ROLE-GAD: ROLE-AWARE GRAPH ANOMALY DETECTION VIA DUAL-CODEBOOK VECTOR QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph anomaly detection (GAD) is fundamental to identifying fraudulent and abnormal behaviors in financial and social networks. Unlike node classification, where classes often form relatively coherent distributions, we empirically show that graph anomalies arise from heterogeneous mechanisms and remain highly fragmented, whereas normal data exhibit fewer recurring semantic and structural patterns. These findings motivate a prototype-based representation principle: recurring normal patterns should be summarized through a compact set of semantic and relational roles, while heterogeneous anomaly mechanisms remain distinguishable through rare role assignments and atypical role interactions. Based on this principle, we propose Role-GAD, a role-aware framework that abstracts node semantics and edge interactions into dual discrete role codebooks and performs role-aware message passing over a role-induced graph augmented with global role-triplet statistics. Extensive experiments on five real-world benchmarks demonstrate that Role-GAD achieves the best average performance while converging up to $10\times$ faster than the strongest accuracy-oriented baseline. Our code and data are available at `https://anonymous.4open.science/r/role-gad-5932`.

1 INTRODUCTION

Graph anomaly detection (GAD) aims to identify rare nodes, edges, or substructures in graph-structured networks that differ significantly from the main patterns, playing a crucial role in high-risk applications such as financial fraud detection, and social media cybersecurity Chandola et al. (2009); Pang et al. (2021). In recent years, graph neural networks (GNNs) have shown immense potential in graph modeling and representation learning by capturing structural patterns Qiao et al. (2025); Liu et al. (2025), such as node classification (CLS), but face unique challenges in anomaly detection Wu et al. (2020).

Anomaly detection on graphs differs from standard node classification in two intertwined ways. First, edges are polysemous. Malicious actors routinely camouflage themselves by deliberately establishing connections with normal users Yang et al. (2025), while other anomalies aggregate into collusive subgroups. Edges incident to anomalies therefore span heterogeneous endpoint-role combinations—camouflage, collusion, and benign-looking ties—that are entangled in the same neighborhood and indistinguishable to label-agnostic message passing. Treating them identically, standard message passing dilutes the discriminative signal of an anomalous node with signals from its normal-looking neighborhood Ju et al. (2025). Second, anomalies do not form a coherent class. Some anomalies are buried inside dense normal regions through such camouflage Wu et al. (2020), while others lie in sparse peripheral areas as a result of rare or extreme deviations Yang et al. (2024). Together, these two phenomena—*edge polysemy* at the edge level and *anomaly dispersion* at the node level—expose a structural mismatch between GAD and the smooth, single-boundary representations that GNNs inherit from node classification, and raise a fundamental question: *how should graph representation learning be redesigned to cope with such heterogeneity?*

To answer this question, we first conduct a systematic empirical analysis from both feature and topological perspectives (see Sec. 2). At the feature level, we observe that anomalous nodes mostly have

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

larger same-label kNN radii than normal nodes, indicating that they usually exhibit more heterogeneous local patterns. At the topological level, normal nodes are more likely to concentrate in a narrow high-homophily regime while anomalies spread across a wide range of structural intervals. since node-wise homophily aggregates over incident edges, this directly implies that anomaly-incident edges form heterogeneous endpoint-role mixture. The two observations reveal a common asymmetry: *normal data are dominated by fewer recurring semantic and structural patterns, whereas anomalies exhibit more dataset-specific and heterogeneous organizations*. This asymmetry suggests that GAD should not rely on learning a unified representation of the anomaly class. Instead, a more suitable representation should abstract recurring regularities into reusable patterns, against which heterogeneous deviations can remain distinguishable. Prototype-based abstraction provides a natural mechanism for such compression, but prototypes alone do not expose how recurring patterns are composed across a graph. For graph reasoning, these patterns should further admit explicit identities that can be repeatedly recognized, compared, and composed across nodes and edges. We therefore represent prototypes as discrete roles: semantic roles summarize recurring node-level patterns, while interaction roles factorize recurring pairwise interaction patterns. Their discrete identities make higher-order role compositions explicitly enumerable, enabling us to characterize global structural regularities through role-triplet statistics and incorporate them into graph message passing.

Building on this representation principle, we propose Role-GAD, a novel **Role**-aware **G**raph **A**nomaly **D**etection framework that instantiates discrete role abstraction through dual-codebook vector quantization. A semantic role codebook assigns node representations to recurring semantic roles, while an interaction role codebook assigns pairwise representations to interaction roles. These discrete assignments induce a role-level graph in which each edge is characterized by a semantic–interaction–semantic triplet. Role-GAD then computes the global regularity of such triplets and incorporates it into role-aware message passing, allowing local interactions to be interpreted in the context of recurring graph-level role organizations. We conduct extensive experiments across five real-world benchmark datasets, where Role-GAD achieves the best average performance, yielding average relative improvements of 9.5% and 4.1% in AUPRC and AUROC over the strongest prior method. Our main contributions are summarized as follows:

- We perform a systematic empirical analysis of graph anomaly detection from feature and structural perspectives, revealing that on most datasets anomalies are substantially more heterogeneous than normal nodes across both semantic and relational mechanisms.
- We introduce a dual-codebook vector quantization framework Role-GAD that abstracts nodes into semantic roles, decomposes edges into interaction roles, and constructs a role-induced graph guided by global role-triplet statistics for robust anomaly modeling.
- We conduct extensive evaluations on five real-world benchmarks to demonstrate that Role-GAD achieves the best average performance while converging up to $10\times$ faster than the strongest accuracy-oriented baseline.

2 Empirical Analysis

Most existing GAD methods learn node representations by jointly exploiting node attributes and graph neighborhoods. However, anomalies may arise from diverse behaviors and interaction patterns rather than forming a coherent class, making their feature and neighborhood organizations fundamentally different from those of normal nodes. Before introducing a particular representation mechanism, we therefore first examine how normal and anomalous nodes are organized in real-world GAD benchmarks. Specifically, we conduct an empirical analysis from two complementary perspectives: (i) proximity among nodes in the attribute space, which characterizes within-label feature organization, and (ii) node-level homophily over graph edges, which characterizes label compositions in local neighborhoods.

2.1 Feature Proximity Patterns

Evaluation protocol. We measure the local scale of each label group in the node-attribute space using Euclidean distances between node attribute vectors. After node-wise ℓ_2 normalization, for each labeled node v we compute $d_k(v)$, its distance to the k-th nearest node with the same label. We use all available labeled normal and anomalous nodes in each dataset and define $r_N(k)$ and $r_A(k)$

as the median $d_k(v)$ within the two groups. We report the ratio

$$R_f(k) = \frac{r_A(k)}{r_N(k)}, \tag{1}$$

with $k = 10$ in Figure 1 and repeat the analysis at $k \in \{5, 20, 30, 50, 100\}$. Values above one indicate that same-label anomalous nodes are more dispersed in attribute space, whereas values below one indicate a compact anomaly geometry. For *Elliptic*, only nodes with known labels are included; unknown transactions are excluded from both groups.

Observation 1: Anomalous nodes usually exhibit more heterogeneous local feature patterns than normal nodes. Figure 1 shows $R_f(10) = 1.24$, 1.65, 1.41, and 2.18 on *YelpChi*, *Tolokers*, *Questions*, and *Reddit*, respectively. Thus, on four of the five datasets, anomalous nodes have larger same-label kNN radii, indicating that their feature patterns are less locally redundant. *Elliptic* is the only exception ($R_f(10) = 0.27$): its illicit transactions form unusually compact local groups, consistent with its construction from transaction snapshots in which illicit activity is temporally concentrated and shares repeated aggregate profiles, producing dense local anomaly modes rather than a dispersed geometry. The results of all five datasets remains similar across the tested neighborhood scales (Appendix C).

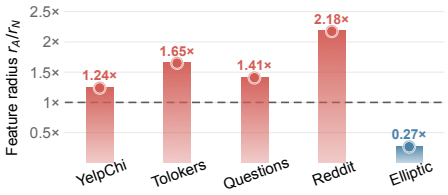


Figure 1: Attribute proximity at $k = 10$.

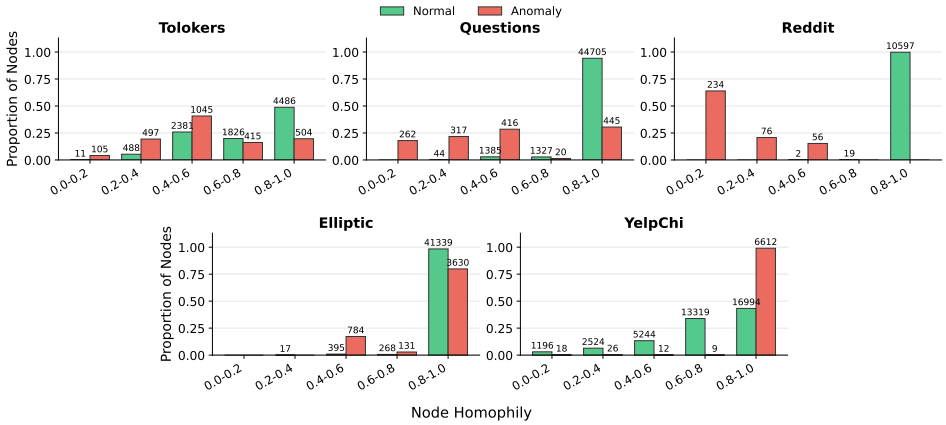


Figure 2: Node-level homophily distributions across five GAD datasets.

