such as chemistry, traffic systems, and social networks. Recently, graph-level anomaly detection (GLAD) has emerged as a critical but challenging task in the graph data mining community [30,11,6,26,4,31,29]. The task of GLAD

aims to identify graphs that deviate significantly from the majority of graphs in a dataset of graphs. Reliable GLAD methods support applications such as toxic compound detection, malicious network pattern discovery, and abnormal brain connectivity identification.

Despite their promising detection performance, most advanced methods for GLAD operate as black-box models. They detect anomalies based on anomaly scores derived from, for example, the reconstruction error of a graph autoencoder [12,6], the learning discrepancy from a knowledge distillation framework [13], the consistency score from cross-view mutual information maximization [11], the distance to a learned hypersphere center in one-class classification [16,30], or the decision boundary learned with pseudo-anomalous graphs [4]. The anomaly scores offered by these methods are self-referential, which means they measure how well a graph conforms to a compressed version of itself or to an augmented view of itself, rather than comparing it to normality. Consequently, it is unclear how the detected anomalies can be understood when the detectors cannot provide human-interpretable explanations or verify whether their detected anomalies are indeed rare and different from the majority of the graphs.

Recent studies [11,26,29] seek to provide explanations for detected anomalies. However, they have notable limitations: they either lack a comparative normal reference to verify why a detected anomaly is deviant, or they rely on learnable latent vectors as prototypes. Such latent prototypes may diverge from actual normal graphs, yielding unreliable baselines for both detection and explanation.

We contend that the limitations above stem from a common root cause: existing methods lack the reasoning paradigm of **learning from concrete cases**. In case-based reasoning (CBR) [1], a new problem is solved by retrieving similar past cases from a case base, comparing the new instance with the, and reusing the associated solutions or explanations. This paradigm is naturally suited to the GLAD task: if a case base of representative normal graphs can be constructed from the dataset, then a new graph can be assessed by retrieving its most similar normal case. If the new graph is dissimilar to all normal cases, it is identified as anomalous, and the contrast between the new graph and its nearest normal case directly provides a human-interpretable explanation for the detection result.

Therefore, we propose **Proto**type-based **G**raph-**L**evel **A**nomaly **D**etection (ProtoGLAD), an intrinsically interpretable framework that leverages case-based reasoning for GLAD. ProtoGLAD constructs a case base of normal prototype graphs and their associated normal clusters. Each prototype is a real graph from the dataset, serving as a representative normal case. Given an anomalous graph, ProtoGLAD retrieves its nearest normal prototype and determines anomalousness based on the similarity to its corresponding closest normal cluster. This design naturally aligns with the CBR cycle of *retrieve*, *reuse*, and *retain*: each detected anomaly can be directly contrasted with its nearest typical normal case, providing an inherent basis for ante-hoc interpretability.

The main contributions of this work are:

– Proposing ProtoGLAD, an interpretable GLAD framework based on the principle of case-based reasoning. ProtoGLAD constructs a case base of nor-

mal prototype graphs directly from the dataset and detects anomalies based on their similarity to the nearest normal prototype and its associated cluster.
– Developing a node-level, contrastive explanation mechanism that identifies the specific substructures, both in the detected anomalous graph and in its nearest normal prototype, that contribute to its anomalousness, providing a fine-grained, contrastive explanation.
– Conducting extensive experiments on multiple real-world and synthetic datasets, demonstrating that ProtoGLAD achieves competitive detection performance compared to state-of-the-art GLAD methods, while providing more human-interpretable, case-based explanations.

2 Related Work

In this section, we review existing works from three perspectives: graph-level anomaly detection methods, explainable GLAD methods, and case-based reasoning with prototype-based explanations.

2.1 Graph-Level Anomaly Detection

GLAD methods are used to identify anomalous graphs that exhibit significant deviations from the majority of graphs in a dataset. Existing GLAD methods can be broadly categorized into two types: **two-step** methods and **end-to-end** methods. **Two-step methods** first generate graph-level embeddings using graph kernels [23] or self-supervised learning [27], then apply a detector such as LOF [3] or Isolation Forest [10]. MUSE [6] also follows this pipeline but uses a GNN-based autoencoder to extract reconstruction error representations for a one-class classifier. **End-to-end methods** leverage GNNs [7] to jointly learn graph representations and detect anomalies in a unified framework. Among them, some approaches presume that the anomaly labels of the graphs are accessible during model training, and thus frame GLAD as a supervised classification problem [28,29]. However, their strong dependence on labeled anomalies renders their application difficult in a real-world scenarios where such annotations are few or, more often than not, nonexistent. Therefore, most approaches focus on unsupervised learning, under the assumption that only normal samples are used during the training phase. For example, some of them adopt GNN-based graph reconstruction [12], where anomalies are identified as graphs that are poorly reconstructed by the model, while some others like GLocalKD [13] and KDGAE [31] use a knowledge distillation framework where anomalies are detected as those having large knowledge discrepancies. In addition, OCGTL [16] combines one-class classification with graph transformation learning to learn a compact normal boundary, while AGDiff [4] leverages latent diffusion models to generate pseudo-anomalous graphs for enhancing the decision boundary.

2.2 Explainable GLAD

Explainable graph-level anomaly detection has attracted increasing attention, and several recent methods have attempted to provide explanations for their

detected anomalies. SIGNET [11] identifies the specific substructures within a single graph that contribute to its anomaly score. It is done by extracting an important "bottleneck" subgraph using the information bottleneck (IB) principle. While it highlights the suspicious subgraph of the detected anomaly, it fails to provide a contrastive reference to a normal counterpart, making it difficult to verify why the highlighted subgraph is considered deviant.

