Graph Evidential Learning for Anomaly Detection

Chunyu Wei
Renmin University of China
Beijing, China
weicy15@icloud.com

Wenji Hu
Renmin University of China
Beijing, China
2024000991@ruc.edu.cn

Xingjia Hao
Guangxi University
Nanning, China
haoxingjia@st.gxu.edu.cn

Yunhai Wang*
Renmin University of China
Beijing, China
cloudseawang@gmail.com

Yueguo Chen
Renmin University of China
Beijing, China
chenyueguo@ruc.edu.cn

Bing Bai
Microsoft MAI
Beijing, China
bingbai@microsoft.com

Fei Wang
Cornell University
New York, United States
few2001@med.cornell.edu

Abstract

Graph anomaly detection faces significant challenges due to the scarcity of reliable anomaly-labeled datasets, driving the development of unsupervised methods. Graph autoencoders (GAEs) have emerged as a dominant approach by reconstructing graph structures and node features while deriving anomaly scores from reconstruction errors. However, relying solely on reconstruction error for anomaly detection has limitations, as it increases the sensitivity to noise and overfitting. To address these issues, we propose Graph Evidential Learning (GEL), a probabilistic framework that redefines the reconstruction process through evidential learning. By modeling node features and graph topology using evidential distributions, GEL quantifies two types of uncertainty: graph uncertainty and reconstruction uncertainty, incorporating them into the anomaly scoring mechanism. Extensive experiments demonstrate that GEL achieves state-of-the-art performance while maintaining high robustness against noise and structural perturbations.

CCS Concepts

• **Information systems** → **Data mining**; • **Computing methodologies** → **Anomaly detection**; *Neural networks.*

Keywords

Graph Neural Network, Anomaly Detection, Evidential Learning

ACM Reference Format:

Chunyu Wei, Wenji Hu, Xingjia Hao, Yunhai Wang, Yueguo Chen, Bing Bai, and Fei Wang. 2025. Graph Evidential Learning for Anomaly Detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736989

*Corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736989

1 Introduction

Anomaly detection focuses on identifying objects that significantly deviate from the majority within a dataset [10]. With the rapid proliferation of relational data driven by the Internet, graph-structured data has become a natural representation for modeling complex, interconnected systems. This has led to growing interest in Graph Anomaly Detection (GAD), which aims to detect anomalous nodes within large-scale graphs. GAD has diverse applications in fields such as fraud detection [66], network intrusion prevention [16], and the identification of abnormal behaviors in social networks [58], biological systems [24], and financial transactions [12].

Despite its practical importance, labeled data in graph settings is often scarce compared to the vast scale of interaction data, making supervised approaches infeasible in most scenarios. Manual labeling of anomalies is not only expensive and time-consuming but also impractical due to the high diversity and rarity of anomalous behaviors. These have driven the adoption of unsupervised methods for GAD, enabling adaptive and scalable solutions that can generalize across various anomaly types and datasets.

Existing unsupervised methods for GAD predominantly rely on **reconstruction-based approaches**, where autoencoders are commonly employed to learn low-dimensional representations of graph data. The central hypothesis is that autoencoders capture the core structure of normal nodes, while anomalies, due to their sparsity and distinctiveness, are poorly reconstructed, resulting in higher reconstruction errors. **GAEs**, which incorporate Graph Neural Networks (GNNs) to encode both topological structures and node attributes, have shown promise in detecting anomalies within large, complex graphs [42].

However, existing anomaly detection methods based on reconstruction error exhibit several limitations, as illustrated in Figure 1. On one hand, most reconstruction-based methods assume that normal nodes can be accurately reconstructed, while anomalous nodes cannot. However, in real-world scenarios, these methods are prone to overfitting, allowing anomalous nodes to be reconstructed with

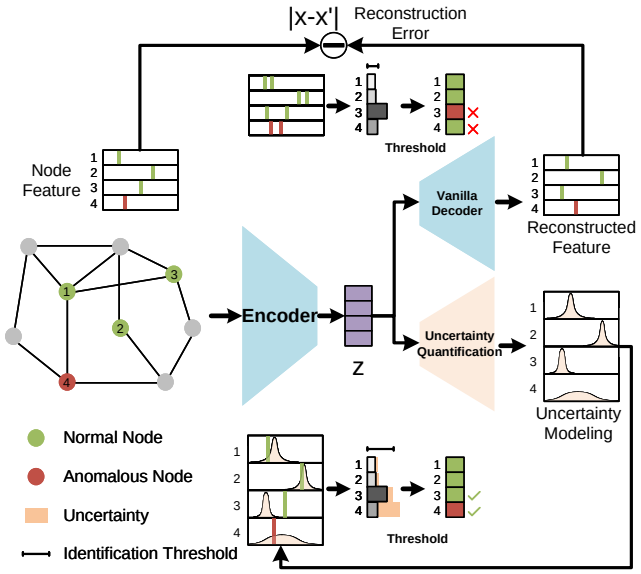


Figure 1: Introducing Uncertainty for anomaly detection. Relying on reconstruction error, nodes 3 and 4 in the graph are misclassified without introducing uncertainty.

high accuracy, thereby evading detection. On the other hand, these methods typically rely on predefined reconstruction error threshold to distinguish between normal and anomalous nodes. In practical applications, this approach is highly sensitive to inherent noise in graph data and reconstruction models, such as inconsistencies in node features or errors in structural encoding. As a result, high-noise normal nodes may fail to be accurately reconstructed, leading to their misclassification as anomalies.

To overcome these challenges, we propose shifting the paradigm of graph anomaly detection from relying on reconstruction error to explicitly modeling uncertainty. This approach aims to address the shortcomings of sensitivity to noise and overfitting, enabling the model to express high uncertainty when encountering anomalies. However, integrating uncertainty into graph reconstruction introduces two fundamental challenges:

- **Uncertainty Diversity.** There exist diverse sources of uncertainty in reconstruction: (1) Graph uncertainty, which arises when anomalies disguise themselves by interacting frequently with normal nodes or mimicking their features, creating inconsistencies in local topological information; (2) Reconstruction uncertainty, which occurs when the reconstruction model, trained predominantly on normal data, encounters anomalies that lie outside the training distribution. Effective anomaly detection requires a unified approach to capture both sources of uncertainty.
- **Modality Heterogeneity.** Graph data inherently combines **continuous node features** and **discrete topological structures**, each following distinct distributional characteristics. Node features are often modeled as continuous distributions in high-dimensional spaces, while topological structures are discrete and binary, representing edge presence or absence. Estimating uncertainty for these heterogeneous modalities requires differentiated modeling strategies while simultaneously capturing their interdependencies within the unified graph structure.

To address the aforementioned challenges simultaneously, we propose Graph Evidential Learning (GEL), a novel framework that integrates uncertainty modeling for graph anomaly detection.

To address **Uncertainty Diversity**, we adopt a higher-order evidential distribution to parameterize the reconstruction process. Instead of separately modeling different sources of uncertainty (e.g., using Monte Carlo Dropout or training multiple models to estimate prediction variance [1]), we directly learn a unified evidential distribution for graph reconstruction. This approach eliminates the need for task-specific uncertainty estimation techniques, enabling a more efficient and generalizable anomaly detection framework. The higher-order evidential distribution generates a set of lower-order reconstruction likelihood functions, thus encapsulating both graph and reconstruction uncertainty simultaneously. By leveraging Bayesian inference, we parameterize this evidential distribution without requiring extensive sampling, as typically required by Bayesian Neural Networks.

To address **Modality Heterogeneity**, we introduce two evidential distributions to model the uncertainties in continuous node features and discrete topological structures, respectively. Specifically, we employ a Normal Inverse-Gamma (NIG) distribution to represent the reconstruction distribution of continuous node features and a Beta distribution to model the reconstruction distribution of discrete topological structures (i.e., the presence or absence of edges between nodes). These two higher-order evidential distributions jointly govern the reconstruction of the graph, enabling us to capture the interdependencies between features and structures in a principled manner. Neural networks are used to parameterize these evidential distributions, allowing efficient sampling of the reconstruction likelihoods for both modalities.

Our key contributions can be summarized as follows:

- We propose a novel paradigm for GAD by shifting from reconstruction error to uncertainty modeling, enhancing the robustness of anomaly detection in noisy graph data. To comprehensively capture the various sources of uncertainty, we model uncertainty using a higher-order evidential distribution.
- We handle the diverse modality of node features and topological structures through a joint evidential distribution.
- Extensive experiments on five datasets demonstrate that GEL achieves state-of-the-art performance in GAD. Ablation studies show that GEL is notably more robust in unsupervised settings.

2 Related Works

2.1 Graph Anomaly Detection

Anomaly detection has been applied in fields like financial fraud detection [26] and network security [60]. Traditional methods, including statistical techniques and distance-based approaches like k-nearest neighbors (k-NN)[41], isolation forests[20], and PCA[29], detect anomalies based on statistical deviations but struggle with high-dimensional or non-Euclidean data [51]. Deep learning approaches, such as autoencoders [46], VAEs[40], and RNNs[49], capture patterns in high-dimensional data but overlook the relational structure in graphs, limiting their anomaly detection capability.