2.2 Structural Homophily Patterns

Evaluation protocol. We characterize local connectivity using node-level homophily $h(v) = |\{u \in \mathcal{N}(v) : y_u = y_v\}|/|\mathcal{N}(v)|$, where y_v is v's label and $\mathcal{N}(v)$ is v's neighborhood. We then partition $h(v)$ into five intervals over $[0, 1]$. In *Elliptic*, we restrict both center and neighboring nodes to those with known labels.

Observation 2: Anomalous nodes usually exhibit more heterogeneous neighborhood compositions than normal nodes. Figure 2 contrasts the node-level homophily distributions of normal and anomalous nodes. Normal nodes concentrate in a narrow high-homophily regime: more than 94% of them fall into the highest-homophily interval on *Questions*, *Reddit*, and *Elliptic*, and they remain high-leaning on *YelpChi* and *Tolokers*. Anomalous nodes, in contrast, spread across homophily intervals in dataset-specific ways: *Reddit* anomalies predominantly occupy low-homophily neighborhoods, *Tolokers* and *Questions* anomalies distribute their mass over several intervals, and *Elliptic* anomalies remain mostly homophilic but retain a non-negligible intermediate-homophily group. On most datasets, this makes the anomaly neighborhood-composition distribution markedly more dispersed than that of normal nodes. *YelpChi* is the exception in the opposite direction: its anomalies

concentrate almost entirely in the highest-homophily interval, reflecting the collusive review campaigns in which fraudulent reviewers preferentially link to one another and aggregate into tightly connected groups. The structural concentration of normals and the dataset-specific dispersion of anomalies jointly support the prototype-based view: normal nodes exhibit relatively concentrated structural roles, whereas anomalies require a more diverse set of role patterns.

Because node-level homophily summarizes the label compositions of a node's incident edges, its dispersion reveals substantial variation in how anomalous nodes interact with their neighbors. In particular, anomaly-centered neighborhoods may contain markedly different mixtures of same-label and cross-label connections across nodes and datasets. This suggests that edges incident to anomalous nodes should not be treated as homogeneous relations: their interaction patterns depend strongly on the roles of the two endpoints. Such heterogeneity motivates explicitly modeling pairwise interactions in addition to node semantics.

2.3 Implications for Role-Based Modeling

The preceding observations suggest a prototype-based view of graph anomaly detection. Obs. 1 shows that normal-node attributes exhibit fewer recurring modes than anomalous attributes, motivating the abstraction of node semantics into a compact set of representative patterns. Obs. 2 further shows that normal nodes are organized around recurring endpoint-role compositions, while anomalous nodes exhibit heterogeneous mixtures of normal-like, camouflage, and collusive relations. This motivates modeling edge interactions through a separate set of relational prototypes. Based on these findings, we represent recurring node and edge patterns as discrete semantic and interaction roles. The purpose of discretization is not to claim that continuous models are incapable of capturing such patterns; rather, discrete assignments provide explicit and stable role identities, enabling the construction of a role-induced graph. This yields two complementary design principles: node prototypes summarize recurring semantic regularities, while edge prototypes factorize heterogeneous relational mechanisms. Their combination enables role-aware message passing that suppresses camouflage-induced signal mixing and exploits global role-configuration regularities for anomaly discrimination.

3 Method

3.1 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote an attributed graph, where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathbf{X}$ represents the node feature matrix. Each node $v \in \mathcal{V}$ is associated with a binary label $y_v \in \{0, 1\}$ indicating whether the node is normal or anomalous. The goal of graph anomaly detection is to accurately identify anomalous nodes.

3.2 Latent Node and Edge Encoder

To preserve the raw distributional diversity of inputs prior to quantization, we adopt two independent encoders without neighborhood aggregation. The node encoder maps each attribute vector to a latent embedding $\mathbf{z}_v = f_{\text{enc}}^{(v)}(\mathbf{x}_v)$. For each edge $(u, v) \in \mathcal{E}$, we first form an interaction feature:

$$\mathbf{e}_{uv} = [\mathbf{x}_u \parallel \mathbf{x}_v \parallel \mathbf{x}_u \odot \mathbf{x}_v], \tag{2}$$

where $\parallel$ and $\odot$ denote concatenation and element-wise multiplication, respectively, and then encode it as $\mathbf{z}_{uv} = f_{\text{enc}}^{(e)}(\mathbf{e}_{uv})$. The resulting $\{\mathbf{z}_v\}_{v\in\mathcal{V}}$ and $\{\mathbf{z}_{uv}\}_{(u,v)\in\mathcal{E}}$ serve as inputs to the dual codebooks below.

3.3 Dual Role Vector Quantization and Role-Induced Graph Abstraction

We discretize $\{\mathbf{z}_v\}$ and $\{\mathbf{z}_{uv}\}$ via two independent codebooks, assigning each representation to its nearest prototype in the corresponding codebook.

Semantic Role Codebook We construct a semantic role codebook $\mathbf{C}^{(v)} = \{\mathbf{c}_1^{(v)}, \ldots, \mathbf{c}_{K_v}^{(v)}\}$, where each entry represents a prototypical node role capturing recurring semantic patterns. Each

node embedding z_v is mapped to its closest prototype, yielding a quantized representation $\mathbf{z}_v^q$:

$$k_v = \arg\min_k \|\mathbf{z}_v - \mathbf{c}_k^{(v)}\|_2^2, \qquad \mathbf{z}_v^q = \mathbf{c}_{k_v}^{(v)}. \tag{3}$$

Interaction Role Codebook To capture diverse relational behaviors that characterize anomalous activities, we further introduce an interaction role codebook $\mathbf{C}^{(e)} = \{\mathbf{c}_1^{(e)}, \ldots, \mathbf{c}_{K_e}^{(e)}\}$. Each latent edge representation z_{uv} is discretized as z_{uv}^q:

$$k_{uv} = \arg\min_k \|\mathbf{z}_{uv} - \mathbf{c}_k^{(e)}\|_2^2, \qquad \mathbf{z}_{uv}^q = \mathbf{c}_{k_{uv}}^{(e)}. \tag{4}$$

Role-Aware Decoding To preserve informative patterns during discretization, we introduce role-aware decoders for both semantic and interaction roles:

$$\hat{\mathbf{z}}_v = f_{\text{dec}}^{(v)}(\mathbf{z}_v^q), \qquad \hat{\mathbf{z}}_{uv} = f_{\text{dec}}^{(e)}(\mathbf{z}_{uv}^q), \tag{5}$$

with reconstruction objectives:

$$\mathcal{L}_{\text{rec}}^{(v)} = \sum_{v \in \mathcal{V}} \|\hat{\mathbf{z}}_v - \mathbf{z}_v\|_2^2, \quad \mathcal{L}_{\text{rec}}^{(e)} = \sum_{(u,v) \in \mathcal{E}} \|\hat{\mathbf{z}}_{uv} - \mathbf{z}_{uv}\|_2^2. \tag{6}$$

A commitment regularization further stabilizes role assignments:

$$\mathcal{L}_{\text{vq}} = \sum_{v \in \mathcal{V}} \|\mathbf{z}_v - \text{sg}(\mathbf{z}_v^q)\|_2^2 + \sum_{(u,v) \in \mathcal{E}} \|\mathbf{z}_{uv} - \text{sg}(\mathbf{z}_{uv}^q)\|_2^2, \tag{7}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator.

Role Codebook Updates For each role prototype k, we maintain exponentially smoothed statistics of assignment counts and embedding sums. Let n_k denote the number of latent representations assigned to prototype k in the current update, and $\mathcal{Z}_k = \sum_i \mathbf{z}_i \, \mathbb{I}(k_i = k)$ be the corresponding embedding sum, where $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition holds and 0 otherwise. We update the exponentially smoothed cluster size N_k and embedding accumulator $\mathbf{m}_k$ as:

$$N_k \leftarrow \gamma N_k + (1 - \gamma) n_k, \tag{8}$$

$$\mathbf{m}_k \leftarrow \gamma \mathbf{m}_k + (1 - \gamma) \mathcal{Z}_k, \tag{9}$$

where $\gamma \in (0, 1)$ denotes the decay coefficient (default 0.99). Each role prototype is then updated by normalized averaging as $\mathbf{c}_k = \mathbf{m}_k / \tilde{N}_k$, where $\tilde{N}_k = N_k + \epsilon$ is the Laplace-smoothed cluster size and $\epsilon = 10^{-5}$ ensures numerical stability. The same update scheme is applied to both codebooks.

Role-Induced Graph Representation Through dual role quantization, we augment the original graph with discrete role identities: $\mathcal{G}_r = (\mathcal{V}, \mathcal{E}, \mathcal{Z}^v, \mathcal{Z}^e)$, where $\mathcal{Z}^v = \{z_v^q | v \in \mathcal{V}\}$ and $\mathcal{Z}^e = \{z_{uv}^q | (u, v) \in \mathcal{E}\}$. Each node is associated with a semantic role and each edge is characterized by an interaction role, enabling role-aware graph reasoning. This discrete role abstraction explicitly compresses node semantics and factorizes pairwise interactions into discrete prototype spaces, revealing typical roles underlying normal graph behaviors.