GLADPro [26] aims to provide a general understanding of the model's behavior by learning global prototypes. It extends the IB principle to learn a set of prototypes that capture important subgraph patterns across the entire dataset. However, a key limitation is that these prototypes are modeled as learnable vectors in a latent space, rather than actual graphs directly discovered from the dataset, and high-dimensional latent vectors are difficult for humans to interpret.

More recently, X-GAD [29] aggregates common structural and attribute characteristics across anomalous graphs to construct anomaly prototypes. By contrasting graphs with these learned anomaly prototypes, X-GAD provides explanations that highlight recurring anomalous patterns shared among abnormal graphs. However, X-GAD relies on the availability of anomaly labels during training, which are often expensive or inaccessible. Furthermore, the presupposition of anomaly prototypes is a restrictive assumption, as anomalies do not necessarily exhibit consistent patterns.

2.3 Case-Based Reasoning and Prototype-Based Explanation

Case-based reasoning (CBR) [1] solves new problems by retrieving and adapting solutions from similar past cases stored in a case base. Prototype-based methods represent a specific CBR paradigm that constructs case bases from typical instances, for example, performing classification by comparing a query to learned class prototypes [18]. A core principle of CBR is that similar problems yield similar solutions, which has a natural connection to explainability: the system can present the retrieved cases as concrete evidence to justify its decisions [2]. Prototype-based explanations have been explored in various domains, especially in deep learning for image recognition [5,9], where predictions are explained by comparing an input to similar prototypical instances.

3 Preliminaries

3.1 Problem Definition

Let $M = \{\mathcal{G}_i\}_{i=1}^{n}$ be a set of n attributed graphs, where each graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ has a set of nodes $\mathcal{V}_i$ and a set of edges $\mathcal{E}_i$. Each node $v \in \mathcal{V}$ is associated with an attribute vector $\mathbf{x}_v \in \mathbb{R}^m$ with m attributes. We assume the majority of graphs in M are normal and can be grouped into k distinct normal clusters $C = \{C_j\}_{j=1}^{k}$, where each cluster has a prototype P_i. The anomalous graphs are few and dissimilar from the majority of the graphs in M. This process requires a similarity measure $sim(\cdot, \cdot)$ that computes the similarity between a single graph and a set (or distribution) of graphs.

Definition 1 (Graph-Level Anomaly Detection). *Graph-level anomaly detection aims to identify anomalous graphs which have characteristics different from those of the* ***majority*** *of the graphs in the given set M of graphs, and to rank all the graphs on the basis of their anomaly scores such that the anomalous graphs are ranked higher than the normal ones.*

Definition 2 (Normal and Anomalous Graphs). *Given $M = \{\mathcal{G}_i\}_{i=1}^{n}$, a set of k normal clusters $C = \{C_j\}_{j=1}^{k}$ discovered from M, and a point-set similarity function $sim(\mathcal{G}, C_j)$ that measures the similarity between a graph $\mathcal{G}$ and the distribution of cluster C_j:*

- *A graph $\mathcal{G}^N$ is* ***normal*** *if it belongs to a cluster C_j and is close to that cluster's overall distribution.*
- *A graph $\mathcal{G}^A$ is* ***anomalous*** *if it does not belong to any cluster and is dissimilar to all normal clusters' distributions.*

and they have the following relationship:

$$\max_j \left(sim(\mathcal{G}^N, C_j)\right) \gg \max_j \left(sim(\mathcal{G}^A, C_j)\right)$$

3.2 Weisfeiler-Lehman Graph Kernels

Graph kernels are a class of methods for measuring the similarity between graphs. One of the most influential embedding techniques is the Weisfeiler-Lehman (WL) scheme [8,17], which captures the dependencies in a graph in the form of subtrees. There have been several extensions and variations of the WL scheme, for instance, a different version [22] of the WL scheme has been proposed to handle attributed graphs. In this approach, each graph is encoded as a histogram that reflects the distribution of node embeddings produced by the WL process.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an attributed graph and $\mathcal{N}(v)$ be the set of nodes in the 1-hop neighborhood of node v, excluding v.

Definition 3. *A Weisfeiler-Lehman (WL) scheme [22] that embeds node v associated with $\mathcal{G}$, denoted as $\mathcal{G}(v)$, is defined as $\mathbf{x}_v^{(h)} = [\mathbf{x}_v^0, \mathbf{x}_v^1, \ldots, \mathbf{x}_v^h]$, where $\mathbf{x}_v^h$ represents the embedding of $\mathcal{G}(v)$ at iteration h:*

$$\mathbf{x}_v^h = \frac{1}{2}\big(\mathbf{x}_v^{(h-1)} + \frac{1}{|\mathcal{N}(v)|}\sum\nolimits_{u \in \mathcal{N}(v)} \mathbf{x}_u^{(h-1)}\big)$$

3.3 Isolation Kernel

Isolation kernel (IK) is a data-dependent kernel that derives directly from data without explicit learning, and it has no closed-form expression [21].

Let $D = \bigcup_{v \in \mathcal{V}_i, i \in [1,n]} \mathbf{x}_v$ be the set of all node vectors in the given dataset M of graphs. Here we use IK to map each node vector $\mathbf{x}_v \in \mathbb{R}^m$ into a vector $\varphi(\mathbf{x}_v \mid D) \in \{0,1\}^{t \times \psi}$ as is done in [25], where the use of IK has been shown to be the key to improving the discriminative power of the WL scheme.

Let $\mathbb{H}_\psi(D)$ denote the set of all partitionings H admissible from $\mathcal{D} \subset D$, where each point $\hat{\mathbf{x}}_v \in \mathcal{D}$ has equal probability of being selected from D and $|\mathcal{D}| = \psi$. Each $\theta[\hat{\mathbf{x}}_v] \in H$ isolates a point $\hat{\mathbf{x}}_v \in \mathcal{D}$.