GNNs [35] excel at utilizing graph structure for anomaly detection [31]. For example, Mul-GAD [44] integrates node attributes and structure, while Kumagai et al. [36] address class imbalance

using semi-supervised GCNs. AddGraph [70] leverages temporal GCNs for dynamic GAD. However, these methods often require substantial labeled data, which is scarce [45]. Unsupervised approaches, such as structure-based methods [61], distance-based methods like Node2Vec [25], and subgraph-based techniques [65], detect anomalies by analyzing graph topology and local subgraphs. Spectral and density-based methods, like LOF [9], detect anomalies based on graph irregularities [27]. Deep learning-based methods, including GAEs [33], reconstruct graph structures to detect anomalies based on reconstruction errors [21]. Variants like VGAEs [33] and GraphGANs [71] enhance robustness. Methods such as DOMINANT [18] and GAD-NR [55] extend GAEs by reconstructing node neighborhoods.

However, traditional GAEs face two main issues: reliance on reconstruction error, making them sensitive to noise, and inability to handle overfitting samples, leading to misclassification. We address these challenges by introducing uncertainty as an anomaly criterion to improve robustness and reliability.

2.2 Uncertainty Quantification

Uncertainty Quantification (UQ) has become essential for improving the reliability and interpretability of neural network predictions [11], especially in safety-critical areas like autonomous driving [47], healthcare [50], and finance [7]. Bayesian Neural Networks (BNNs) were an early approach to UQ by assigning prior distributions to model parameters and enabling posterior inference [8, 28, 63]. However, their practical use is limited by high-dimensional inference intractability, computational costs of methods like variational inference, and challenges in choosing priors with limited domain knowledge [22].

Ensemble methods, including MC Dropout [23] and Deep Ensembles [38], offer an alternative by leveraging diverse predictions from multiple models. While effective, they incur high computational overhead and face challenges in maintaining ensemble diversity, particularly in resource-limited settings.

Evidential Learning provides a more efficient UQ alternative by directly modeling uncertainty via higher-order distributions over predictions [59]. In regression, it uses a Normal-Inverse-Gamma distribution to model Gaussian parameters [3], and in classification, it employs the Dirichlet distribution for class probabilities [59]. Unlike BNNs and ensembles, evidential learning enables efficient uncertainty estimation in a single forward pass, making it suitable for real-time applications. It has been applied successfully in image classification [59], regression [3], multi-view learning [43], and OOD detection [14].

Building on this, we introduce evidential learning to unsupervised graph representation learning by modeling graph topology and node features as a joint higher-order evidential distribution, enhancing uncertainty assessment in graph reconstruction.

3 Preliminaries

3.1 Problem Definition

Attributed Graph. An *attributed graph* consists of nodes, their attributes, and the relations (edges) among them [55]. Formally, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of N nodes, $\mathcal{E} = \{e_{ij} \mid v_i \text{ and } v_j \text{ are connected}\}$ is the edge set, and $\mathbf{X} \in \mathbb{R}^{N\times d}$ is the feature matrix, where each row $\mathbf{x}_i \in \mathbb{R}^d$ represents the d-dimensional attributes of node v_i. The edge set $\mathcal{E}$ can be represented by an *adjacency matrix* $\mathbf{A} \in \mathbb{N}^{N\times N}$, where $\mathbf{A}_{ij} = 1$, if $e_{ij} \in \mathcal{E}$, and 0 otherwise. The *degree matrix* $\mathbf{D} \in \mathbb{N}^{N\times N}$ is a diagonal matrix where $\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{A}_{ij}$, and $\mathbf{D}_{ij} = 0$, $\forall i \neq j$. Here, $\mathbf{D}_{ii}$ denotes the degree of node v_i, with off-diagonal entries being zero.

Unsupervised Graph Anomaly Detection. Given an *attributed graph* $\mathcal{G}$, the node set $\mathcal{V}$ is divided into two disjoint subsets: the anomalous node set $\mathcal{V}_a$ and the normal node set $\mathcal{V}_n$, such that $\mathcal{V}_a \cap \mathcal{V}_n = \emptyset$ and $\mathcal{V}_a \cup \mathcal{V}_n = \mathcal{V}$. The goal of unsupervised graph anomaly detection is to assign anomaly labels for all nodes by estimating the probability $p(v_i \in \mathcal{V}_a \mid \mathcal{G}, v_i \in \mathcal{V})$, which represents the likelihood that node v_i is anomalous, given the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$.

3.2 Reconstruction with GAE

An autoencoder (AE) compresses high-dimensional data into a latent representation via an encoder and reconstructs it through a decoder. AEs capture normal data properties, with large reconstruction errors indicating anomalies. This extends to GAEs, where the encoder uses a GNN to incorporate node features and topology. Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$, the encoder is:

$$\mathbf{Z} = \text{Enc}(\mathcal{G}) = \text{GNN}(\mathbf{X}, \mathbf{A}), \tag{1}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times d'}$ is the latent representation, with each $\mathbf{Z}_i \in \mathbb{R}^{d'}$ corresponding to node $v_i \in \mathcal{V}$, and $d' \ll d$. To capture higher-order neighbor information, GNNs stack multiple graph convolution layers. At layer l, node embeddings $\mathbf{H}^{(l)}$ are updated by aggregating neighbor information:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right), \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix with self-loops, and $\mathbf{W}^{(l)}$ is the weight matrix. The output $\mathbf{H}^{(L)}$ is the compressed representation $\mathbf{Z}$.

Feature Reconstruction. The decoder reconstructs node features by taking $\mathbf{Z}$ as input and outputting $\hat{\mathbf{X}} \in \mathbb{R}^{N\times d}$. The decoder Dec_f is a multi-layer perceptron (MLP):

$$\hat{\mathbf{X}} = \text{Dec}_f(\mathbf{Z}) = \text{MLP}(\mathbf{Z}). \tag{3}$$

Topology Reconstruction. To reconstruct the graph topology, the decoder predicts the adjacency matrix $\mathbf{A}$ by estimating edge presence. For nodes i and j, the decoder Dec_t computes the probability of an edge as:

$$\hat{\mathbf{A}} = \text{Dec}_t(\mathbf{Z}) = \sigma\left(\mathbf{Z}^\top\mathbf{Z}\right), \tag{4}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function.

We adopt Mean Squared Error (MSE) for optimizing both feature and topology reconstruction.

4 Methodology

Figure 2 illustrates the basic framework of GEL. In Section 3.2, we introduced the decoders Dec_f and Dec_t designed to reconstruct node features and graph topology from the latent embedding $\mathbf{Z}$. However, these neural network-based decoders lack a measure of confidence in the reconstruction process, which can lead to unreliable outputs. To address this limitation, we propose an evidential framework, which decouples the graph reconstruction process into two stages: (1) generating evidence for graph components, and (2) deriving reconstruction results based on evidence.

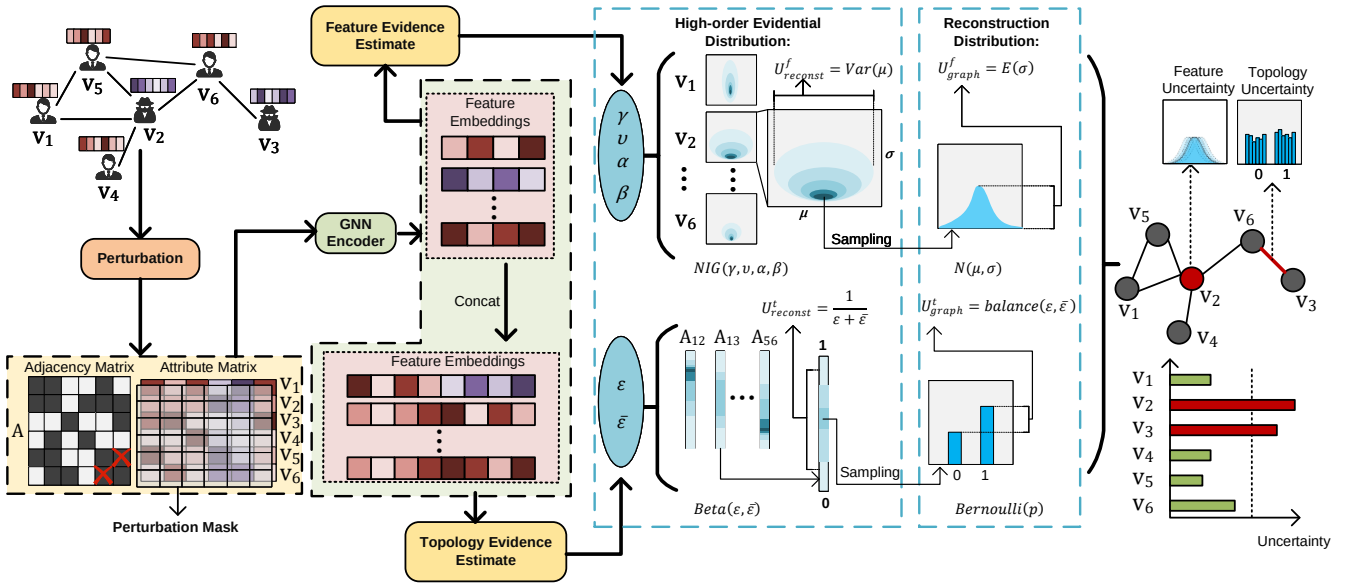


Figure 2: The overview of the GEL framework. GEL models high-order evidential distribution to calculate feature uncertainty and topology uncertainty during the graph reconstruction process, leveraging these uncertainties for anomaly detection.