3.4 Role-Induced Graph Reasoning

Dual vector quantization captures typical semantic and interaction prototypes at the node and edge levels. Building upon these discrete role representations, we perform role-aware message passing on the role-induced graph to model dependencies among roles. In addition, we incorporate global role triplet statistics to capture structural regularities by modeling the co-occurrence patterns.

Role Triplet Statistics Beyond local pairwise interactions, anomalous behaviors often manifest as combinations of semantic roles and interaction types. For each edge (u, v), we define a role triplet: $T_{uv} = (k_u, k_{uv}, k_v)$, where k_u and k_v denote the role types of nodes u and v, and k_{uv} denotes the role type of the edge (u, v). The global frequency of each triplet configuration is computed as:

$$\text{Count}(s_1, i, s_2) = \sum_{(u,v) \in \mathcal{E}} \mathbb{I}[k_u = s_1, \, k_{uv} = i, \, k_v = s_2], \tag{10}$$

where $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition holds and 0 otherwise. These frequencies are transformed into normalized structural priors via logarithmic smoothing:

$$w_{s_1 i s_2} = \sigma(\log(\text{Count}(s_1, i, s_2) + 1) - \mu), \tag{11}$$

where μ is the mean of non-zero log-count values and $\sigma(\cdot)$ denotes the sigmoid function. The resulting tensor $w_{s_1 i s_2}$ encodes the regularity of each role configuration, with lower values corresponding to structurally atypical patterns, thereby providing a **global view** of how role interactions are organized across the entire graph.

Role-Aware Attention and Message Passing Building upon the role-induced graph and global triplet structural priors, we perform role-aware message passing to model dependencies among semantic roles and interaction mechanisms. For each edge (u, v), the attention score jointly incorporates quantized node roles, edge roles, and the corresponding triplet regularity:

$$\alpha_{uv} = \text{LeakyReLU}\big(\mathbf{a}^\top \big[\mathbf{W}_s \mathbf{z}_u^q \parallel \mathbf{W}_t \mathbf{z}_v^q \parallel \mathbf{W}_e \mathbf{z}_{uv}^q\big] + \eta\, w_{s_1 i s_2}\big), \tag{12}$$

where $\mathbf{W}_s$, $\mathbf{W}_t$, and $\mathbf{W}_e$ are learnable projection matrices for source node roles, target node roles, and interaction roles, respectively, and $\mathbf{a}$ is the attention vector. The scalar η is a learnable parameter that balances local relational information and global structural regularity. The normalized attention coefficients are obtained via softmax over the neighborhood of v, and the updated node representation is computed as:

$$\mathbf{h}_v = \sum_{u \in \mathcal{N}(v)} \alpha_{uv} \mathbf{W}_m \mathbf{z}_u^q, \tag{13}$$

where $\mathbf{W}_m$ is a learnable transformation matrix. This role-aware attention mechanism is stacked across multiple GNN layers to enable iterative propagation of structural information over the role-induced graph.

Through this formulation, frequently observed role configurations are reinforced during message propagation, while structurally rare or anomalous configurations receive diminished influence. Consequently, the model jointly leverages local role interactions and global structural regularities to reason over role organizations.

3.5 Joint Training Objective

The final node representations are used for anomaly classification. The complete training objective combines classification loss with vector quantization losses:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_1 \mathcal{L}_{\text{rec}}^{(v)} + \lambda_2 \mathcal{L}_{\text{rec}}^{(e)} + \lambda_3 \mathcal{L}_{\text{vq}}, \tag{14}$$

where $\mathcal{L}_{\text{cls}}$ denotes a weighted cross-entropy loss to address class imbalance in anomaly detection, and λ_1, λ_2, and λ_3 are hyperparameters that balance the contributions of the node reconstruction loss, edge reconstruction loss, and vector quantization loss, respectively.

4 Experiments

4.1 Experimental Settings

Datasets We evaluate our method on five graph anomaly detection datasets spanning multiple application domains: the online review network YelpChi (Rayana & Akoglu, 2015), the crowdsourcing platform Tolokers (Platonov et al., 2023), the social media interaction graphs Questions (Platonov et al., 2023) and Reddit (Kumar et al., 2019), as well as the financial transaction network Elliptic (Weber et al., 2019). Details of datasets are provided in Appendix E.

Baselines We compare our proposed framework with four groups of baseline models: (1) **Standard Graph Neural Networks (GNNs)**, including GCN (Kipf, 2016), GIN (Xu et al., 2018), GraphSAGE (Hamilton et al., 2017) and GAT (Veličković et al., 2017); (2) **General graph autoencoder models**, GAE (Kipf & Welling, 2016), VGAE (Kipf & Welling, 2016) and VQVAE(GCN) (Van Den Oord et al., 2017). (3) **Autoencoder-based Methods for Graph Anomaly Detection**, such

as DOMINANT (Ding et al., 2019), DONE (Bandyopadhyay et al., 2020), ADONE (Bandyopadhyay et al., 2020), AnomalyDAE (Fan et al., 2020), GUIDE (Yuan et al., 2021), and GAD-NR (Roy et al., 2024) (4) **Dedicated GNN-based anomaly detection methods**, including GAS (Li et al., 2024), PCGNN (Liu et al., 2021), BWGNN (Tang et al., 2022), GHRN (Gao et al., 2023b), XGBGraph (Tang et al., 2023b), CONSISGAD (Chen et al., 2024), GODM (Liu et al., 2023), CGenGA (Ma et al., 2024), and CGADM (Wei et al., 2026). Details of all baselines are provided in Appendix F.

Settings For each dataset, we randomly split nodes into 20% for training, 10% for validation, and 70% for testing, as in CGADM (Wei et al., 2026). Following the standard evaluation protocol for graph anomaly detection (Han et al., 2022), we report AUROC and AUPRC as performance metrics. All experiments are repeated ten times with different random splits, and the averaged results are reported. The corresponding standard deviations for Role-GAD are reported in Appendix D.3. Detailed implementation settings are provided in Appendix G.

Table 1: Performance comparison on GAD benchmarks. The **best** and runner-up results are highlighted in bold and underlined, respectively.

Model	YelpChi		Tolokers		Questions		Reddit		Elliptic		Average	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
Standard GNNs												
GCN	23.59	59.89	41.44	73.58	10.27	67.73	5.65	62.55	80.19	95.12	32.23	71.77
GIN	38.13	77.40	37.89	74.02	11.23	68.07	5.38	65.25	83.88	96.21	35.30	76.19
GraphSAGE	50.23	83.24	43.73	77.30	13.86	70.64	5.78	63.67	86.16	96.61	39.95	78.29
GAT	46.64	80.95	42.18	76.66	13.19	68.19	5.42	63.55	87.59	97.11	39.00	77.29
General Graph Autoencoder Models												
GAE	23.98	59.88	42.49	74.19	9.55	58.90	6.21	63.54	83.72	95.65	33.19	70.45
VGAE	26.67	62.93	43.18	75.04	12.43	70.97	8.38	67.95	83.03	95.47	34.74	74.47
VQVAE(GCN)	24.26	61.73	34.69	71.09	8.38	61.20	5.90	65.63	80.37	95.46	30.72	71.02
Autoencoder-based Methods for GAD												
DOMINANT	86.69	97.19	35.37	70.07	11.55	64.94	4.67	58.52	74.01	93.66	42.46	76.88
DONE	31.21	72.42	36.43	70.67	10.39	59.67	5.75	61.63	64.03	91.01	29.56	71.08
ADONE	31.03	71.64	35.27	70.08	10.75	61.82	5.44	60.87	65.54	91.04	29.61	71.09
AnomalyDAE	39.74	78.01	36.22	72.62	10.31	66.84	4.87	62.67	89.87	97.49	36.20	75.53
GUIDE	91.48	97.45	40.85	73.00	9.15	62.66	5.57	63.92	80.99	94.13	45.61	78.23
GAD-NR	69.10	90.92	40.88	75.69	10.07	66.94	4.81	59.34	58.30	92.90	36.63	77.16
Dedicated GNN-based Anomaly Detection Methods												
GAS	39.18	78.63	42.39	74.55	12.41	66.09	5.66	61.23	87.54	97.14	37.44	75.53
PCGNN	45.32	79.61	36.76	71.28	13.79	69.12	4.13	54.58	67.29	93.88	33.46	73.69
BWGNN	49.15	81.85	45.02	77.80	14.64	69.96	5.42	60.63	87.90	96.99	40.43	77.45
GHRN	49.78	82.36	45.25	77.98	14.61	69.32	5.85	63.51	88.13	97.04	40.72	78.04
XGBGraph	75.91	91.85	44.47	77.28	14.33	64.90	4.59	60.58	90.47	94.35	45.95	77.79
CONSISGAD	41.74	79.35	40.59	76.03	12.85	70.54	5.57	66.99	86.42	96.38	37.43	77.86
GODM	51.77	84.33	46.15	76.42	15.11	68.86	5.55	62.10	85.89	93.92	40.89	77.13
CGenGA	52.76	85.65	44.89	78.95	15.34	68.46	5.78	64.78	87.36	96.07	41.23	78.78
CGADM	76.54	92.69	46.02	79.68	18.51	69.41	5.79	65.85	**97.03**	**99.34**	48.78	81.39
Role-GAD	**94.90**	**98.64**	**50.08**	**80.74**	**18.74**	**77.23**	**11.45**	**69.43**	91.96	97.59	**53.43**	**84.73**