Definition 4 ([21,15]). *The IK of* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *is defined to be the expectation taken over the distribution of partitionings* $H \in \mathbb{H}_\psi(D)$ *that both* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *fall into the same isolating partition* $\theta[\hat{\mathbf{x}}_v] \in H$*, where* $\hat{\mathbf{x}}_v \in \mathcal{D} \subset D$*,* $\psi = |\mathcal{D}|$*:*

$$\begin{aligned} \kappa_\psi(\mathbf{x}_1, \mathbf{x}_2 \mid D) &= \mathbb{E}_{\mathbb{H}_\psi(D)}\left[\mathbb{1}(\mathbf{x}_1, \mathbf{x}_2 \in \theta \mid \theta \in H)\right] \\ &\approx \frac{1}{t}\sum\nolimits_{i=1}^{t} \mathbb{1}(\mathbf{x}_1, \mathbf{x}_2 \in \theta \mid \theta \in H_i) \\ &= \frac{1}{t}\sum\nolimits_{i=1}^{t} \sum\nolimits_{\theta \in H_i} \mathbb{1}(\mathbf{x}_1 \in \theta)\mathbb{1}(\mathbf{x}_2 \in \theta) \end{aligned} \tag{1}$$

where $\mathbb{1}(\cdot)$ is an indicator function; κ_ψ is constructed using a finite number of partitionings H_i, $i = 1, \ldots, t$, where each H_i is created using randomly subsampled $\mathcal{D}_i \subset D$; and θ is a shorthand for $\theta[\hat{\mathbf{x}}_v]$.

Definition 5 (Feature map of isolation kernel. [20]). *For node vector* $\mathbf{x}_v \in \mathbb{R}^m$*, the feature mapping* $\varphi : \mathbb{R}^m \to \{0,1\}^{t\times\psi}$ *of* κ_ψ *maps* $\mathbf{x}_v$ *into a vector that represents the partitions in all the partitionings* $H_i \in \mathbb{H}_\psi(D)$*,* $i = 1, \ldots, t$*, where* $\mathbf{x}_v$ *falls into either one of the* ψ *hyperspheres or none in each partitioning* H_i*.*

Rewriting Equation 1 using φ gives:

$$\kappa_\psi(\mathbf{x}_1, \mathbf{x}_2 \mid D) = \frac{1}{t}\langle \varphi(\mathbf{x}_1 \mid D), \varphi(\mathbf{x}_2 \mid D)\rangle$$

4 Proposed Framework: ProtoGLAD

ProtoGLAD performs graph-level anomaly detection through three steps: (1) **Case representation**: each graph is embedded into a fixed-dimensional vector; (2) **Case base construction and retrieval**: a density-adaptive point-set kernel discovers normal prototypes and clusters from the dataset, and a new graph is assessed by retrieving its nearest normal cluster; (3) **Case-based explanation**: for each detected anomaly, a contrastive explanation is generated by comparing it with its nearest normal prototype at the node level. An overview of ProtoGLAD and an illustration of its contrastive explanation is provided in Fig. 1.

4.1 Graph Embedding

The Weisfeiler-Lehman (WL) embedding used here is identical to what we introduced in section 3.2. The only exception is that we use the IK mapped vector $\varphi(\mathbf{x}_v)$ rather than node vector $\mathbf{x}_v$ as the input representation of the WL scheme, which is the same as in [25].

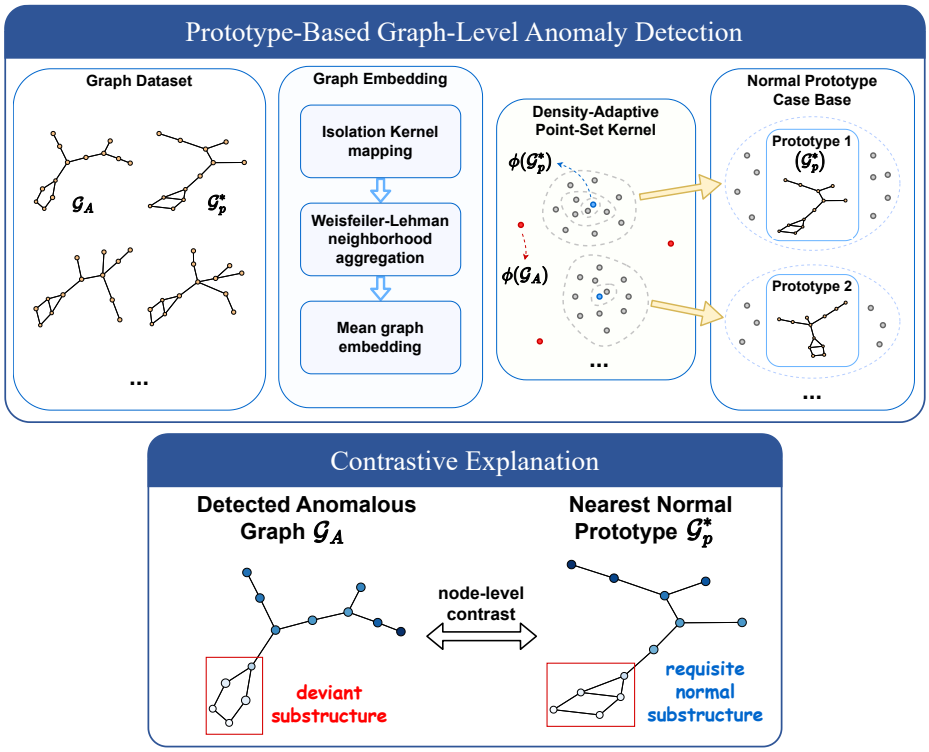


Fig. 1. An overview of ProtoGLAD and the contrastive explanation it generates.