4.1 Learning the Evidence and Uncertainty

4.1.1 Feature Evidence. In traditional graph autoencoders, decoders directly output a point estimate for each node feature, but such estimates fail to capture the inherent uncertainty of predictions. We propose modeling node feature estimates as continuous distributions to reflect this uncertainty. Specifically, we assume a Gaussian prior $\mathcal{N}_0(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ for the node feature $\mathbf{X}$.

To model the uncertainty of the Gaussian parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$, we adopt a second-order probabilistic framework. We place a Gaussian prior on the mean $\boldsymbol{\mu}$, and an Inverse-Gamma prior on the variance $\boldsymbol{\sigma}$, as follows:

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\gamma}, \boldsymbol{\sigma}^2 \boldsymbol{\nu}^{-1}), \quad \boldsymbol{\sigma}^2 \sim \Gamma^{-1}(\boldsymbol{\alpha}, \boldsymbol{\beta}),$$

where $\mathcal{N}(\cdot)$ denotes a Gaussian distribution, and $\Gamma^{-1}(\cdot)$ refers to an Inverse-Gamma distribution. The hyperparameters $\boldsymbol{\gamma}$, $\boldsymbol{\nu}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ are all of the same size as the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, and encode the evidence supporting the Gaussian distribution $\mathcal{N}_0$. Specifically, we have $\boldsymbol{\gamma} \in \mathbb{R}$, $\boldsymbol{\nu} > 0$, $\boldsymbol{\alpha} > 1$, and $\boldsymbol{\beta} > 0$. For clarity, we denote the hyperparameters corresponding to a specific node's feature vector $\mathbf{X}_{ih}$ in $\mathbf{X}$ by using their non-bold versions: μ, σ, α, β, γ, and ν.

We aim to estimate the posterior distribution $q(\mu, \sigma^2) = p(\mu, \sigma^2 | \mathbf{X}_{ih})$. To approximate the true posterior [52], we assume that the distribution can be factorized as $q(\mu, \sigma^2) = q(\mu)q(\sigma^2)$, which allows us to efficiently model it using the Normal-Inverse-Gamma (NIG) distribution, a conjugate prior for the Gaussian. The joint density function of the NIG distribution is given by:

$$p(\mu, \sigma^2 \mid \gamma, \nu, \alpha, \beta) = \frac{\beta^\alpha \sqrt{\nu}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left\{-\frac{2\beta + \nu(\gamma - \mu)^2}{2\sigma^2}\right\}, \tag{5}$$

where $\Gamma(\cdot)$ represents the Gamma function.

The parameters of this conjugate prior can be interpreted as "virtual observations," hypothetical data points used to estimate specific properties [17].

Remark 4.1. *In the NIG distribution, μ is estimated from ν virtual samples with a mean of γ, while σ^2 is inferred from α virtual samples with a sum of squared deviations proportional to ν. Larger ν and α indicate stronger evidential support, reducing uncertainty in the prior.*

This interpretation allows us to quantify the confidence in the reconstructed feature distribution by linking the hyperparameters γ, ν, α, and β to virtual observations. Using the NIG distribution, we define two types of uncertainty:

- **Reconstruction Uncertainty:** This measures the variance of μ in $\mathcal{N}_0$, reflecting the model's uncertainty in reconstruction, defined as $\mathcal{U}^f_{\text{reconst}} = \frac{\beta}{\nu(\alpha-1)}$.
- **Graph Uncertainty:** This represents the expected value of σ^2 in $\mathcal{N}_0$, expressed as $\mathcal{U}^f_{\text{graph}} = \frac{\beta}{\alpha-1}$, capturing inherent graph randomness.

These uncertainties reflect both the data's inherent randomness and the model's ability to reliably reconstruct it.

4.1.2 Topological Evidence. For topology reconstruction, traditional decoders use a sigmoid function to predict the presence of an edge e_{ij} between nodes v_i and v_j, providing a point estimate in the range $[0, 1]$. However, this approach fails to quantify prediction uncertainty. To address this, we model the existence of edge e_{ij} using a Bernoulli distribution $\mathcal{B}(p_{ij})$, where p_{ij} is the edge's probability, and $\boldsymbol{p}$ represents the probability of all edges in $\mathcal{E}$ existing.

We adopt Subjective Logic (SL) [30] to quantify uncertainty in the Bernoulli distribution. SL extends Dempster-Shafer Theory (DST) [15] by using a Dirichlet distribution to formalize belief assignments, enabling a rigorous application of evidential theory to quantify evidence and uncertainty. Following CEDL [2], we leverage SL to define a probabilistic framework for topological reconstruction based on topological evidence.

Proposition 4.1. *A Beta distribution parameterized by evidence variables can represent the density of probability assignments for discrete outcomes, effectively modeling both second-order probabilities and associated uncertainties.*

To disentangle reconstruction results from uncertainty, we introduce two positive evidence variables, $\mathbf{E}_{ij}$ and $\bar{\mathbf{E}}_{ij}$, for each potential edge (i, j), where $\mathbf{E}_{ij}$ represents the evidence supporting the edge's existence and $\bar{\mathbf{E}}_{ij}$ represents the evidence against it, with both being positive. We propose to model the probabilities p_{ij} as a Beta distribution:

$$p(p_{ij} \mid \varepsilon_{ij}, \bar{\varepsilon}_{ij}) = \frac{\Gamma(\varepsilon_{ij} + \bar{\varepsilon}_{ij})}{\Gamma(\varepsilon_{ij})\Gamma(\bar{\varepsilon}_{ij})} p_{ij}^{\varepsilon_{ij}-1}(1 - p_{ij})^{\bar{\varepsilon}_{ij}-1}, \tag{6}$$

where $\varepsilon_{ij} = \mathbf{E}_{ij} + 1$ and $\bar{\varepsilon}_{ij} = \bar{\mathbf{E}}_{ij} + 1$ are the Beta parameters.

The evidence strength for edge (i, j), called Beta strength, is given by:

$$S_{ij} = \varepsilon_{ij} + \bar{\varepsilon}_{ij}, \tag{7}$$

indicating that a larger S_{ij} reflects more information about the edge, whether supporting or opposing its presence. The predicted edge probability is $\hat{p}_{ij} = \frac{\varepsilon_{ij}}{S_{ij}}$.

To quantify the uncertainty in topological reconstruction, we define two complementary measures:

- **Reconstruction Uncertainty:** This measures the uncertainty from the reconstruction process, reflecting the model's confidence in predicting edge presence or absence. It is inversely proportional to the Beta strength $\mathcal{U}^t_{\text{reconst}} = \frac{1}{S_{ij}}$.
- **Graph Uncertainty:** This captures inherent uncertainty in the graph structure, arising from conflicting or imbalanced evidence about edge existence. It is defined as:

$$\mathcal{U}^t_{\text{graph}} = (b_{ij} + \bar{b}_{ij}) \left(1 - \frac{|b_{ij} - \bar{b}_{ij}|}{b_{ij} + \bar{b}_{ij}}\right), \tag{8}$$

  where $b_{ij} = \frac{\mathbf{E}_{ij}}{S_{ij}}$ and $\bar{b}_{ij} = \frac{\bar{\mathbf{E}}_{ij}}{S_{ij}}$ represent the belief masses supporting and opposing edge e_{ij}, respectively. This measure reflects the balance between the evidence for and against the edge's existence. Higher graph uncertainty indicates balanced evidence, while lower uncertainty occurs with a clear belief in the edge's presence or absence.

By incorporating evidence variables and leveraging the Beta distribution, this framework allows us to disentangle predictions and uncertainties, making the reconstruction process more robust to graph variability and reconstruction ambiguity.

4.2 From the Evidence to Reconstruction

We now integrate feature and topology reconstruction into a unified evidential framework utilizing the evidence learned from the latent embeddings $\mathbf{Z}$. Given the embeddings obtained from the GAE encoder in Section 3.2, we model the joint distribution of the reconstructed graph as:

$$p(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{p} \mid \mathbf{Z}) = \prod_{i=1}^{N}\prod_{h=1}^{D} p(\mu_{ih}, \sigma^2_{ih} \mid \mathbf{Z}_i) \prod_{i=1}^{N}\prod_{j=1}^{N} p(p_{ij} \mid \mathbf{Z}_i, \mathbf{Z}_j), \tag{9}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ represent the mean and variance of node features, and $\mathbf{p}$ denotes the edge existence probabilities. $p(\mu_{ih}, \sigma^2_{ih} \mid \mathbf{Z}_i)$ models the distribution for the h-th feature of node v_i, and $p(p_{ij} \mid \mathbf{Z}_i, \mathbf{Z}_j)$ models edge existence.