4.2 Overall Comparison

Table 1 reports the performance of Role-GAD against a broad suite of baselines on five graph anomaly detection benchmarks. We make the following key observations: (1) Overall, Role-GAD achieves the *best macro-average* performance across benchmarks, reaching 53.43% AUPRC and 84.73% AUROC, with relative improvements of 9.5% in AUPRC and 4.1% in AUROC over the strongest prior method. In particular, on the *Reddit* dataset, which exhibits highly diverse interaction distributions, Role-GAD achieves an AUPRC of 11.45%, corresponding to a 37% relative improvement over the strongest baseline. These results demonstrate that role abstraction provides an effective modeling paradigm for graph anomaly detection, especially in scenarios with complex and fragmented anomaly distributions. (2) Although VQVAE(GCN) applies quantization, its average AUPRC (30.72%) is significantly lower than Role-GAD (53.43%), highlighting the necessity of our dual-codebook design and global triplet-level structural regularization. (3) The gains are consistent with the empirical analysis in Section 2: on most datasets anomalies exhibit more heterogeneous

organizations than normal nodes and are therefore poorly characterized by a single recurring pattern. This is reflected in the results, where Role-GAD is the strongest method on four of the five benchmarks and remains competitive on *Elliptic*, whose anomalies instead form compact, recurring transaction groups.

Convergence efficiency We further compare the wall-clock time required to reach each method's validation-selected checkpoint on the two edge-dense datasets. As shown in Table 2, Role-GAD reaches higher test AUROC substantially earlier than CGADM on both datasets. In particular, on YelpChi, Role-GAD reaches 97.88% AUROC in approximately 255 seconds, compared with 92.69% in approximately 2,700 seconds for CGADM, corresponding to over a 10× reduction in time to the selected checkpoint. Full comparisons with additional baselines and the timing protocol are provided in Appendix D.1.

Table 2: Convergence efficiency on edge-dense datasets.

Method	AUROC	Time (s)
YelpChi		
CGADM	92.69	∼2700
Role-GAD	**97.88**	∼**255**
Tolokers		
CGADM	79.68	∼631
Role-GAD	**81.70**	∼**30**

4.3 Role Codebook Interpretability

We analyze the learned roles on *Elliptic*, using a semantic codebook with 1024 prototypes and an interaction codebook with 4 prototypes. We examine role–label association at two granularities: individual semantic roles at the node level, and role triplets at the interaction level.

Node-level semantic roles. Figure 3 relates the usage count of each semantic role to its anomaly ratio, where each point is one semantic-codebook entry. Most frequently used roles exhibit anomaly ratios close to zero, indicating that the corresponding prototypes are predominantly associated with normal nodes. In contrast, a small subset of sparsely populated roles has high anomaly ratios, showing that anomalous nodes concentrate in or mix into particular regions of the learned discrete role space. This association is already visible before role-induced message passing, suggesting that the semantic codebook itself carries label-relevant structure that the subsequent propagation further refines.

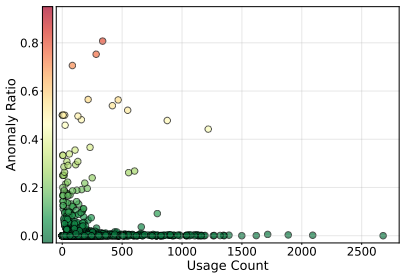


Figure 3: Anomaly ratios and usage counts of semantic roles on *Elliptic*.

Interaction-level role triplets. For each ground-truth interaction type—Normal–Normal, Anomaly–Anomaly, and Anomaly–Normal—we select the 15 most frequent role triplets and examine their occurrence counts and compositions (Figure 4). High-frequency triplets from Normal–Normal interactions are dominated by normal edges, whereas triplets from Anomaly–Anomaly interactions have lower support but high anomaly concentration, and Anomaly–Normal triplets show more mixed compositions. These patterns indicate that the codebooks capture recurring interaction configurations tied to different combinations of the two interacting nodes' labels, although they do not by themselves establish behavioral semantics such as camouflage or collusion.

This role-level view echoes the feature-space analysis in Section 2: *Elliptic* was the exception whose anomalies form compact local groups (Observation 1), and here those anomalies resolve into a few sparsely populated, anomaly-heavy roles and triplets rather than a single prototype. The two perspectives agree that even *Elliptic*'s clustered anomalies decompose into multiple recurring dense modes—heterogeneity that a continuous representation tends to average away, but that discrete roles preserve as explicit, reusable identities. This is precisely the regularity Role-GAD is built to expose: it recasts entangled, camouflaged anomaly patterns into a compact vocabulary of semantic and interaction roles on which anomaly discrimination can directly act.

4.4 Ablation Study

We conduct an ablation study to examine the contributions of semantic role modeling, interaction role modeling, and global role-triplet modeling. Since global modeling operates on role triplets

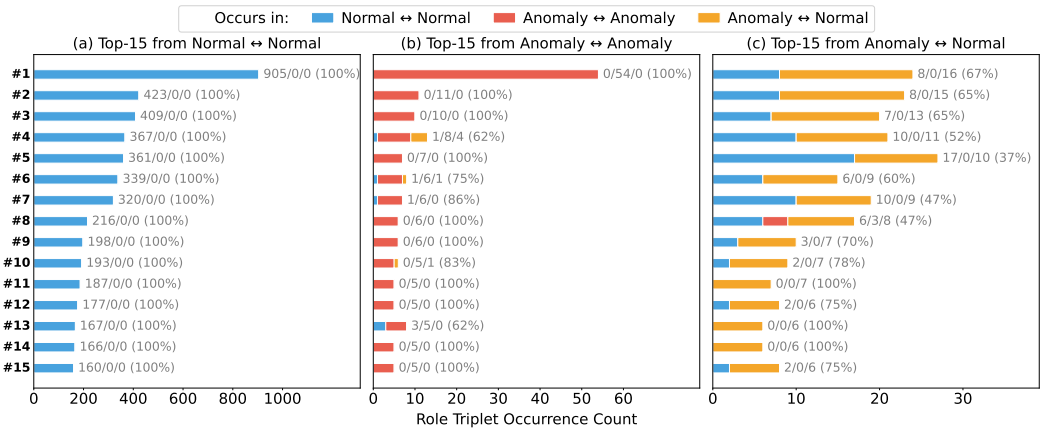


Figure 4: Occurrence distributions of top-15 role triplets selected from each interaction type.

jointly defined by semantic and interaction role assignments, it is only applicable when both codebooks are present. Table 3 therefore explicitly reports the components retained in each variant.

We assess each component through controlled comparisons. First, comparing Semantic Role Only with Dual Roles isolates the contribution of interaction role modeling. Introducing interaction roles yields substantial improvements across all three datasets, with a particularly large AUPRC gain on YelpChi (18.57% to 90.66%), demonstrating the importance of explicitly modeling pairwise interaction patterns beyond node-level semantics. Second, comparing Interaction Role Only with Dual Roles isolates the additional contribution of semantic role modeling. Semantic roles provide consistent, though smaller, improvements across the three datasets. Finally, comparing Dual Roles with the full Role-GAD isolates the contribution of global role-triplet modeling, which consistently improves both AUPRC and AUROC across all datasets. Together, these controlled comparisons demonstrate the complementary contributions of semantic roles, interaction roles, and their global role-triplet organization.

Table 3: Ablation study of Role-GAD. S, I, and G denote semantic role modeling, interaction role modeling, and global role-triplet modeling, respectively.

	Components			YelpChi		Tolokers		Elliptic	
Setting	S	I	G	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC
Interaction Role Only	×	✓	×	90.37	97.21	48.15	80.41	87.15	96.18
Semantic Role Only	✓	×	×	18.57	51.35	46.41	79.93	85.83	96.25
Dual Roles	✓	✓	×	90.66	97.44	48.63	80.46	90.97	96.77
Role-GAD	✓	✓	✓	**94.90**	**98.64**	**50.08**	**80.74**	**91.96**	**97.59**

Additional analyses. We provide further results on the GADBench evaluation protocol (Appendix D.2), training and design analyses (Appendices D.5–D.6), and hyperparameter sensitivity (Appendices D.7, D.8, and D.9).

5 Conclusion

In this work, we showed that graph anomaly detection differs from conventional node classification in that normal patterns are relatively recurrent, whereas anomalies remain fragmented across heterogeneous semantic and relational mechanisms. Based on these findings, we introduced a prototype-based role abstraction principle and instantiated it through Role-GAD, which maps node semantics and edge interactions into dual discrete role spaces and performs global role-level reasoning over a role-induced graph. Extensive experiments demonstrate that our method achieves the best average performance and favorable convergence efficiency across diverse benchmarks, and that the learned roles are interpretable.