Let $\varphi^0(\mathbf{x}_v) = \varphi(\mathbf{x}_v)$ be the IK mapped node vector $\mathbf{x}_v$ for a node v in a graph $\mathcal{G}$. The WL embedding at iteration $h > 0$ is recursively defined as:

$$\varphi^h(\mathbf{x}_v) = \frac{1}{2}\big(\varphi^{h-1}(\mathbf{x}_v) + \frac{1}{|\mathcal{N}(v)|}\sum\nolimits_{u\in\mathcal{N}(v)} \varphi^{h-1}(\mathbf{x}_u)\big) \tag{2}$$

We see each graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as having a representative sample of WL-embedded node vectors $\varphi^h(\mathbf{x}_v)$ $(\forall v \in \mathcal{V})$ of an unknown distribution, where each node vector is represented using the IK-WL embedding. Then, a mean embedding can be used to represent each graph.

For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, its embedding is given as the mean embeddings of all nodes in $\mathcal{G}$:

$$\phi(\mathcal{G}) = \frac{1}{|\mathcal{V}|}\sum\nolimits_{v\in\mathcal{V}} \varphi^h(\mathbf{x}_v).$$

The similarity between any pair of two graphs is calculated by a dot product between the two mean embeddings or feature mean maps, *i.e.*,

$$sim(\mathcal{G}, \mathcal{G}') = \langle \phi(\mathcal{G}),\ \phi(\mathcal{G}')\rangle$$

4.2 Anomalous Graph Detection

Once graph-level embeddings are obtained for each graph in the dataset, the GLAD problem can be reformulated as a point anomaly detection task in the embedding space. Each graph $\mathcal{G}_i$ is represented by a fixed-dimensional vector $\phi(\mathcal{G}_i)$ to capture its structural and semantic properties.

The point-set kernel [19] between a point x and a set C of points is:

$$K(x, C) = \langle \varphi(x), \Phi(C) \rangle$$

and

$$\Phi(C) = \frac{1}{|C|} \sum\nolimits_{y \in C} \varphi(y)$$

where $\langle a, b \rangle$ denotes a dot product between two vectors a and b; Φ is the kernel mean map of K; and φ is the feature map of a point-to-point kernel κ, for which we use IK as it has a finite-dimensional feature map, and its similarity adapts to local density of the data distribution of a given dataset.

ProtoGLAD, as detailed in Algorithm 1, is an extension of the `psKC` algorithm [19] to deal with graph data. ProtoGLAD employs the point-set kernel K to discover clusters by first locating a high-density data point, which is treated as the prototype for that cluster. A cluster is then "grown" from this prototype by absorbing all points that are similar to the cluster, based on a growth rate ρ. The process repeats for subsequent clusters using the set Π of unassigned points. It continues until Π is empty or no remaining point has a similarity to an existing cluster greater than a threshold τ. The final anomaly score for any graph $\mathcal{G}_i$ is the negative of its point-set similarity to its nearest cluster. Graphs that are most dissimilar to all clusters are thus identified as anomalies.

4.3 Contrastive Case-Based Explanation

Upon obtaining the anomaly score for a detected anomalous graph $\mathcal{G}_A$, we identify its most similar normal prototype $\mathcal{G}_p^*$. Then, for $\mathcal{G}_A$, we assign each node $u \in \mathcal{V}_A$ a score measuring how well it aligns with $\mathcal{G}_p^*$. We define the node normality score as the inner-product similarity between the node embedding and the nearest normal prototype embedding:

$$c(u) = \langle \varphi^h(u), \phi(\mathcal{G}_p^*) \rangle.$$

Intuitively, a low $c(u)$ indicates that node u contributes little to the similarity between $\mathcal{G}_A$ and the normal prototype $\mathcal{G}_p^*$, thereby highlighting this node's contribution to the anomalousness of $\mathcal{G}_A$. Analogously, scores for nodes in $\mathcal{G}_p^*$ can also be computed w.r.t. $\mathcal{G}_A$. This identifies the lowest-scored nodes in $\mathcal{G}_p^*$ that are least similar to the nodes in $\mathcal{G}_A$, *i.e.*, the normal substructure that $\mathcal{G}_A$ lacks.

The pair of $\mathcal{G}_A$ and $\mathcal{G}_p^*$, along with their respective lowest-scored nodes, provides a comprehensive, contrastive explanation of the divergence between the anomalous graph $\mathcal{G}_A$ and its most similar normal prototype $\mathcal{G}_p^*$. Specifically, this contrastive view not only highlights the deviant substructure present within $\mathcal{G}_A$

Algorithm 1 ProtoGLAD: Prototype-Based Graph-Level Anomaly Detection

Input: Graph dataset $M = \{\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)\}_{i=1}^n$, similarity threshold $\boldsymbol{\tau}$, growth rate ϱ
Output: Anomaly scores $S = \{s_i\}_{i=1}^n$, normal prototype graphs $P = \{\mathcal{G}_p^j\}_{j=1}^k$