We model uncertainty in feature and topology reconstructions using the NIG and Beta distributions (Equations 5 and 6). To parameterize these, we use two neural networks f_{θ_1} and f_{θ_2} to estimate the distribution parameters directly from $\mathbf{Z}$:

$$[\boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\alpha}, \boldsymbol{\beta}] = f_{\theta_1}(\mathbf{Z}), \quad [\boldsymbol{\varepsilon}, \bar{\boldsymbol{\varepsilon}}] = f_{\theta_2}(\mathbf{Z}).$$

Here, f_{θ_1} outputs the parameters $\boldsymbol{\gamma}$ (mean), $\boldsymbol{\nu}$ (degree of freedom), $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ (scale parameters) for the NIG distribution, and f_{θ_2} outputs the evidence parameters $\boldsymbol{\varepsilon}$ and $\bar{\boldsymbol{\varepsilon}}$ for the Beta distribution. Please refer to Appendix D for detailed implementation.

With the learned evidential parameters, we express higher-order distributions encapsulating both reconstruction and associated uncertainties. For feature reconstruction with the NIG distribution, the mean of μ_{ih} estimates $\mathbf{X}_{ih}$:

$$\textbf{Feature Reconstruction:} \quad \hat{\mathbf{X}}_{ih} = \mathbb{E}[\mu_{ih}] = \gamma_{ih}. \tag{10}$$

For topology reconstruction with the Beta distribution, the mean of $\mathbf{p}_{ij}$ estimates $\mathbf{A}_{ij}$:

$$\textbf{Topology Reconstruction:} \quad \hat{\mathbf{A}}_{ij} = \mathbb{E}[p_{ij}] = \frac{\varepsilon_{ij}}{\varepsilon_{ij} + \bar{\varepsilon}_{ij}}, \tag{11}$$

4.3 Optimization

We define the joint evidential distribution $p(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{p} \mid \mathbf{Z})$ in Equ 9. The optimization process is framed as a multi-task learning problem with two objectives: (1) maximizing model evidence to enhance reconstruction accuracy, and (2) minimizing evidence for incorrect predictions to enforce uncertainty when the model is wrong.

4.3.1 Data Perturbation. In unsupervised graph reconstruction, using a fixed graph for evidential learning can lead to overfitting, particularly in uncertainty modeling, undermining anomaly detection. To improve generalization and ensure reliable uncertainty estimates, we introduce perturbations to node features and graph topology during each training iteration. Specifically, we add Gaussian noise to node features: $\tilde{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{n}_i$, where $\mathbf{n}_i \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. For topology, we apply dropout to the adjacency matrix using a random mask $\mathbf{M}$, where $M_{ij} \sim \text{Bernoulli}(1 - p)$, resulting in the perturbed adjacency matrix $\tilde{\mathbf{A}} = \mathbf{A} \odot \mathbf{M}$. These perturbations expose the model to diverse data variations, mitigating overfitting and improving uncertainty estimation for anomaly detection.

4.3.2 Maximizing Reconstruction Fit. To maximize the likelihood of the observed graph data, we utilize the evidential distributions for feature and topological reconstruction. For feature reconstruction, the negative log-likelihood (NLL) loss is defined as:

$$\begin{aligned}\mathcal{L}_f^{\text{NLL}} = \sum_{i=1}^{N}\sum_{h=1}^{d} \Bigg(& \frac{1}{2}\log\left(\frac{\pi}{\nu_{ih}}\right) - \alpha_{ih}\log(\Omega_{ih}) \\ & + \left(\alpha_{ih} + \frac{1}{2}\right)\log\left((X_{ih} - \gamma_{ih})^2\nu_{ih} + \Omega_{ih}\right) \\ & + \log\left(\frac{\Gamma(\alpha_{ih})}{\Gamma\left(\alpha_{ih} + \frac{1}{2}\right)}\right)\Bigg)\end{aligned} \tag{12}$$

where $\Omega_{ih} = 2\beta_{ih}(1 + \nu_{ih})$. Please refer to Appendix A for detailed derivation. Similarly, for topological reconstruction, the NLL loss is:

$$\mathcal{L}_t^{\text{NLL}} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathbf{A_{ij}}\log(\frac{S_{ij}}{\varepsilon_{ij}}) + (1 - \mathbf{A_{ij}})\log(\frac{S_{ij}}{\bar{\varepsilon}_{ij}})\right), \tag{13}$$

where S_{ij} is the Beta strength as defined in Equ 7.

4.3.3 Minimizing Evidence for Errors. To penalize incorrect reconstructions, we minimize evidence strength in regions with high error, preventing evidence from becoming excessively large, which could lead to NaN or Inf values during training. For topological reconstruction, this is achieved by minimizing the Kullback-Leibler (KL) divergence between the predicted Beta distribution and the non-informative prior Beta(1, 1):

$$\mathcal{L}_t^{\mathrm{R}} = \sum_{i=1}^{N}\sum_{j=1}^{N} |\mathbf{A}_{ij} - \frac{\varepsilon_{ij}}{\varepsilon_{ij}+\bar{\varepsilon}_{ij}}| \cdot \mathrm{KL}[\mathrm{Beta}(\varepsilon_{ij},\bar{\varepsilon}_{ij}) \parallel \mathrm{Beta}(1,1)]. \quad (14)$$

For feature reconstruction, instead of using KL divergence with a zero-evidence prior, we penalize the predicted evidence based on reconstruction error:

$$\mathcal{L}_f^{\mathrm{R}} = \sum_{i=1}^{N}\sum_{h=1}^{d} |X_{ih} - \gamma_{ih}| \cdot (2\nu_{ih} + \alpha_{ih}). \quad (15)$$

This loss discourages large evidence values when prediction errors are high, promoting increased uncertainty in regions with poor predictions.

4.3.4 Multi-task Learning Framework. We combine the two objectives into a unified multi-task learning framework, with the total loss defined as::

$$\mathcal{L} = \lambda_1 \mathcal{L}_f^{\mathrm{NLL}} + \lambda_2 \mathcal{L}_t^{\mathrm{NLL}} + \lambda_3 \mathcal{L}_f^{\mathrm{R}} + \lambda_4 \mathcal{L}_t^{\mathrm{R}}, \quad (16)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are hyperparameters balancing the contribution of each objective. This framework optimizes both feature and topology reconstruction while accounting for uncertainty in both.

4.4 Anomaly Detection

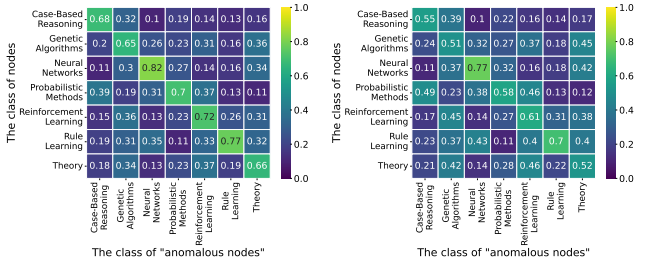
We define the anomaly score of a node v based on uncertainties from the GAE, where a higher score indicates the node is more likely to be anomalous. To improve robustness, we combine uncertainty with reconstruction error. Given modality heterogeneity, GEL introduces parameters λ_{g}, λ_{r}, λ_{f}, and λ_{t} to balance these factors. The anomaly score y_v is:

$$\begin{aligned} y_v =& \lambda_{\mathrm{f}}(\lambda_{\mathrm{g}}\mathcal{U}^f_{\mathrm{graph}} + \lambda_{\mathrm{r}}\mathcal{U}^f_{\mathrm{reconst}}) + \lambda_{\mathrm{t}}(\lambda_{\mathrm{g}}\mathcal{U}^t_{\mathrm{graph}} + \lambda_{\mathrm{r}}\mathcal{U}^t_{\mathrm{reconst}}) \\ &+ |\mathbf{X}_v - \hat{\mathbf{X}}_v| + \sum_{j\in N(v)} |\mathbf{A}_{vj} - \hat{\mathbf{A}}_{vj}|, \end{aligned}$$

where $N(v)$ denotes the neighbors of v.

4.5 Analysis

To validate the effectiveness of uncertainty quantification in graph anomaly detection, we conduct experiments on the Cora dataset to test two hypotheses: (1) uncertainty measures are effective for identifying anomalies, and (2) incorporating uncertainty improves detection performance. We trained the Graph Evidential Learning (GEL) model, excluding nodes from each of the seven classes to create artificial anomalies, and computed normalized anomaly scores based on the model's uncertainty estimates. The model assigned significantly higher uncertainty to anomalous nodes compared to in-distribution nodes (Figure 3a), confirming the first hypothesis. To test the second hypothesis, we compared GEL against a baseline reconstruction error-based model, GCNAE [33]. The results (Figure 3b) show that GEL's uncertainty-based detection outperforms the baseline, which solely relies on reconstruction error. This highlights that traditional models may overfit and lack a mechanism to quantify uncertainty, while GEL's evidential framework improves anomaly detection by reducing overfitting.



(a) GEL: uncertainty estimates. **(b) GCNAE: reconstruction error.**

Figure 3: Heatmaps of average normalized anomaly scores on the Cora dataset. Rows correspond to the class omitted during training (anomalous class), columns represent the class labels during testing.