AI USE STATEMENT

We did not use generative AI tools for any tasks requiring mandatory disclosure under the ICLR 2027 AI Policy for Authors. We used generative AI tools to revise and polish the manuscript for clarity and readability, and to assist with literature discovery. All AI-suggested references were independently verified for their existence and relevance, and the original papers were reviewed by the authors before citation. All AI-assisted content was reviewed and validated by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

This work develops a graph anomaly detection method and evaluates it exclusively on existing public benchmark datasets; we do not collect new user data or conduct studies involving human subjects. As graph anomaly detection can be applied to sensitive domains such as fraud detection, online platform moderation, and cybersecurity, we acknowledge that erroneous predictions may lead to false accusations or other adverse consequences if model outputs are used directly for real-world decision making. Therefore, Role-GAD should be regarded as a decision-support technique rather than a substitute for appropriate human review and domain-specific validation. Our experiments are intended to evaluate the methodological effectiveness of the proposed approach under established benchmark settings, rather than to make claims about individual users or entities in real-world systems. We follow the usage conditions of the publicly available datasets and report results in aggregate without attempting to identify or infer information about individual entities.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The complete formulation of Role-GAD, including the dual role codebooks, role-induced graph construction, role-triplet statistics, message-passing mechanism, and training objective, is provided in Section 3. Section 4 describes the datasets, evaluation protocol, baseline methods, implementation settings, and evaluation metrics used in our experiments. We report results over multiple random splits and provide ablation, sensitivity, and efficiency analyses to examine the behavior and robustness of the proposed method. The source code and data necessary to reproduce our experiments are provided through the anonymous repository accompanying this submission. Additional implementation details and experimental configurations are included in the supplementary materials.

REFERENCES

Charu C Aggarwal. An introduction to outlier analysis. In *Outlier analysis*, pp. 1–34. Springer, 2016.

Sambaran Bandyopadhyay, Lokesh N, Saley Vishal Vivek, and M Narasimha Murty. Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proceedings of the 13th international conference on web search and data mining*, pp. 25–33, 2020.

Yoann Boget, Magda Gregorova, and Alexandros Kalousis. Discrete graph auto-encoder. *arXiv preprint arXiv:2306.07735*, 2023.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

Nan Chen, Zemin Liu, Bryan Hooi, Bingsheng He, Rizal Fathony, Jun Hu, and Jia Chen. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *The twelfth international conference on learning representations*, 2024.

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

Mucong Ding, Kezhi Kong, Jingling Li, Chen Zhu, John Dickerson, Furong Huang, and Tom Goldstein. Vq-gnn: A universal framework to scale up graph neural networks using vector quantization. *Advances in Neural Information Processing Systems*, 34:6733–6746, 2021.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

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

Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pp. 357–365, 2023a.

Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM web conference 2023*, pp. 1528–1538, 2023b.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. *Advances in neural information processing systems*, 35:32142–32159, 2022.

Xuanwen Huang, Yang Yang, Yang Wang, Chunping Wang, Zhisheng Zhang, Jiarong Xu, Lei Chen, and Michalis Vazirgiannis. Dgraph: A large-scale financial dataset for graph anomaly detection. *Advances in Neural Information Processing Systems*, 35:22765–22777, 2022.

Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

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

Xuan Li, Chunjing Xiao, Ziliang Feng, Shikang Pang, Wenxin Tai, and Fan Zhou. Controlled graph neural networks with denoising diffusion for anomaly detection. *Expert systems with applications*, 237:121533, 2024.

Kay Liu, Hengrui Zhang, Ziqing Hu, Fangxin Wang, and Philip S Yu. Data augmentation for supervised graph outlier detection with latent diffusion models. *arXiv preprint arXiv:2312.17679*, 2023.

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

Zemin Liu, Yuan Li, Nan Chen, Qian Wang, Bryan Hooi, and Bingsheng He. A survey of imbalanced learning on graphs: Problems, techniques, and future directions. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Xiaoxiao Ma, Ruikun Li, Fanzhen Liu, Kaize Ding, Jian Yang, and Jia Wu. Graph anomaly detection with few labels: A data-centric approach. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2153–2164, 2024.

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

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

Hezhe Qiao and Guansong Pang. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. *Advances in Neural Information Processing Systems*, 36:49490–49512, 2023.

Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. Deep graph anomaly detection: A survey and new perspectives. *IEEE Transactions on Knowledge and Data Engineering*, 37(9):5106–5126, 2025.

Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pp. 985–994, 2015.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Amit Roy, Juan Shu, Jia Li, Carl Yang, Olivier Elshocht, Jeroen Smeets, and Pan Li. Gad-nr: Graph anomaly detection via neighborhood reconstruction. In *Proceedings of the 17th ACM international conference on web search and data mining*, pp. 576–585, 2024.

Tarek Seghair, Olfa Besbes, Takoua Abdellatif, and Sami Bihiri. Vq-vgae: Vector quantized variational graph auto-encoder for unsupervised anomaly detection. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 2370–2375. IEEE, 2024.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, pp. 21076–21089. PMLR, 2022.

Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

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

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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

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

Chunyu Wei, Haozhe Lin, Yueguo Chen, and Yunhai Wang. Conditional diffusion anomaly modeling on graphs. *Advances in Neural Information Processing Systems*, 38:153459–153495, 2026.

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.

Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z Li. Mole-bert: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*, 2023.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Chengdong Yang, Hongrui Liu, Daixin Wang, Zhiqiang Zhang, Cheng Yang, and Chuan Shi. Flag: Fraud detection with llm-enhanced graph neural network. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5150–5160, 2025.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

Qien Yu, Shengxin Dai, Ran Dong, and Soichiro Ikuno. Attention-based vector quantized variational autoencoder for anomaly detection by using orthogonal subspace constraints. *Pattern Recognition*, 164:111500, 2025.

Xu Yuan, Na Zhou, Shuo Yu, Huafei Huang, Zhikui Chen, and Feng Xia. Higher-order structure based anomaly detection on attributed networks. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 2691–2700. IEEE, 2021.

Long Zeng, Jianxiang Yu, Jiapeng Zhu, Qingsong Zhong, and Xiang Li. Hierarchical vector quantized graph autoencoder with annealing-based code selection. In *Proceedings of the ACM on Web Conference 2025*, pp. 3772–3782, 2025.

A Overall Framework Illustration

Figure 5 provides an overview of the motivation and overall architecture of Role-GAD. It contrasts the structural characteristics of conventional node classification and graph anomaly detection, and illustrates how Role-GADinstantiates discrete role abstraction through dual-codebook vector quantization.

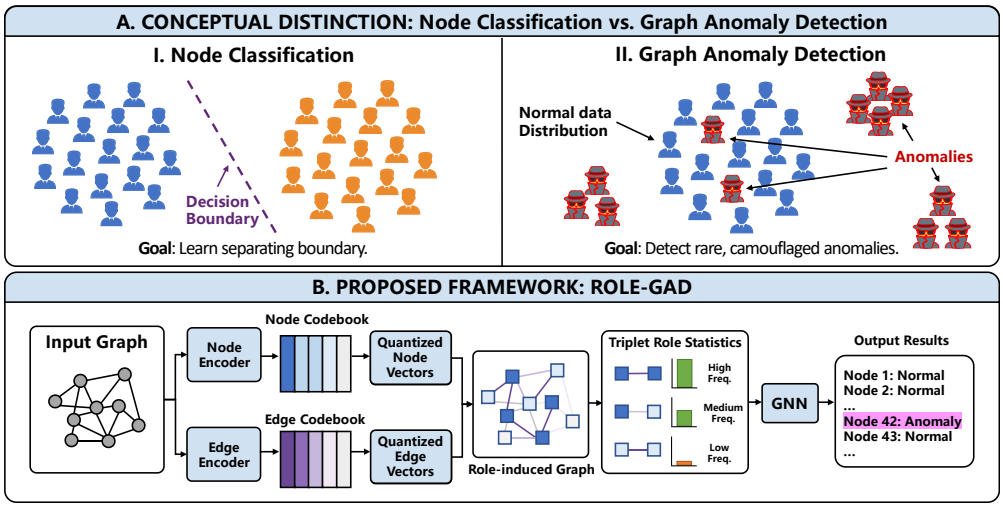


Figure 5: Illustration of graph anomaly detection and the proposed Role-GAD framework. (A) Conceptual contrast between node classification, where classes form compact clusters with smooth decision boundaries, and graph anomaly detection, where anomalies are dispersed and intertwined with dominant normal patterns. (B) The Role-GAD framework discretizes node semantics and edge interaction patterns via dual codebooks, constructs a role-induced graph augmented with global triplet statistics, and performs role-aware message passing for anomaly detection.

B Related Work

B.1 Graph Anomaly Detection

Graph anomaly detection (GAD) focuses on identifying rare and irregular graph patterns and has been widely applied in real-world scenarios such as fraud detection, spam review identification, and network security (Gao et al., 2023a; Huang et al., 2022; Qiao & Pang, 2023; Tang et al., 2023b). In contrast to standard graph learning tasks that model majority patterns, GAD is characterized by extreme class imbalance and complex anomaly mechanisms arising from diverse attributes, irregular

structures, and high-order interactions (Aggarwal, 2016; Chandola et al., 2009; Pang et al., 2021; Liu et al., 2025). To address these challenges, existing GAD methods have explored diverse modeling paradigms with progressively enhanced expressive power. Reconstruction-based autoencoders model normal patterns and detect anomalies via reconstruction errors, such as AnomalyDAE (Fan et al., 2020), GUIDE (Yuan et al., 2021), and GAD-NR (Roy et al., 2024). Beyond reconstruction, discriminative graph neural networks directly learn decision boundaries to separate anomalous nodes under supervised or weakly supervised learning, exemplified by GAS (Li et al., 2024) and PCGNN (Liu et al., 2021). To capture irregularities that neighborhood smoothing tends to overlook, spectral-domain approaches such as BWGNN (Tang et al., 2022) and GHRN (Gao et al., 2023b) emphasize high-frequency structural signals. Further, generative frameworks, exemplified by CGADM (Wei et al., 2026) and GODM (Liu et al., 2023), explicitly model graph data distributions for anomaly scoring. Despite the diversity, most existing GAD methods encode all nodes in continuous latent spaces and optimize reconstruction or discriminative objectives without explicitly abstracting recurring normal patterns or factorizing heterogeneous edge interactions. When combined with label-agnostic neighborhood aggregation, this representation objective may obscure heterogeneous anomaly mechanisms through signal mixing.