Obtain graph embedding $g_i = \phi(\mathcal{G}_i)$ for each graph $\mathcal{G}_i$ in M, $G_{all} = \{g_i\}_{i=1}^n$
$\Pi = G_{all}$, $k = 0$
while $|\Pi| > 1$ **do**
  $g_p = \arg\max_{g \in \Pi} K(g, \Pi)$
  $g_q = \arg\max_{g \in \Pi \setminus \{g_p\}} K(g, \{g_p\})$
  $\gamma = (1 - \varrho) \times K(g_q, \{g_p\})$
  if $\gamma \leq \boldsymbol{\tau}$ **then**
    break
  end if
  $k = k + 1$
  $\mathcal{G}_p^k = g_p$, $C^k = \{g_p, g_q\}$
  while $\gamma > \boldsymbol{\tau}$ **do**
    $C^k = \{g \in \Pi \mid K(g, C^k) > \gamma\}$
    $\gamma = (1 - \varrho)\gamma$
  end while
  $\Pi = \Pi \setminus C^k$
end while
for $i = 1$ to n **do**
  $s_i = -\max_{j=1,\ldots,k} K(g_i, C^j)$
end for
return $S = \{s_i\}_{i=1}^n$, $P = \{\mathcal{G}_p^j\}_{j=1}^k$

that leads to its anomalousness, but also identifies the requisite normal substructure from $\mathcal{G}_p^*$ that $\mathcal{G}_A$ should have possessed. Notably, while the contrastive explanation produced by ProtoGLAD is similar to counterfactual explanations, it remains distinct: whereas counterfactuals focus on modifying the input to alter the model's output [24]—typically by generating synthetic instances—our approach retrieves concrete, normal prototypes directly from the case base.

5 Experiments

In this section, we evaluate ProtoGLAD by addressing three research questions:

- **RQ1.** How accurately does ProtoGLAD detect anomalous graphs?
- **RQ2.** Can ProtoGLAD provide a human-interpretable, high-quality explanation for each of its detected anomalies?
- **RQ3.** How do the key components of ProtoGLAD contribute to its overall anomaly detection effectiveness?

5.1 Experimental Setup

Datasets. We employ 10 publicly available real-world datasets from the TU-Dataset benchmark [14], covering bioinformatics (AIDS, BZR, COX2, DD, MU-

Table 1. Anomaly detection performance in terms of AUC (%). For each dataset, the best result is presented in **boldface**, and the second best result is underlined.

Dataset	WL-iForest	MUSE	GLADC	GLocalKD	SIGNET	OCGTL	GLADPro	KDGAE	ProtoGLAD
AIDS	61.4	**99.8**	96.7	96.9	97.6	97.5	52.6	83.3	98.7
BZR	52.8	68.0	61.1	68.1	77.9	61.8	60.6	55.3	**85.6**
COX2	51.2	67.3	57.7	60.2	64.5	57.9	63.7	55.2	**70.9**
DD	70.4	80.3	57.2	75.3	71.2	78.5	76.6	73.6	**81.1**
MUTAG	71.1	85.1	73.2	83.3	76.6	79.5	89.1	67.1	**89.8**
PROTEINS	62.1	72.3	75.8	76.1	73.3	70.7	73.6	58.2	**79.4**
NCI1	50.4	70.1	68.3	65.3	71.3	58.1	59.2	59.1	**72.1**
DHFR	51.7	62.5	60.4	61.8	**63.4**	51.0	60.9	52.4	62.4
IMDB-B	59.9	63.3	**65.4**	49.9	63.0	60.1	62.1	59.2	64.9
REDDIT-B	55.7	74.4	**82.1**	77.9	71.4	73.9	63.1	58.0	79.1
B-TYPE	78.1	72.6	53.0	75.9	78.6	74.7	69.9	54.8	**86.2**
B-COUNT	88.7	87.7	82.6	75.8	70.0	87.9	66.7	74.0	**91.2**
B-SIZE	76.2	60.0	64.3	74.9	65.7	79.6	69.5	63.6	**88.3**
Avg. Rank	7.00	3.77	5.46	4.54	4.23	5.23	5.69	7.69	**1.38**

TAG, PROTEINS, NCI1, DHFR) and social networks (IMDB-BINARY, REDDIT-BINARY). Following standard GLAD evaluation protocols [30,13], the minority class in a binary dataset is treated as anomalous and the majority as normal.

In addition, we construct three synthetic datasets with ground-truth explanation annotations, following [11]. Each graph is composed of a base structure (tree, ladder, or wheel with 25–35 nodes) and one or more attached motifs that determine its label. For B-TYPE (motif type), normal graphs have a house motif and anomalies have a 5-cycle motif. For B-COUNT (motif count), normal graphs have 1–2 house motifs and anomalies have 3–4. For B-SIZE (motif size), normal graphs have a cycle with 3–5 nodes and anomalies have a cycle with 6–8 nodes. Each dataset contains 500 graphs with a 10% anomaly ratio. The ground-truth anomalous substructures are defined as the nodes and edges within the motifs. Quantitative evaluations are based on 5-fold cross-validation for all datasets. [1]

Baselines. Eight unsupervised GLAD methods are selected as baselines:

- **Two-step Methods**: *WL-iForest* [17,10] and *MUSE* [6].
- **End-to-End Deep Learning Methods**: Six state-of-the-art end-to-end GLAD methods based on deep learning are used: *GLADC* [12], *GLocalKD* [13], *OCGTL* [16], *SIGNET* [11], *GLADPro* [26], and *KDGAE* [31].

Among these baselines, only SIGNET and GLADPro are capable of providing explanations for their detection results.

5.2 Anomaly Detection Performance (RQ1)

We adopt the Area Under the ROC Curve (AUC) for quantitative performance evaluation as it is a robust metric for handling the severe class imbalance inherent in GLAD tasks [13,29,30]. Higher AUC indicates better performance.

[1] Code is available at `https://github.com/IsolationKernel/ProtoGLAD`.

Table 2. Quantitative evaluation of explanations on synthetic datasets (AUC, %).