5 Experiments

5.1 Experimental Setup

Datasets. We conduct experiments on five publicly available and popular real-world graph datasets to evaluate the effectiveness and generalizability of GEL in anomaly point detection tasks. These datasets span a diverse range of domains, including social media (Weibo and Reddit), entertainment (Disney and Books), and corporate communication (Enron), demonstrating the broad applicability of our approach. Please refer to Appendix C for more details.

Baselines and Metrics. Following BOND [42], we select nineteen baseline anomaly detection models for comparison. The baseline models include non-graph-based methods such as LOF [9], IF [4], and MLPAE [56], as well as clustering and matrix factorization-based anomaly detection algorithms like SCAN [61], Radar [39], and ANOMALOUS [53]. Furthermore, we compared GEL with other models employing graph neural networks, such as AnomalyDAE [21], GCNAE [34], DOMINANT [19], GAD-NR [55], G3AD [6], MuSE [32], OCGIN [67], and adversarial learning-based GAAN [13] and contrastive learning-based CONAD [62]. For detailed description, please refer to Appendix B. Consistent with [55, 69], we evaluate anomaly detection performance using the Area Under the ROC Curve (AUC) and Recall@K metrics [1].

5.2 Performance Comparison (RQ1)

We evaluate anomaly detection on five datasets. Best-performing models are highlighted in **bold**, and the best results among baselines are underlined. 'OOM_C' indicates models that ran out of GPU memory. Detailed results are presented in Tables 1 and 2. Key observations include:

- GEL significantly outperforms other methods on four out of five datasets in both AUC and Recall@K. Specifically, GEL achieves an average AUC improvement of 4.64% over the best baseline, demonstrating its effectiveness in capturing representational and

[1]The code is available on https://github.com/wuanjunruc/GEL.

Table 1: Results on Weibo, Reddit, Disney, Books, Enron datasets w.r.t AUC.

Algorithm	Weibo	Reddit	Disney	Books	Enron
LOF	56.5 ± 0.0 (56.5)	57.2 ± 0.0 (57.2)	47.9 ± 0.0 (47.9)	36.5 ± 0.0 (36.5)	46.4 ± 0.0 (46.4)
IF	53.5 ± 2.8(57.5)	45.2 ± 1.7(47.5)	57.6 ± 2.9(63.1)	43.0 ± 1.8(47.5)	40.1 ± 1.4(43.1)
MLPAE	82.1 ± 3.6(86.1)	50.6 ± 0.0(50.6)	49.2 ± 5.7(64.1)	42.5 ± 5.6(52.6)	73.1 ± 0.0(73.1)
SCAN	63.7 ± 5.6(70.8)	49.9 ± 0.3(50.0)	50.5 ± 4.0(56.1)	49.8 ± 1.7(52.4)	52.8 ± 3.4(58.1)
Radar	**98.9 ± 0.1(99.0)**	54.9 ± 1.2(56.9)	51.8 ± 0.0(51.8)	52.8 ± 0.0(52.8)	80.8 ± 0.0(80.8)
ANOMALOUS	**98.9 ± 0.1(99.0)**	54.9 ± 5.6(60.4)	51.8 ± 0.0(51.8)	52.8 ± 0.0(52.8)	80.8 ± 0.0(80.8)
GCNAE	90.8 ± 1.2(92.5)	50.6 ± 0.0(50.6)	42.2 ± 7.9(52.7)	50.0 ± 4.5(57.9)	66.6 ± 7.8(80.1)
DOMINANT	85.0 ± 14.6(92.5)	56.0 ± 0.2(56.4)	47.1 ± 4.5(54.9)	50.1 ± 5.0(58.1)	73.1 ± 8.9(85.0)
DONE	85.3 ± 4.1(88.7)	53.9 ± 2.9(59.7)	41.7 ± 6.2(50.6)	43.2 ± 4.0(52.6)	46.7 ± 6.1(67.1)
AdONE	84.6 ± 2.2(87.6)	50.4 ± 4.5(58.1)	48.8 ± 5.1(59.2)	53.6 ± 2.0(56.1)	44.5 ± 2.9(53.6)
AnomalyDAE	91.5 ± 1.2(92.8)	55.7 ± 0.4(56.3)	48.8 ± 2.2(55.4)	62.2 ± 8.1(73.2)	54.3 ± 11.2(69.1)
GAAN	92.5 ± 0.0(92.5)	55.4 ± 0.4(56.0)	48.0 ± 0.0(48.0)	54.9 ± 5.0(61.9)	73.1 ± 0.0(73.1)
GUIDE	OOM_C	OOM_C	38.8 ± 8.9(52.5)	48.4 ± 4.6(63.5)	OOM_C
CONAD	85.4 ± 14.3(92.7)	56.1 ± 0.1(56.4)	48.0 ± 3.5(53.1)	52.2 ± 6.9(62.9)	71.9 ± 4.9(84.9)
G3AD	95.1 ± 1.35(96.5)	62.1 ± 0.22(63.1)	65.3 ± 1.7(67.6)	54.0 ± 4.2(58.7)	72.39 ± 2.9(75.4)
MuSE	89.7 ± 4.1(95.3)	53.7 ± 3.0(57.1)	67.3 ± 1.5(69.1)	64.3 ± 2.1(66.8)	64.0 ± 3.77(68.2)
OCGIN	73.2 ± 2.8(76.2)	51.8 ± 1.9(53.4)	56.1 ± 1.5(57.7)	64.4 ± 2.5(67.8)	54.1 ± 1.8(56.5)
GAD-NR	87.71 ± 5.39(92.09)	57.99 ± 1.67(59.90)	76.76 ± 2.75(80.03)	65.71 ± 4.98(69.79)	80.87 ± 2.95(82.92)
GEL	89.32±3.36 (92.92)	**62.76±2.42(64.20)**	**78.21±4.94(82.48)**	**70.79±3.16(74.20)**	**82.34±2.93(85.76)**

Table 2: Results on Weibo, Disney, Enron datasets w.r.t Recall@K.

Dataset	MLPAE	SCAN	Radar	GCNAE	DONE	AdONE	GAAN	GUIDE	CONAD	GAD-NR	GEL
Weibo (Recall@500)	51.82 ± 0.22	11.53 ± 0.00	53.60 ± 0.00	52.74 ± 0.60	42.94 ± 5.29	48.30 ± 3.09	53.14 ± 0.39	OOM_C	26.28 ± 4.12	60.22 ± 2.67	**65.53 ± 5.23**
Disney (Recall@50)	32.86 ± 12.66	39.29 ± 0.00	39.29 ± 0.00	47.14 ± 12.04	43.57 ± 4.16	37.14 ± 5.80	40.71 ± 6.62	40.00 ± 2.67	21.43 ± 0.00	67.85 ± 4.10	**78.57 ± 2.70**
Enron (Recall@1000)	9.84 ± 4.11	7.38 ± 0.00	12.57 ± 0.00	8.20 ± 0.73	8.52 ± 4.19	2.13 ± 2.58	9.18 ± 1.31	OOM_C	10.38 ± 0.00	15.23 ± 1.41	**20.00 ± 3.94**

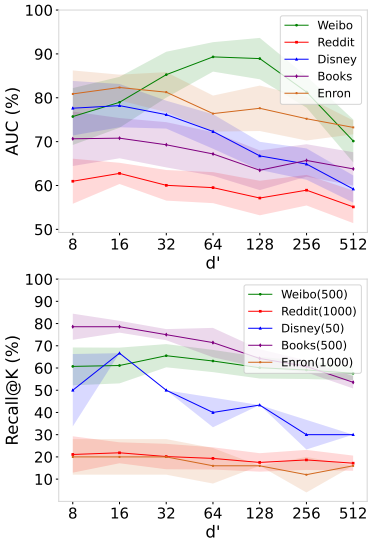


Figure 4: Impact of hidden layer dimension.

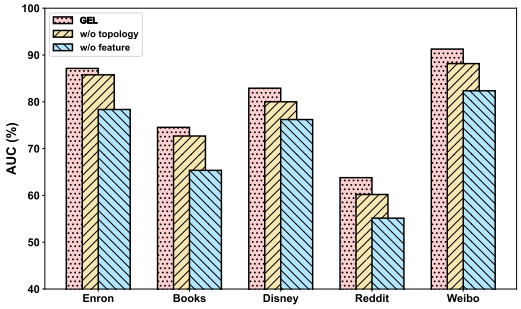


Figure 5: Impact of removing Different Modality.

structural information in graph data. By quantifying uncertainty in graph reconstruction, GEL uncovers hidden anomalous patterns and achieves robust anomaly detection.

- GEL shows substantial improvements on the Reddit and Books datasets, with AUC increases of 8.2% and 7.7%, respectively. The low anomaly rates (less than 3.5%) in these datasets highlight GEL's precision in handling imbalanced data distributions.
- GAE-based methods (e.g., AnomalyDAE, DOMINANT, GCNAE, DONE) generally surpass traditional methods (LOF, IF, MLPAE, SCAN) due to their ability to learn key features and capture complex graph relationships. Traditional methods require manual feature selection, challenging in high-dimensional datasets, and often generalize poorly. For example, LOF detects anomalies based on local density but fails to identify attribute-based anomalies. Similarly, GEL employs graph reconstruction and quantifies uncertainty to effectively handle complex data structures and reveal inherent uncertainties.