B.2 Vector Quantized Variational Autoencoder

Vector Quantized Variational Autoencoder (VQ-VAE) (Van Den Oord et al., 2017) introduces discrete prototype-based latent representations via a learned codebook and has demonstrated remarkable success in computer vision for modeling diverse and compositional data structures (Razavi et al., 2019; Esser et al., 2021). Inspired by these advances, several recent studies extend VQ-VAE-style discrete encoding to graph neural networks for representation learning (Zeng et al., 2025). For example, VQ-GNN (Ding et al., 2021) explores vector quantization mainly as a dimensionality reduction tool. Mole-BERT (Xia et al., 2023) and DGAE (Boget et al., 2023) apply VQ-VAE to molecular graph representation learning. Despite this progress in graph representation learning, discrete latent representations remain underexplored for graph anomaly detection. To the best of our knowledge, only a few studies, such as VQ-VGAE (Seghair et al., 2024), directly adopt VQ-based encoders for graph anomaly detection by optimizing reconstruction objectives. However, these approaches primarily treat VQ-VAE as a generic graph autoencoder and fail to fully exploit the semantic prototype abstraction and role interpretability enabled by discrete codebooks. Beyond graph domains, recent advances in computer vision anomaly detection further highlight the advantages of VQ-VAE-style discrete representations, where prototype-based latent spaces effectively mitigate reconstruction shortcuts and better capture diverse abnormal patterns (Esser et al., 2021; Yu et al., 2025). These findings highlight the promise of discrete prototype modeling for anomaly detection, yet such approaches remain largely unexplored on graphs, despite the multi-modal nature of real-world anomalies.

C Sensitivity of Feature-Space Proximity to k

We examine whether the feature-space pattern in Figure 1 depends on the neighbor rank k. Figure 6 repeats the full-population same-label analysis at $k \in \{5, 20, 30, 50, 100\}$, spanning small to large neighborhood scales. The four-dataset contrast is stable across every k: ***YelpChi*** ($R_f \approx 1.24$), ***Tolokers*** (1.44–1.94), ***Questions*** (1.39–1.43), and ***Reddit*** (1.84–2.19) retain ratios above one, whereas ***Elliptic*** stays below one at every scale ($R_f = 0.22$–0.43). The Elliptic ratio drifts mildly upward with k as the neighborhood grows beyond its dense local anomaly cores, but never crosses one. This stability indicates that the higher anomaly radius on the four datasets and the compact Elliptic geometry are not artifacts of the choice $k = 10$.

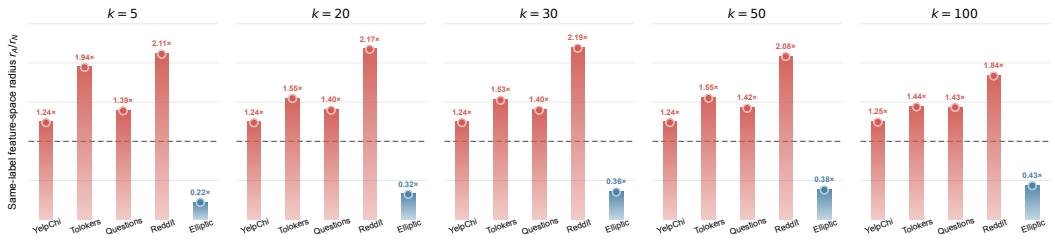


Figure 6: Sensitivity of full-population node-attribute proximity to the neighbor rank. Each panel reports $R_f(k) = r_A(k)/r_N(k)$ for $k \in \{5, 20, 30, 50, 100\}$; anomaly-larger ratios ($R_f > 1$) are shown in red and the compact *Elliptic* geometry ($R_f < 1$) in blue. The direction of every dataset is preserved across all neighborhood scales.

D Additional Robustness and Evaluation Results

We organize the supplementary experiments by their relevance to the main empirical claims. We first examine performance under the standard GADBench protocol and quantify variability across random splits. We then evaluate whether loss weighting drives the results and analyze training stability and individual design choices. Hyperparameter sensitivity results are presented last.

D.1 Convergence Efficiency

We evaluate convergence efficiency on two edge-dense benchmarks, *YelpChi* (3.8M edges) and *Tolokers* (519K edges), where graph propagation and edge-level modeling incur costs proportional to graph size. All methods are trained under the same hardware and experimental settings. After each epoch, we evaluate the current model on the validation set, select the checkpoint with the highest validation AUROC, record the elapsed training time required to reach it, and evaluate that checkpoint once on the test set. Figure 7 plots the resulting test AUROC against its corresponding time to the selected checkpoint.

Role-GAD provides a favorable accuracy–time trade-off on both datasets. On *YelpChi*, it reaches a test AUROC of 97.88% in approximately 255s, whereas CGADM reaches 92.69% in approximately $2{,}700$s; thus, Role-GAD reaches its validation-selected checkpoint around $10\times$ sooner while obtaining higher test AUROC. XGBGraph reaches its checkpoint sooner because it uses a non-neural optimization pipeline, but Role-GAD improves test AUROC by approximately six points. On *Tolokers*, Role-GAD reaches 81.70% test AUROC in approximately 30s, compared with 79.68% in approximately 631s for CGADM. These results measure time to each method's own validation-selected checkpoint, rather than time to a common target accuracy or asymptotic scaling with graph size.

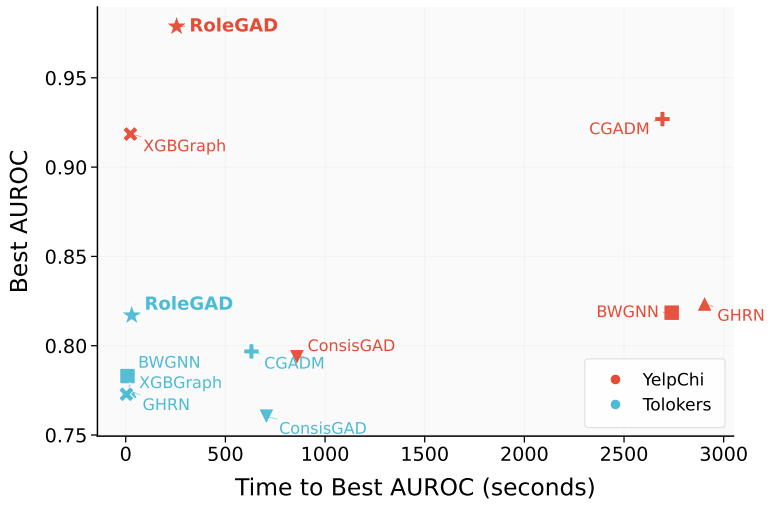


Figure 7: Test AUROC of the validation-selected checkpoint versus the wall-clock time required to reach it on *YelpChi* and *Tolokers*. Upper-left is better.

D.2 Evaluation under the GADBench Protocol

The main comparison follows the 20/10/70 train/validation/test protocol used by CGADM. To determine whether the results depend on that particular split, we additionally evaluate Role-GAD using the 40/20/40 protocol adopted by GADBench (Tang et al., 2023a). This complementary evaluation covers the five main datasets together with *Weibo* and *Amazon*. For each dataset, Table 4 identifies the strongest reported GADBench baseline used for comparison and reports its AUPRC alongside the AUPRC of Role-GAD.

Table 4: AUPRC comparison under the GADBench 40/20/40 protocol. "Best GADBench baseline" denotes the strongest result reported in GADBench for each dataset. NA denotes Neighborhood Averaging. Gain is the absolute AUPRC improvement in percentage points.

	Best GADBench baseline			
Dataset	Method	AUPRC	Role-GAD AUPRC	Gain (pp)
Elliptic	RF-Graph	78.86	**94.54**	**+15.68**
YelpChi	XGB-Graph	91.11	**95.02**	**+3.91**
Tolokers	XGB-Graph	53.92	**57.67**	**+3.75**
Questions	BWGNN	18.57	**20.09**	**+1.52**
Reddit	NA	9.70	**13.88**	**+4.18**
Weibo	XGB-Graph	97.06	**98.05**	**+0.99**
Amazon	XGB-Graph	93.33	**93.52**	**+0.19**

Under this protocol, Role-GAD obtains higher AUPRC than the listed GADBench result on all seven datasets. The absolute gains are largest on *Elliptic* and *Reddit*, whereas the margins on *Weibo* and *Amazon* are comparatively small. These results show that the method remains competitive under a second, widely used split protocol and is not tied exclusively to the 20/10/70 setting. However, the comparator values are reported GADBench results rather than fully matched reruns on identical random splits. The results are therefore complementary to Table 1, should not be compared numerically across protocols, and should not be interpreted as a paired significance test.