	B-TYPE		**B-COUNT**		**B-SIZE**	
Method	N-AUC	E-AUC	N-AUC	E-AUC	N-AUC	E-AUC
SIGNET	81.3	83.2	60.9	60.7	82.1	86.4
GLADPro	63.3	58.5	63.0	59.7	68.8	67.1
ProtoGLAD	**88.4**	**91.6**	**64.4**	**67.2**	**91.5**	**94.2**

As shown in Table 1, ProtoGLAD achieves the best average rank across all 13 datasets, ranking first on 9 datasets and second on 3. Two points stand out. First, ProtoGLAD consistently outperforms all baselines on molecular datasets (BZR, COX2, MUTAG, PROTEINS) with large margins, suggesting that the density-adaptive prototype discovery is effective when normal graphs form multiple distinct clusters. Second, on the two social network datasets (IMDB-B and REDDIT-B) that lack node attributes, ProtoGLAD ranks second, behind GLADC. This is expected, as the IK-WL embedding relies on node features, and its discriminative power is reduced when only structural information is available.

5.3 Case-Based Explanations (RQ2)

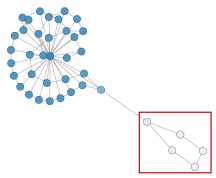
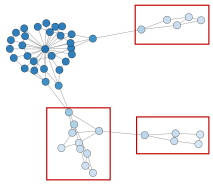
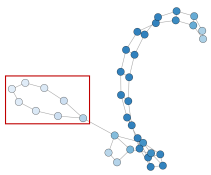
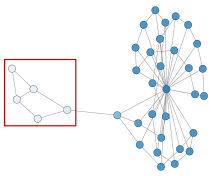
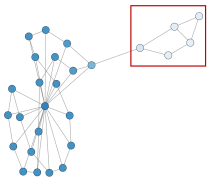
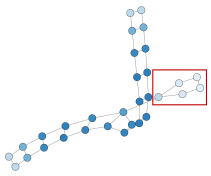
We assess the interpretability of ProtoGLAD from two aspects: (1) whether the node score $c(u)$ correctly localizes anomaly-inducing substructures on synthetic data with available node- or edge-level ground truth, and (2) whether ProtoGLAD provides more informative explanations than the two explainable GLAD methods, SIGNET and GLADPro, on real-world data.

Synthetic Data Validation. Leveraging ground-truth annotations of anomalous motifs from synthetic datasets, we evaluate the explanations quantitatively via node-level AUC (N-AUC) and edge-level AUC (E-AUC), which measure the ability to rank nodes or edges belonging to the anomalous motifs lower than normal ones [11,26]. We compare ProtoGLAD with SIGNET and GLADPro, the only two baselines capable of providing explanations.

The evaluation results are reported in Table 2. ProtoGLAD achieves the best N-AUC and E-AUC on all three datasets. The advantage is most pronounced on B-TYPE and B-SIZE, where the anomaly is characterized by a structurally distinct motif—ProtoGLAD's contrastive comparison with a concrete normal prototype directly exposes the differing substructure. On B-COUNT, where normal and anomalous graphs share the same motif type and differ only in the number of occurrences, all methods find the task more challenging, but ProtoGLAD still achieves the best results. GLADPro's relatively low scores reflect the difficulty of interpreting abstract latent prototypes at the node level.

Table 3 visualizes ProtoGLAD's case-based explanations on the three synthetic datasets, where the ground-truth anomalous motif is known. Each detected anomaly is presented alongside its corresponding retrieved normal prototype. Node colors indicate the normality score $c(u)$: darker nodes are consistent with the prototype, while lighter nodes highlight the anomaly-inducing motifs.

Table 3. Contrastive explanations produced by ProtoGLAD on the synthetic datasets. Each detected anomaly is shown alongside its retrieved normal prototype. Darker nodes indicate high similarity to the prototype; lighter nodes indicate anomaly-inducing substructures, which are highlighted by red bounding boxes.

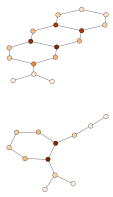
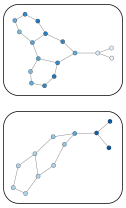
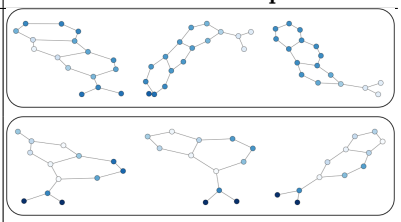
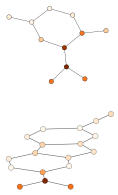
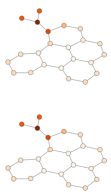
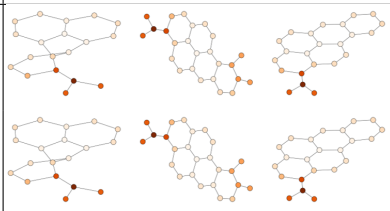
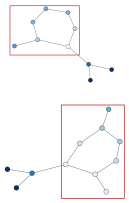
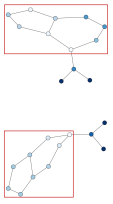
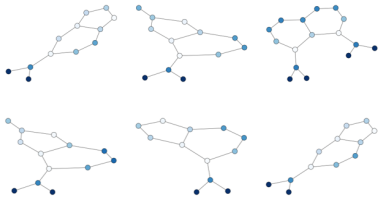
Dataset	B-TYPE	B-COUNT	B-SIZE
Anomalous Graph			
Normal Prototype			

Across all three datasets, the lighter regions align well with the ground-truth motifs, confirming that ProtoGLAD's contrastive explanation mechanism correctly identifies the substructures responsible for anomalousness.

Real-world Data Validation. We provide a qualitative comparison of the explanations produced by ProtoGLAD, SIGNET, and GLADPro on the MUTAG dataset in Table 4. For each detected anomaly, the retrieved normal prototype provides a concrete reference for comparison, and the node colors reveal where the anomaly deviates from its nearest normal case.