5.3 Ablation Study

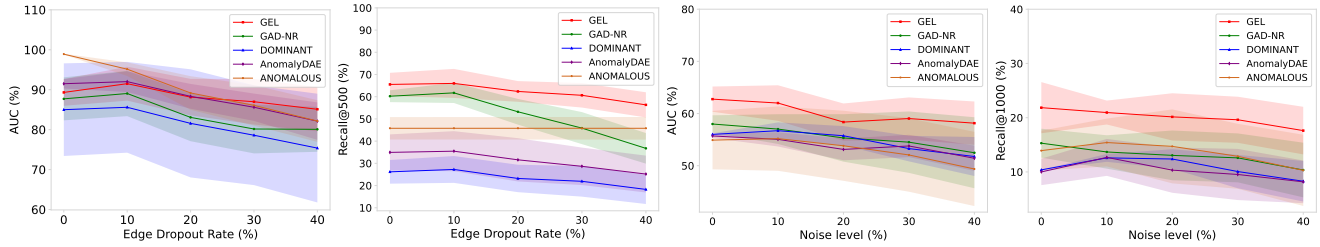
Modality heterogeneity in graph reconstruction challenges GEL to integrate uncertainties from both node features and topological structures. To assess the impact of uncertainties in different modalities on GEL's performance, we conducted an ablation study with two variants: 1) **w/o feature**: GEL disregards uncertainty in node attributes by removing the feature evidence estimate, relying solely on topological structure uncertainty for anomaly detection. 2) **w/o topology**: GEL ignores uncertainty in the topological structure by removing the topology evidence estimate, using only node attribute uncertainty for anomaly detection.

Figure 5 presents our key observations: 1) GEL achieves optimal performance when uncertainties from both modalities are considered; removing any modality results in a noticeable performance decline. 2) The performance degradation is more pronounced when the feature modality is omitted, indicating that uncertainty in node attributes plays a critical role in anomaly detection. This suggests that node attributes provide richer information, enhancing the model's capability to detect anomalies.

5.4 Parameter study

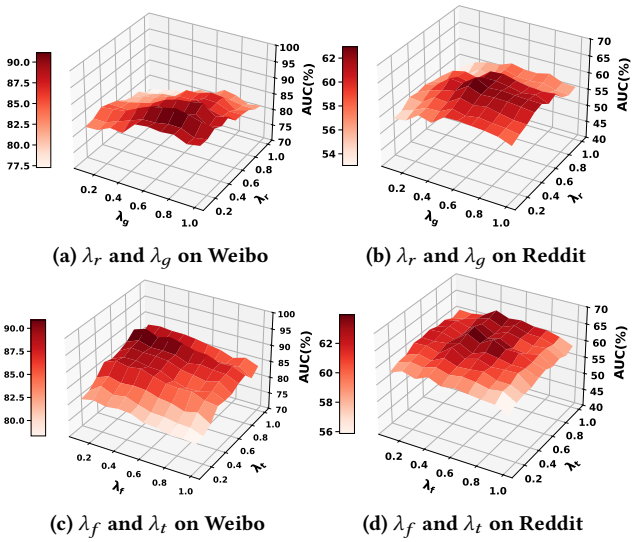
5.4.1 Impact of Latent Dimension d'. Figure 4 illustrates GEL's performance on the Weibo and Disney datasets as the latent dimension d' increases. We observe that performance improves with larger d', enabling the model to capture more reconstruction evidence and leading to a more accurate high-order evidential distribution, thus enhancing uncertainty quantification.

For the Weibo dataset, performance increases as d' grows from 8 to 128; for the Disney dataset, it improves from 8 to 16. This difference likely stems from the varying node feature dimensions: Weibo has 400-dimensional features, whereas Disney has 28-dimensional



(a) Weibo with dropout (b) Weibo with dropout (c) Reddit with noise (d) Reddit with noise

Figure 6: Changes of AUC and Recall@K as the level of data disturbance increases.



(a) λ_r and λ_g on Weibo (b) λ_r and λ_g on Reddit

(c) λ_f and λ_t on Weibo (d) λ_f and λ_t on Reddit

Figure 7: Impacts of weights on diverse uncertainty.

features. Datasets with higher-dimensional node features require larger latent dimensions to capture richer evidence.

However, when d' exceeds optimal values, performance declines due to overfitting. Overfitting is particularly detrimental in self-supervised models like GEL, as it leads to capturing noise and irrelevant details, reducing performance. GEL, based on a GAE architecture, aims to capture essential node information for uncertainty quantification; an excessively large d' may hinder its generalization ability.

5.4.2 Impact of λ_r and λ_g. As detailed in Section 4.4, GEL models uncertainty through reconstruction uncertainty (U_{reconst}), reflecting the amount of evidence in reconstruction, and graph uncertainty (U_{graph}), representing the imbalance or adversarial nature of the evidence. We examine the impact of the weights λ_r and λ_g by adjusting them accordingly.

Figure 7(a) and (b) illustrate that on the Weibo dataset, GEL achieves optimal performance when $\lambda_r = 0.7$ and $\lambda_g = 0.3$. This indicates that anomaly detection on Weibo relies more on reconstruction uncertainty, with moderate graph uncertainty aiding in handling adversarial evidence. A higher weight on reconstruction uncertainty allows the model to capture sufficient evidence for constructing a high-order evidential distribution, while a moderate weight on graph uncertainty helps manage the adversarial aspects effectively. In contrast, on the Reddit dataset, the best performance occurs when $\lambda_r = 0.5$ and $\lambda_g = 0.5$, suggesting that equal consideration of both uncertainties is beneficial. This finding implies that the optimal weighting of uncertainties is influenced by dataset characteristics and anomaly patterns. Specifically, anomalies in Weibo involve insufficient evidence, emphasizing reconstruction uncertainty, whereas anomalies in Reddit are more affected by evidence conflicts. These results demonstrate the flexibility and effectiveness of GEL in handling different anomaly patterns through the two types of uncertainty.

5.4.3 Impact of λ_f and λ_t. We evaluated the effect of varying λ_f and λ_t on GEL's performance using the Weibo and Reddit datasets. As shown in Figure 7(c) and (d), GEL achieves optimal performance on Weibo when $\lambda_f = 0.8$ and $\lambda_t = 0.2$, whereas on Reddit, the best performance occurs at $\lambda_f = 0.6$ and $\lambda_t = 0.4$.

These results indicate that a higher λ_f makes anomaly detection more reliant on feature uncertainty estimates ($\boldsymbol{\gamma}$, $\boldsymbol{\nu}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$), which is beneficial for datasets with significant feature uncertainty differences. Conversely, a higher λ_t emphasizes topology uncertainty estimates ($\boldsymbol{\epsilon}$, $\bar{\boldsymbol{\epsilon}}$), aiding in scenarios with pronounced topology uncertainty differences.

The differing optimal weights suggest that the importance of feature and topology uncertainties varies across datasets, highlighting the need to balance λ_f and λ_t according to dataset characteristics.

5.4.4 Impact of λ_1, λ_2, λ_3 and λ_4. We conducted experiments on multi-task training weights (λ_1, λ_3), setting $\lambda_2 = 1 - \lambda_1$ and $\lambda_4 = 1 - \lambda_3$ for efficiency. Results on Weibo show optimal performance at $\lambda_1 = 0.7$ and $\lambda_3 = 0.3$ (AUC: 89.22 ± 3.21), with similar patterns across other datasets.

Table 3: AUC Results for Different λ_1 and λ_3 on Weibo

$\lambda_3 \backslash \lambda_1$	0.1	0.3	0.5	0.7	0.9
0.1	82.02 ± 1.19	84.51 ± 3.17	84.93 ± 1.33	86.14 ± 0.22	86.28 ± 1.57
0.3	81.85 ± 3.24	83.93 ± 2.59	85.86 ± 3.29	**89.22 ± 3.21**	88.49 ± 2.54
0.5	79.76 ± 2.72	82.41 ± 3.52	85.02 ± 2.43	87.57 ± 2.70	88.24 ± 0.83
0.7	80.09 ± 2.08	83.13 ± 2.55	82.69 ± 1.95	84.16 ± 0.39	85.11 ± 3.01
0.9	79.22 ± 1.21	81.67 ± 1.96	83.57 ± 2.85	84.40 ± 3.12	83.61 ± 2.47

5.5 Robustness Study

To evaluate robustness, we compared GEL with baselines under varying levels of noise and structural perturbations. Introducing increasing levels of noise to node features (Figure 6(c) and (d)), we observed that while performance declined in all models due to

their reliance on feature quality, GEL exhibited significantly less degradation than the baselines.

Similarly, we assessed the models under different proportions of edge dropout (Figure 6(a) and (b)). At a dropout rate of 10%, GEL and most baselines showed improved Recall@K, likely due to the regularization effect enhancing generalization [54]. However, as dropout rates increased further, performance declined in all models. Notably, under higher dropout conditions, GEL's performance degradation was much smaller compared to GAD-NR.

GEL's robustness arises from its construction of high-order evidential and reconstruction distributions by capturing evidence from multiple dimensions to quantify uncertainty. This enables GEL to better handle missing or noisy data, maintaining relatively stable performance even under significant perturbations.