D.3 Uncertainty across Random Splits

To assess robustness across data splits, Table 5 reports the mean and standard deviation of Role-GAD over ten independent random splits under the 20/10/70 protocol. Role-GAD exhibits stable

performance across most datasets, with standard deviations below one percentage point for both AUPRC and AUROC on *YelpChi*, *Elliptic*, *Tolokers*, and *Questions*. On *Reddit*, the AUROC shows relatively higher variability, with a standard deviation of 2.49 points. These results indicate that the overall performance of Role-GAD is generally robust to different data splits, while the results on *Reddit* should be interpreted with greater uncertainty.

Table 5: Mean $\pm$ standard deviation over ten random splits.

Dataset	AUROC	AUPRC
YelpChi	98.64 ± 0.23	94.90 ± 0.63
Elliptic	97.59 ± 0.21	91.96 ± 0.48
Tolokers	80.74 ± 0.49	50.08 ± 0.91
Questions	77.23 ± 0.54	18.74 ± 0.64
Reddit	69.43 ± 2.49	11.45 ± 0.88

D.4 Weighted versus Unweighted Classification Loss

The main objective uses class-weighted cross-entropy to address label imbalance. To determine whether the observed performance is primarily attributable to this loss choice rather than to the proposed architecture, we compare weighted and unweighted cross-entropy while holding Role-GAD fixed. Table 6 shows mixed effects: the weighted loss is slightly lower on *YelpChi* and *Elliptic*, nearly unchanged on *Tolokers* and *Questions*, and 1.64 AUPRC points higher on *Reddit*. Thus, weighting is not uniformly responsible for the gains and is most consequential on *Reddit* among the tested datasets. This within-model comparison does not by itself establish the fairness of baseline implementations or resolve differences between supervised and originally unsupervised methods.

Table 6: AUPRC with weighted and unweighted cross-entropy.

Dataset	Weighted	Unweighted	Difference
YelpChi	93.16	93.61	-0.45
Elliptic	90.58	90.94	-0.36
Tolokers	50.09	50.06	$+0.03$
Questions	17.98	18.04	-0.06
Reddit	12.61	10.97	$+1.64$

D.5 Training-Time Codebook Utilization

Vector-quantized models can suffer from codebook collapse, in which only a small fraction of entries remain active. Role-GAD uses two stabilization mechanisms: inactive entries are re-initialized from encoder outputs, and codebooks are updated using exponential moving averages with Laplace smoothing. To examine the resulting training behavior, we track the fraction of entries that receive at least one assignment during training on *YelpChi*, using a node codebook of size 256 and an edge codebook of size 64. Table 7 reports utilization at representative epochs.

Table 7: Codebook utilization during training on *YelpChi*.

Epoch	Node codebook (%)	Edge codebook (%)
1	100.0	100.0
10	84.2	88.4
50	98.2	96.9
100	95.5	100.0
200	98.2	99.7
300	98.9	100.0
500	100.0	100.0

Both codebooks are fully utilized at initialization. Their utilization temporarily decreases at epoch 10 to 84.2% for the node codebook and 88.4% for the edge codebook, but subsequently recovers to above 95% from epoch 50 onward and reaches 100% at epoch 500. This trajectory indicates no persistent codebook collapse in the reported *YelpChi* run, although the behavior may vary across datasets, initializations, and hyperparameter settings.

D.6 Hadamard Interaction Feature

The edge encoder uses $[\mathbf{x}_u \parallel \mathbf{x}_v \parallel \mathbf{x}_u \odot \mathbf{x}_v]$, where concatenation retains endpoint-specific information and the Hadamard product explicitly represents feature-wise interactions between the two endpoints. Although an MLP applied to concatenated features may learn such interactions implicitly, the explicit product provides a direct multiplicative feature. To isolate its contribution, Table 8 compares otherwise matched variants with and without $\mathbf{x}_u \odot \mathbf{x}_v$. The Hadamard term improves AUPRC by 1.42 points on *YelpChi*, 0.89 on *Tolokers*, and 0.56 on *Questions*, while its effect on *Elliptic* is negligible. Hence, it is useful on several datasets but is not uniformly necessary for performance.

Table 8: AUPRC with and without the Hadamard interaction term in the edge encoder.

Dataset	With	Without	Difference
YelpChi	93.61	92.19	+1.42
Elliptic	90.94	90.91	+0.03
Tolokers	50.06	49.17	+0.89
Questions	18.04	17.48	+0.56

D.7 EMA Decay Sensitivity

The EMA decay coefficient γ controls the responsiveness of codebook updates to current assignments, with smaller values enabling faster adaptation and larger values providing stronger temporal smoothing. We evaluate $\gamma \in 0.95, 0.99, 0.999$ while keeping all other settings fixed. As shown in Table 9, the maximum AUPRC variation across the tested values is only 0.73 percentage points over all five datasets. The default $\gamma = 0.99$ achieves the best or comparable performance on most datasets, while no single value consistently performs best across all datasets. Overall, Role-GAD exhibits limited sensitivity to the EMA decay coefficient within the evaluated range.

Table 9: AUPRC sensitivity to the EMA decay coefficient γ.

Dataset	$\gamma = 0.95$	$\gamma = 0.99$	$\gamma = 0.999$
YelpChi	93.08	93.61	93.26
Elliptic	90.94	90.76	90.75
Tolokers	49.33	50.06	50.04
Questions	17.73	17.95	18.04
Reddit	10.97	10.56	10.39

D.8 Loss-Weight Sensitivity

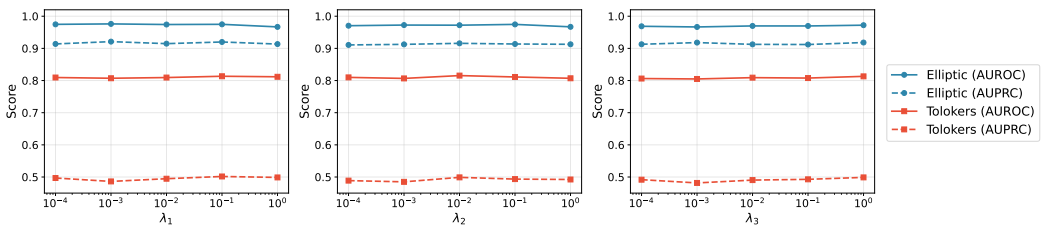


Figure 8: Sensitivity of Role-GAD to the loss-balancing coefficients λ_1, λ_2, and λ_3 on *Elliptic* and *Tolokers*. Each coefficient is varied from 10^{-4} to 10^{0} while the other hyperparameters are held fixed.

Role-GAD optimizes a multi-component objective consisting of node reconstruction, edge reconstruction, and vector-quantization commitment losses. To examine the sensitivity of this objective, we vary the three balancing coefficients λ_1, λ_2, and λ_3, which weight the node-reconstruction loss, edge-reconstruction loss, and VQ-commitment loss, respectively. Each coefficient is varied independently from 10^{-4} to 10^{0} while all other hyperparameters are held fixed. Figure 8 reports both AUROC and AUPRC on *Elliptic* and *Tolokers*.

Across the tested ranges, performance remains relatively stable for all three coefficients, with no abrupt degradation under the evaluated one-factor-at-a-time perturbations. The limited variation with respect to λ_3 is consistent with stable discrete-role learning under the EMA-based codebook update used by Role-GAD. Overall, these results indicate that Role-GAD is not highly sensitive to the precise loss-weight choices on the two evaluated datasets and within the tested ranges. Because the analysis covers two datasets and varies one coefficient at a time, it should be interpreted as a local sensitivity check rather than evidence that hyperparameter tuning is unnecessary in every setting.

D.9 Codebook Size

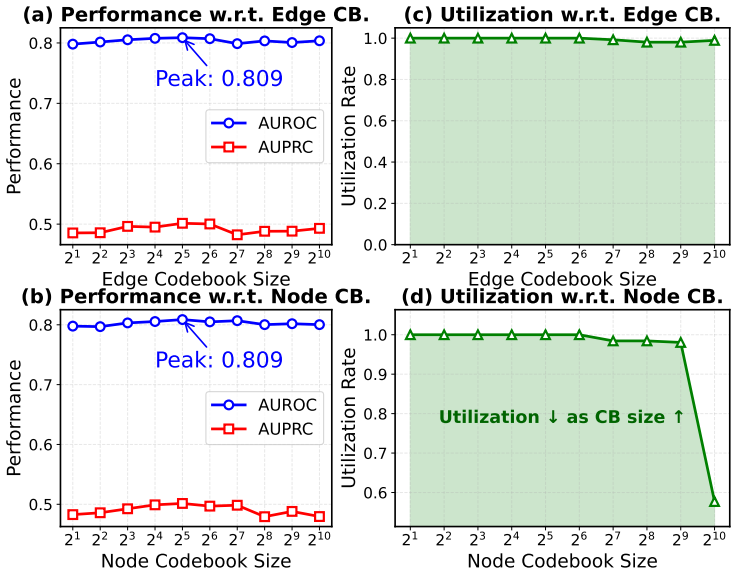


Figure 9: Impact of codebook size on performance and utilization on *Tolokers*. "CB." denotes codebook. When varying one codebook from 2^1 to 2^{10}, the other is fixed at 32.