- **SIGNET** highlights a "bottleneck subgraph" within each anomalous graph, but does not provide a normal reference, making it difficult to verify whether the detected anomaly truly deviates from the majority. Notably, its top-1 ranked anomaly (the first row in Table 4) is identified as normal by both GLADPro and ProtoGLAD. For visual comparison, we use ProtoGLAD to obtain the normal prototype and cluster examples for the anomalous graphs detected by SIGNET. As illustrated, the top-1 anomalous graph detected by SIGNET is visually similar to its nearest normal prototype as well as to other graphs within the corresponding normal cluster. This also highlights the risk of misinterpretation when no contrastive normal reference is available.
- **GLADPro** learns global prototypes. However, as its prototypes are abstract latent vectors, they can only be visualized indirectly by retrieving a set of top-k similar graphs from the dataset. While this provides a general sense of normal patterns, no explicit summary of normal graphs is available.
- **ProtoGLAD** provides a direct, contrastive explanation for each detected anomaly. Not only does ProtoGLAD highlight the anomalous substructure, but it also finds the specific real normal prototype to which the anomaly is

Table 4. Comparison of explanations produced by SIGNET, GLADPro, and ProtoGLAD on MUTAG. For each method, we show the top-2 anomalous graphs detected, their normal prototype graphs, and additional normal reference graphs. For ProtoGLAD, prototypes are real graphs discovered from the dataset. For GLADPro, the closest graph to its latent prototype is shown. For SIGNET which does not identify prototypes, we use ProtoGLAD to provide the normal reference.

Method	Anomalous Graph	Normal Prototype	Additional Normal Reference Graphs
SIGNET			
GLADPro			
ProtoGLAD			

most similar. Node colors describe the similarity to the prototype: darker nodes indicate consistency with normality (high similarity), while **lighter nodes highlight the anomalous deviations** (low similarity).

A notable observation from Table 4 is that GLADPro incorrectly flags a ground-truth normal graph as the second detected anomaly. We attribute this to the inherent instability of its prototype learning mechanism: since its prototypes are randomly initialized learnable vectors, these vectors may drift into sparse regions that do not correspond to any dense normal distribution. This is evidenced in Table 5, where the second prototype retrieved by GLADPro is structurally more similar to the first anomalous graph shown in Table 4 rather than a typical normal graph. In contrast, ProtoGLAD explicitly identifies four concrete prototypes. Not only does ProtoGLAD capture the dominant normal pattern identified by GLADPro, but it also manages to identify others that GLADPro fails to discover. This confirms that the data-dependent approach of ProtoGLAD ensures both the *validity* and the *coverage* of the normal reference set.

Table 5. A comparison of identified prototypes on the MUTAG dataset.

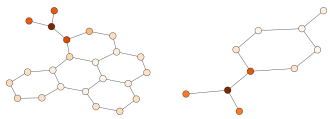
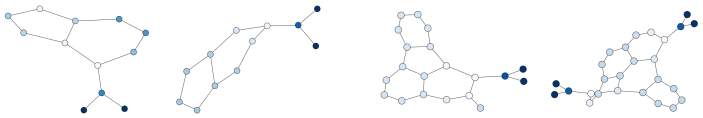
Method	Identified Prototypes
GLADPro	
ProtoGLAD	

Table 6. Performance of the variants of ProtoGLAD in terms of AUC (%). Δ denotes the average AUC drop relative to the full ProtoGLAD method.

Variant	AIDS	BZR	COX2	DD	MUTAG	PROT.	NCI1	DHFR	IMDB	REDDIT	B-TYPE	B-CNT	B-SIZE	Avg.Δ
w/o IK	95.9	67.8	55.9	78.8	86.7	77.2	67.4	60.6	62.1	72.3	85.2	83.7	85.5	−5.4
w/o WL	61.1	79.8	69.7	77.2	72.6	69.6	62.7	61.7	59.5	42.6	80.3	87.7	84.6	−10.8
w/o Proto.	98.7	55.8	46.3	20.3	12.2	32.3	31.7	59.0	51.1	76.8	80.9	87.1	81.5	−24.3
Full	**98.7**	**85.6**	**70.9**	**81.1**	**89.8**	**79.4**	**72.1**	**62.4**	**64.9**	**79.1**	**86.2**	**91.2**	**88.3**	—

5.4 Ablation Study (RQ3)

To verify the contribution of each component in ProtoGLAD, we evaluate three variants: (1) **w/o IK**: replace IK with the original node features as input to the WL scheme; (2) **w/o WL**: remove WL neighborhood aggregation and use only the IK-mapped node vectors averaged as the graph embedding; (3) **w/o Prototypes**: replace the multi-prototype clustering with a single global center.

As shown in Table 6, the multi-prototype mechanism is the most critical component in ProtoGLAD, yielding an average AUC drop of 24.3% upon its removal. This confirms that normal graphs form multiple clusters that cannot be adequately represented by a single center. Removing WL aggregation leads to an average drop of 10.8%, indicating that neighborhood structural information is essential. Removing IK causes a moderate drop of 5.4%, validating that density-adaptive similarity improves the quality of graph embeddings and case retrieval.

6 Conclusions

We present ProtoGLAD, an interpretable framework that brings case-based reasoning to the task of graph-level anomaly detection. ProtoGLAD constructs a case base of normal prototype graphs directly from the dataset using a density-adaptive point-set kernel, assesses new graphs by retrieving their nearest normal cases, and explains each detected anomaly through a contrastive comparison with its retrieved prototype, identifying substructures leading to anomalousness.

Experiments on 10 real-world and 3 synthetic datasets demonstrate that ProtoGLAD achieves the best average rank among 8 baselines in detection performance, while providing quantitatively and qualitatively superior case-based, contrastive explanations compared to existing explainable GLAD methods.