5.6 Complexity Analysis

GEL's computational complexity is comparable to standard GAE models with minimal additional overhead. We provide analysis from theoretical, implementation, and experimental perspectives. From a theoretical perspective, GEL's complexity remains in line with standard GAEs:

- Encoder: $O(E \cdot Z + N \cdot Z \cdot D)$ – identical to standard GAEs.
- Feature evidence network (f_{θ_1}): $O(N \cdot Z \cdot D)$ – comparable to feature decoder in GAEs.
- Topology evidence network (f_{θ_2}): $O(E \cdot Z \cdot D)$ – comparable to topology decoder in GAEs.
- Uncertainty computation: $O(N \cdot D + E)$ – negligible additional cost.

Where N is the number of nodes, E is the number of edges, Z is the latent dimension, and D is the feature dimension. The computational bottleneck remains in the encoder part, which is identical to standard GAE models.

The primary difference in GEL's decoder is outputting distribution parameters instead of point estimates, which adds negligible overhead in modern GPU-accelerated frameworks due to parallelization.

We compared training and inference times with the recent GAD-NR method based on GAE, which is a strong baseline. The results are shown in Tables 4 and 5.

Table 4: Training Time Comparison

Algorithm	Weibo	Reddit	Disney	Books	Enron
GAD-NR	1m51s	1m24s	57s	1m12s	1m16s
GEL	2m1s	1m29s	1m5s	1m22s	1m24s

Table 5: Inference Time Comparison

Algorithm	Weibo	Reddit	Disney	Books	Enron
GAD-NR	12s	11.1s	4.2s	6.2s	10.6s
GEL	16.5s	14.8s	5.8s	8.7s	12.3s

We also evaluated GEL on the large-scale DGraphFin dataset (3.7M nodes, 4.3M edges) and compared it with the GAD-NR. The results are shown in Table 6.

Table 6: Performance on DGraphFin Dataset

Method	AUC	Recall@10000	Training	Inference	Memory
GAD-NR	69.93	45.46	35m39s	7m57s	21.9GB
GEL	74.57	51.34	41m34s	8m9s	23.5GB

These results show that GEL maintains a 6.64% higher AUC than GAD-NR at million-node scale, requires only 17% more training time, and operates within the memory constraints of a single NVIDIA RTX 3090 GPU (24GB).

6 Conclusions and Limitations

We introduced *Graph Evidential Learning* (GEL), a novel framework that shifts graph anomaly detection from relying on reconstruction error to modeling uncertainty. GEL addresses the challenges of uncertainty diversity and modality heterogeneity by employing higher-order evidential distributions for both node features and topological structures. GEL quantifies both graph uncertainty and reconstruction uncertainty, enhancing robustness against noise and overfitting. Extensive experiments on benchmark datasets demonstrate that GEL achieves state-of-the-art performance.

While GEL advances graph anomaly detection through uncertainty modeling, our framework is currently oriented towards static graphs with fixed structures and attributes; extending GEL to dynamic graphs where topology and features evolve over time presents an avenue for future work. Lastly, although we employ specific evidential distributions tailored to continuous and discrete modalities, these choices may not capture all types of uncertainties in diverse datasets. Exploring alternative or more flexible evidential distributions could further enhance GEL's ability to detect a wider variety of anomalies.

Acknowledgments

This research was supported by the National Key R&D Program of China (No. 2023YFC3304701), NSFC (No.62132017 and No.U2436209), the Shandong Provincial Natural Science Foundation (No.ZQ2022JQ32), the Beijing Natural Science Foundation (L247027), the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China, and the Young Elite Scientists Sponsorship Program by CAST under contract No. 2022QNRC001. It was also supported by Big Data and Responsible Artificial Intelligence for National Governance, Renmin University of China.

References

[1] Taiga Abe, Estefany Kelly Buchanan, Geoff Pleiss, Richard S. Zemel, and John P. Cunningham. 2022. Deep Ensembles Work, But Are They Necessary?. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[2] Eduardo Aguilar, Bogdan Raducanu, Petia Radeva, and Joost van de Weijer. 2023. Continual Evidential Deep Learning for Out-of-Distribution Detection. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023 - Workshops, Paris, France, October 2-6, 2023*. IEEE, 3436–3446.

[3] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. 2020. Deep evidential regression. *Advances in neural information processing systems* 33 (2020), 14927–14937.

[4] Sambaran Bandyopadhyay, N. Lokesh, and M. Narasimha Murty. 2019. Outlier Aware Network Embedding for Attributed Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium*

on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. AAAI Press, 12–19.

[5] Sambaran Bandyopadhyay, Lokesh N, Saley Vishal Vivek, and M. Narasimha Murty. 2020. Outlier Resistant Unsupervised Deep Architectures for Attributed Network Embedding. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 25–33.

[6] Yuanchen Bei, Sheng Zhou, Jinke Shi, Yao Ma, Haishuai Wang, and Jiajun Bu. 2024. Guarding Graph Neural Networks for Unsupervised Graph Anomaly Detection. *arXiv preprint arXiv:2404.16366* (2024).

[7] Txus Blasco, J Salvador Sánchez, and Vicente García. 2024. A survey on uncertainty quantification in deep learning for financial time series prediction. *Neurocomputing* 576 (2024), 127339.

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*. PMLR, 1613–1622.

[9] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein (Eds.). ACM, 93–104.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 3 (2009), 15:1–15:58.

[11] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. 2020. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in neural information processing systems* 33 (2020), 1356–1367.

[12] Tianyi Chen and Charalampos E. Tsourakakis. 2022. AntiBenford Subgraphs: Unsupervised Anomaly Detection in Financial Networks. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 2762–2770.

[13] Zhenxing Chen, Bo Liu, Meiqing Wang, Peng Dai, Jun Lv, and Liefeng Bo. 2020. Generative Adversarial Attributed Network Anomaly Detection. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1989–1992.

[14] Peng Cui and Jinjia Wang. 2022. Out-of-distribution (OOD) detection based on deep learning: A review. *Electronics* 11, 21 (2022), 3500.

[15] Arthur P. Dempster. 2008. A Generalization of Bayesian Inference. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, Ronald R. Yager and Liping Liu (Eds.). Studies in Fuzziness and Soft Computing, Vol. 219. Springer, 73–104.

[16] Ping Deng and Yong Huang. 2025. Edge-featured multi-hop attention graph neural network for intrusion detection system. *Comput. Secur.* 148 (2025), 104132.

[17] Persi Diaconis and Donald Ylvisaker. 1979. Conjugate priors for exponential families. *The Annals of statistics* (1979), 269–281.

[18] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep Anomaly Detection on Attributed Networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2-4, 2019*, Tanya Y. Berger-Wolf and Nitesh V. Chawla (Eds.). SIAM, 594–602.

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

[20] Zhiguo Ding and Minrui Fei. 2013. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes* 46, 20 (2013), 12–17.

[21] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. 2020. Anomalydae: Dual Autoencoder for Anomaly Detection on Attributed Networks. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 5685–5689.

[22] Vincent Fortuin, Adrià Garriga-Alonso, Mark van der Wilk, and Laurence Aitchison. 2021. BNNpriors: A library for Bayesian neural network inference with different prior distributions. *Software Impacts* 9 (2021), 100079.

[23] Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a Bayesian approximation. *arXiv preprint arXiv:1506.02157* (2015).

[24] Aleksandr N. Grekov, Aleksey A. Kabanov, Elena V. Vyshkvarkova, and Valeriy V. Trusevich. 2023. Anomaly Detection in Biological Early Warning Systems Using Unsupervised Machine Learning. *Sensors* 23, 5 (2023), 2687.

[25] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 855–864.

[26] Waleed Hilal, S. Andrew Gadsden, and John Yawney. 2022. Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. *Expert Syst. Appl.* 193 (2022), 116429.

[27] Weiming Hu, Jun Gao, Bing Li, Ou Wu, Junping Du, and Stephen J. Maybank. 2020. Anomaly Detection Using Local Kernel Density Estimation and Context-Based Regression. *IEEE Trans. Knowl. Data Eng.* 32, 2 (2020), 218–233.

[28] Chao Huang, Chengliang Liu, Zheng Zhang, Zhihao Wu, Jie Wen, Qiuping Jiang, and Yong Xu. 2022. Pixel-Level Anomaly Detection via Uncertainty-aware Prototypical Transformer. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, João Magalhães, Alberto Del Bimbo, Shin'ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni (Eds.). ACM, 521–530.

[29] Ling Huang, XuanLong Nguyen, Minos Garofalakis, Michael Jordan, Anthony Joseph, and Nina Taft. 2006. In-network PCA and anomaly detection. *Advances in neural information processing systems* 19 (2006).

[30] Audun Jøsang. 2016. *Subjective Logic - A Formalism for Reasoning Under Uncertainty*. Springer.

[31] Hwan Kim, Byung Suk Lee, Won-Yong Shin, and Sungsu Lim. 2022. Graph anomaly detection with graph neural networks: Current status and challenges. *IEEE Access* 10 (2022), 111820–111829.

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

[33] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR* abs/1611.07308 (2016).

[34] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[35] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[36] Atsutoshi Kumagai, Tomoharu Iwata, and Yasuhiro Fujiwara. 2021. Semi-supervised anomaly detection on attributed graphs. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[37] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 1269–1278.