To investigate how codebook capacity affects performance and utilization, we vary one codebook size from 2^1 to 2^{10} while fixing the other at 32 on *Tolokers*. Figure 9 reports AUROC, AUPRC, and the fraction of active entries for both codebooks. Larger codebooks do not monotonically improve performance, and the best observed performance occurs while utilization remains high. The node-codebook utilization decreases at the largest tested capacity, whereas the edge-codebook utilization remains comparatively high. A plausible explanation is that *Tolokers* contains approximately 44 stored edges per node, providing many more edge assignments than node assignments and therefore sustaining the use of a larger interaction codebook. Overall, the tested configuration is reasonably stable over a range of capacities on *Tolokers*; this single-dataset analysis does not imply that a common capacity is optimal for every dataset.

E Details of Datasets

We adopt five widely used graph anomaly detection benchmarks covering financial transactions, crowdsourcing platforms, and social media scenarios. Each dataset contains over 100 anomalous nodes with anomaly ratios below 25% and an average anomaly ratio of 10%, reflecting the intrinsic class imbalance in GAD. Detailed dataset statistics are provided in Table 10. Specifically:

Table 10: Descriptive statistics of the datasets.

Dataset	#Nodes	#Edges	Feat. Dim	Anomaly Ratio
YelpChi	45,954	3,846,979	32	14.5%
Tolokers	11,758	519,000	10	21.8%
Questions	48,921	153,540	301	3.0%
Reddit	10,984	168,016	64	3.3%
Elliptic	203,769	234,355	166	9.8%

- **YelpChi** constructs a user-review interaction graph from Yelp, where nodes are described by hand-crafted behavioral and statistical features. Anomalous nodes mainly correspond to spam or fraudulent reviewers.
- **Tolokers** is a crowdsourcing interaction graph, in which nodes represent workers connected through shared tasks. Features summarize user profiles and behavioral performance statistics, with anomalies reflecting abnormal or low-quality working patterns.
- **Questions** is derived from a social question-answering platform, where node features are semantic embeddings extracted from user-generated textual descriptions, and anomalies indicate irregular or suspicious user behaviors.
- **Reddit** represents user interaction graphs in online communities, with hand-crafted activity-based features capturing posting and engagement patterns. Anomalous users typically exhibit coordinated or abnormal behaviors.
- **Elliptic** models a Bitcoin transaction network, where nodes represent transactions and edges denote fund flows. Node features capture temporal dynamics and transaction attributes, and anomalies correspond to illicit or fraudulent activities.

F Details of Baselines

We provide detailed descriptions of all baseline models used in our experiments. The baselines are organized into four categories: (1) standard graph neural networks, (2) general graph autoencoder models, (3) autoencoder-based graph anomaly detection methods, and (4) dedicated GNN-based anomaly detection approaches.

1. **Standard Graph Neural Networks:**

- GCN (Kipf, 2016): This method learns node representations through spectral-based graph convolution by aggregating normalized neighborhood features.
- GIN (Xu et al., 2018): This model enhances expressive power by applying sum-based neighborhood aggregation followed by multilayer perceptrons.
- GraphSAGE (Hamilton et al., 2017): This approach constructs inductive node embeddings via neighborhood sampling and aggregation mechanisms.
- GAT (Veličković et al., 2017): This framework incorporates attention mechanisms to adaptively weight neighboring nodes during message passing.

2. **General Graph Autoencoder Models:**

- GAE (Kipf & Welling, 2016): This method encodes nodes using graph convolutional layers and reconstructs graph structure through inner-product decoding.
- VGAE (Kipf & Welling, 2016): Built upon GAE, this model introduces variational latent variables to capture uncertainty in node representations.
- VQVAE(GCN) (Van Den Oord et al., 2017): This model adopts GCNs as the encoder and maps continuous latent variables to a discrete codebook via vector quantization.

3. **Autoencoder-Based Graph Anomaly Detection Methods:**

- DOMINANT (Ding et al., 2019): This framework jointly reconstructs node attributes and graph topology using dual autoencoder architectures.

- DONE and ADONE (Bandyopadhyay et al., 2020): These methods model structural proximity and attribute similarity, with ADONE further incorporating adversarial learning to enhance anomaly separability.
- AnomalyDAE (Fan et al., 2020): This approach performs anomaly-aware joint reconstruction of node features and graph structure.
- GUIDE (Yuan et al., 2021): This framework employs dual autoencoders to reconstruct node attributes and high-order structural patterns, with an attention mechanism to capture neighborhood structural differences for anomaly detection.
- GAD-NR (Roy et al., 2024): This technique identifies anomalies based on reconstruction errors over local neighborhood structures.

4. **Dedicated GNN-Based Anomaly Detection Methods:**

- GAS (Li et al., 2024): A scalable GNN-based framework for spam review detection that extends GCN to heterogeneous and heterophilic graphs, incorporating KNN-based structure adaptation.
- PCGNN (Liu et al., 2021): A fraud detection framework designed for imbalanced graphs, featuring a label-balanced sampler and a learnable neighbor selection mechanism to construct informative subgraphs.
- BWGNN (Tang et al., 2022): This method addresses high-frequency anomaly patterns by employing Beta kernel-based spectral filters to capture localized structural irregularities.
- GHRN (Gao et al., 2023b): A spectral-domain approach that mitigates heterophily by pruning inter-class edges to emphasize high-frequency anomaly-related components.
- XGBGraph (Tang et al., 2023b): A gradient boosting framework that integrates XGBoost with graph structural features for anomaly detection.
- CONSISGAD (Chen et al., 2024): A consistency-based training approach that leverages learnable graph data augmentation under limited supervision.
- GODM (Liu et al., 2023): A data-centric method that utilizes diffusion models to generate latent positive samples, alleviating label imbalance in graph anomaly detection.
- CGenGA (Ma et al., 2024): A latent diffusion-based data augmentation framework that synthesizes graph samples to enhance supervised anomaly detection with scarce labels.
- CGADM (Wei et al., 2026): This method introduces a prior-guided diffusion mechanism that injects a pre-trained conditional anomaly estimator into both forward and reverse diffusion chains to better preserve high-frequency anomaly signals and mitigate over-smoothing.

G Implementation and Evaluation Details

This section presents the implementation details of Role-GAD, followed by the experimental protocols used for baseline methods and fair comparison. Role-GAD is implemented in PyTorch and optimized using the AdamW optimizer. Role-GAD employs three-layer MLPs as both node and edge encoders and decoders to project input features and constructed edge features into the latent space and reconstruct them accordingly. The node and edge codebooks are initialized using K-means clustering over the corresponding latent embeddings, which provides meaningful role prototypes at the early stage of training and improves convergence stability. Although Role-GAD involves multiple hyperparameters, most of them are fixed across all experiments. The codebook update introduces three hyperparameters that are fixed across all experiments. Specifically, we set the decay coefficient γ to 0.99, the Laplace smoothing constant ϵ to 10^{-5} for numerical stability, and the dead-prototype threshold to 2 consecutive iterations. These parameters are used solely for stabilizing codebook updates and are not tuned per dataset.

For several baselines, we directly adopt the results reported in CGADM (Wei et al., 2026), while the remaining methods are implemented and evaluated under the same experimental settings. Moreover, for originally unsupervised baselines, we retain their original training objectives and introduce an additional cross-entropy classification term so that they use the same label supervision as Role-GAD. We re-tune these supervised adaptations rather than directly reusing the hyperparameters of their original unsupervised settings. This protocol is intended to provide a consistent and reasonable supervised adaptation of each method.

H Complexity Analysis

Let $|\mathcal{V}|$ denote the number of nodes, $|\mathcal{E}|$ the number of edges, D the input feature dimension, d the latent dimension, H the hidden dimension, K_v and K_e the node and edge codebook sizes, and L the number of GNN layers. Both the node and edge encoders/decoders are implemented as MLPs, resulting in a computational complexity of $\mathcal{O}(|\mathcal{V}|DH + |\mathcal{E}|DH)$. Vector quantization requires $\mathcal{O}(|\mathcal{V}|K_v d + |\mathcal{E}|K_e d)$. Each GNN layer has complexity $\mathcal{O}(|\mathcal{V}|H^2 + |\mathcal{E}|H)$. With L layers, the total GNN computation cost is $\mathcal{O}(L(|\mathcal{V}|H^2 + |\mathcal{E}|H))$. Combining all components, the overall time complexity is $\mathcal{O}\big(|\mathcal{V}|DH + |\mathcal{E}|DH + |\mathcal{V}|K_v d + |\mathcal{E}|K_e d + L(|\mathcal{V}|H^2 + |\mathcal{E}|H)\big)$. For sparse graphs where $|\mathcal{E}| \gg |\mathcal{V}|$, the edge-related terms dominate, and the complexity can be written as $\mathcal{O}\big(|\mathcal{E}|(DH + K_e d + LH)\big)$. Therefore, for fixed codebook sizes, the asymptotic expression is linear in the number of edges. The measured time-to-checkpoint comparison in Section D.1 evaluates convergence under the reported setup; it is not a controlled empirical estimate of asymptotic edge scaling.