Acknowledgments. Kai Ming Ting is supported by the National Natural Science Foundation of China (Grant No. 92470116). This project is also supported by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118).

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

Declaration on Generative AI During the preparation of this work, the authors used Claude in order to assist with paraphrasing, stylistic refinement, and grammatical check. After using this tool, the authors thoroughly reviewed and edited the content as needed and take full responsibility for the final work.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications (1994)
2. Bastardo-Rojas, Á., Caro-Martínez, M.: Visual question answering to generate case-based explanations for image classification. In: Proc. of the Int. Conf. on Case-Based Reasoning. pp. 37–51. Springer (2025)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data. pp. 93–104 (2000)
4. Cai, J., Zhang, Y., Liu, F., Ng, S.K.: Leveraging diffusion model as pseudo-anomalous graph generator for graph-level anomaly detection. In: Proc. of the 42nd Int. Conf. on Machine Learning (2025)
5. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: Deep learning for interpretable image recognition. Advances in Neural Information Processing Systems (2019)
6. Kim, S., Lee, S.Y., Bu, F., Kang, S., Kim, K., Yoo, J., Shin, K.: Rethinking reconstruction-based graph-level anomaly detection: Limitations and a simple remedy. Advances in Neural Information Processing Systems **37**, 95931–95962 (2024)
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
8. Leman, A., Weisfeiler, B.: A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsiya (1968)
9. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In: Proc. of the AAAI Conf. on Artificial Intelligence (2018)
10. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proc. of the 8th IEEE Int. Conf. on Data Mining. pp. 413–422 (2008)
11. Liu, Y., Ding, K., Lu, Q., Li, F., Zhang, L.Y., Pan, S.: Towards self-interpretable graph-level anomaly detection. Advances in Neural Information Processing Systems **36**, 8975–8987 (2023)
12. Luo, X., Wu, J., Yang, J., Xue, S., Peng, H., Zhou, C., Chen, H., Li, Z., Sheng, Q.Z.: Deep graph level anomaly detection with contrastive learning. Scientific Reports **12**(1), 19867 (2022)

13. Ma, R., Pang, G., Chen, L., van den Hengel, A.: Deep graph-level anomaly detection by glocal knowledge distillation. In: Proc. of the 15th ACM Int. Conf. on Web Search and Data Mining. pp. 704–714 (2022)
14. Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M.: TU-Dataset: A collection of benchmark datasets for learning with graphs. In: ICML 2020 Workshop on Graph Representation Learning and Beyond (2020)
15. Qin, X., Ting, K.M., Zhu, Y., Lee, V.C.: Nearest-neighbour-induced isolation similarity and its impact on density-based clustering. In: Proc. of the AAAI Conf. on Artificial Intelligence. vol. 33, pp. 4755–4762 (2019)
16. Qiu, C., Kloft, M., Mandt, S., Rudolph, M.: Raising the bar in graph-level anomaly detection. arXiv preprint arXiv:2205.13845 (2022)
17. Shervashidze, N., Schweitzer, P., Van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman graph kernels. Journal of Machine Learning Research **12**(9) (2011)
18. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in Neural Information Processing Systems (2017)
19. Ting, K.M., Wells, J.R., Zhu, Y.: Point-set kernel clustering. IEEE Trans. on Knowledge and Data Engineering (2023)
20. Ting, K.M., Xu, B.C., Washio, T., Zhou, Z.H.: Isolation distributional kernel: A new tool for kernel based anomaly detection. In: Proc. of the 26th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (2020)
21. Ting, K.M., Zhu, Y., Zhou, Z.H.: Isolation kernel and its effect on SVM. In: Proc. of the 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining. pp. 2329–2337 (2018)
22. Togninalli, M., Ghisu, E., Llinares-López, F., Rieck, B., Borgwardt, K.: Wasserstein Weisfeiler-Lehman graph kernels. Advances in Neural Information Processing Systems **32** (2019)
23. Vishwanathan, S.V.N., Schraudolph, N.N., Kondor, R., Borgwardt, K.M.: Graph kernels. Journal of Machine Learning Research **11**, 1201–1242 (2010)
24. Warren, G., Delaney, E., Guéret, C., Keane, M.T.: Explaining multiple instances counterfactually: User tests of group-counterfactuals for XAI. In: Proc. of the Int. Conf. on Case-Based Reasoning. pp. 206–222. Springer (2024)
25. Xu, B.C., Ting, K.M., Jiang, Y.: Isolation graph kernel. In: Proc. of the AAAI Conf. on Artificial Intelligence. vol. 35, pp. 10487–10495 (2021)
26. Yang, Z., Zhang, G., Wu, J., Yang, J., Xue, S., Beheshti, A., Peng, H., Sheng, Q.Z.: Global interpretable graph-level anomaly detection via prototype. In: Proc. of the 31st ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (2025)
27. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems (2020)
28. Zhang, G., Yang, Z., Wu, J., Yang, J., Xue, S., Peng, H., Su, J., Zhou, C., Sheng, Q.Z., Akoglu, L., et al.: Dual-discriminative graph neural network for imbalanced graph-level anomaly detection. Advances in Neural Information Processing Systems **35**, 24144–24157 (2022)
29. Zhang, X., Qin, W., Huang, T., Gao, J.: X-GAD: Explainable graph-level anomaly detection via inter-graph anomalies. Expert Systems with Applications (2025)
30. Zhao, L., Akoglu, L.: On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights. Big Data **11**(3), 151–180 (2023)
31. Zhou, Z., Ahmad, J.M., Liu, X., He, X., Xia, J.: Graph-level anomaly detection of brain connectivity with structural interpretation and knowledge distillation. In: Proc. of the IEEE Int. Conf. on Bioinformatics and Biomedicine (2025)