[38] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30 (2017).

[39] Jundong Li, Harsh Dani, Xia Hu, and Huan Liu. 2017. Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 2152–2158.

[40] Longyuan Li, Junchi Yan, Haiyang Wang, and Yaohui Jin. 2020. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE transactions on neural networks and learning systems* 32, 3 (2020), 1177–1191.

[41] Yang Li, Binxing Fang, Li Guo, and You Chen. 2007. Network anomaly detection based on TCM-KNN algorithm. In *Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, Singapore, March 20-22, 2007*, Feng Bao and Steven Miller (Eds.). ACM, 13–19.

[42] Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, Lichao Sun, Jundong Li, George H. Chen, Zhihao Jia, and Philip S. Yu. 2022. BOND: Benchmarking Unsupervised Outlier Node Detection on Static Attributed Graphs. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[43] Yuanye Liu, Zheyao Gao, Nannan Shi, Fuping Wu, Yuxin Shi, Qingchao Chen, and Xiahai Zhuang. 2024. MERIT: Multi-view Evidential learning for Reliable and Interpretable liver fibrosis sTaging. *arXiv preprint arXiv:2405.02918* (2024).

[44] Zhiyuan Liu, Chunjie Cao, and Jingzhang Sun. 2022. Mul-gad: a semi-supervised graph anomaly detection framework via aggregating multi-view information. *arXiv preprint arXiv:2212.05478* (2022).

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

[46] Milad Memarzadeh, Bryan Matthews, and Ilya Avrekh. 2020. Unsupervised anomaly detection in flight data using convolutional variational auto-encoder. *Aerospace* 7, 8 (2020), 115.

[47] Rhiannon Michelmore, Marta Kwiatkowska, and Yarin Gal. 2018. Evaluating uncertainty quantification in end-to-end autonomous driving control. *arXiv preprint arXiv:1811.06817* (2018).

[48] Emmanuel Müller, Patricia Iglesias Sánchez, Yvonne Mülle, and Klemens Böhm. 2013. Ranking outlier nodes in subspaces of attributed graphs. In *Workshops Proceedings of the 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, Chee Yong Chan, Jiaheng Lu, Kjetil

Nørvåg, and Egemen Tanin (Eds.). IEEE Computer Society, 216–222.
[49] Anvardh Nanduri and Lance Sherry. 2016. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In *2016 Integrated Communications Navigation and Surveillance (ICNS)*. Ieee, 5C2–1.
[50] Venkat Nemani, Luca Biggio, Xun Huan, Zhen Hu, Olga Fink, Anh Tran, Yan Wang, Xiaoge Zhang, and Chao Hu. 2023. Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mechanical Systems and Signal Processing* 205 (2023), 110796.
[51] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)* 54, 2 (2021), 1–38.
[52] Giorgio Parisi and Ramamurti Shankar. 1988. Statistical field theory. (1988).
[53] Zhen Peng, Minnan Luo, Jundong Li, Huan Liu, and Qinghua Zheng. 2018. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 3513–3519.
[54] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
[55] Amit Roy, Juan Shu, Jia Li, Carl Yang, Olivier Elshocht, Jeroen Smeets, and Pan Li. 2024. GAD-NR: Graph Anomaly Detection via Neighborhood Reconstruction. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, Luz Angelica Caudillo-Mata, Silvio Lattanzi, Andrés Muñoz Medina, Leman Akoglu, Aristides Gionis, and Sergei Vassilvitskii (Eds.). ACM, 576–585.
[56] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, Gold Coast, Australia, QLD, Australia, December 2, 2014*, Ashfaqur Rahman, Jeremiah D. Deng, and Jiuyong Li (Eds.). ACM, 4.
[57] Patricia Iglesias Sánchez, Emmanuel Müller, Fabian Laforet, Fabian Keller, and Klemens Böhm. 2013. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs. In *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, Hui Xiong, George Karypis, Bhavani Thuraisingham, Diane J. Cook, and Xindong Wu (Eds.). IEEE Computer Society, 647–656.
[58] David Savage, Xiuzhen Zhang, Xinghuo Yu, Pauline Lienhua Chou, and Qingmai Wang. 2016. Anomaly detection in online social networks. *CoRR* abs/1608.00301 (2016).
[59] Murat Sensoy, Lance Kaplan, and Melih Kandemir. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems* 31 (2018).
[60] Cengiz Togay, Ahmet Kasif, Cagatay Catal, and Bedir Tekinerdogan. 2022. A Firewall Policy Anomaly Detection Framework for Reliable Network Security. *IEEE Trans. Reliab.* 71, 1 (2022), 339–347.
[61] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. 2007. SCAN: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, Pavel Berkhin, Rich Caruana, and Xindong Wu (Eds.). ACM, 824–833.
[62] Zhiming Xu, Xiao Huang, Yue Zhao, Yushun Dong, and Jundong Li. 2022. Contrastive Attributed Network Anomaly Detection with Data Augmentation. In *Advances in Knowledge Discovery and Data Mining - 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16-19, 2022, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 13281)*, João Gama, Tianrui Li, Yang Yu, Enhong Chen, Yu Zheng, and Fei Teng (Eds.). Springer, 444–457.
[63] Bang Xiang Yong and Alexandra Brintrup. 2022. Bayesian autoencoders with uncertainty quantification: Towards trustworthy anomaly detection. *Expert Syst. Appl.* 209 (2022), 118196.
[64] Xu Yuan, Na Zhou, Shuo Yu, Huafei Huang, Zhikui Chen, and Feng Xia. 2021. Higher-order Structure Based Anomaly Detection on Attributed Networks. In *2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15-18, 2021*, Yixin Chen, Heiko Ludwig, Yicheng Tu, Usama M. Fayyad, Xingquan Zhu, Xiaohua Hu, Suren Byna, Xiong Liu, Jianping Zhang, Shirui Pan, Vagelis Papalexakis, Jianwu Wang, Alfredo Cuzzocrea, and Carlos Ordonez (Eds.). IEEE, 2691–2700.
[65] Zirui Yuan, Minglai Shao, and Qiben Yan. 2023. Motif-Level Anomaly Detection in Dynamic Graphs. *IEEE Trans. Inf. Forensics Secur.* 18 (2023), 2870–2882.
[66] Qingfeng Zeng, Li Lin, Rui Jiang, Weiyu Huang, and Dijia Lin. 2025. NNEnsLeG: A novel approach for e-commerce payment fraud detection using ensemble learning and neural networks. *Inf. Process. Manag.* 62, 1 (2025), 103916.
[67] Lingxiao Zhao and Leman Akoglu. 2023. On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights. *Big Data* 11, 3 (2023), 151–180.
[68] Tong Zhao, Chuchen Deng, Kaifeng Yu, Tianwen Jiang, Daheng Wang, and Meng Jiang. 2020. Error-Bounded Graph Anomaly Loss for GNNs. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1873–1882.
[69] Lecheng Zheng, John R. Birge, Yifang Zhang, and Jingrui He. 2024. Towards Multi-view Graph Anomaly Detection with Similarity-Guided Contrastive Clustering. *CoRR* abs/2409.09770 (2024).
[70] Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN.. In *IJCAI*, Vol. 3. 7.
[71] Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu. 2019. One-Class Adversarial Nets for Fraud Detection. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 1286–1293.

A Derivation of $\mathcal{L}_f^{\text{NLL}}$

In this subsection, we derive the negative log-likelihood loss for feature (ie. Eq. 12) of a NIG distribution. For convenience, we omit the subscript $_{ih}$ of the hyperparameters $\gamma, \upsilon, \alpha, \beta$ and denote them as as $\mathbf{m}$. We have:

$$
\begin{aligned}
& p(\mathbf{X}_{ih}|\mathbf{m}) \\
&= \int_\theta p(\mathbf{X}_{ih}|\theta)p(\theta|\mathbf{m})\, d\theta \\
&= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} p(\mathbf{X}_{ih}|\mu,\sigma^2)p(\mu,\sigma^2|\mathbf{m})\, d\mu d\sigma^2 \\
&= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} p(\mathbf{X}_{ih}|\mu,\sigma^2)p(\mu,\sigma^2|\gamma,\upsilon,\alpha,\beta)\, d\mu d\sigma^2 \\
&= \int_{\sigma^2=0}^{\infty}\int_{\mu=-\infty}^{\infty} \left[\sqrt{\frac{1}{2\pi\sigma^2}}\exp\left\{-\frac{(\mathbf{X}_{ih}-\mu)^2}{2\sigma^2}\right\}\right] \\
&\qquad \left[\frac{\beta^\alpha\sqrt{\upsilon}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}\left(\frac{1}{\sigma^2}\right)^{\alpha+1}\exp\left\{-\frac{2\beta+\upsilon(\gamma-\mu)^2}{2\sigma^2}\right\}\right] d\mu d\sigma^2 \\
&= \int_{\sigma^2=0}^{\infty} \frac{\beta^\alpha\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/\upsilon}\Gamma(\alpha)}\exp\left\{-\frac{2\beta+\frac{\upsilon(\mathbf{X}_{ih}-\gamma)^2}{1+\upsilon}}{2\sigma^2}\right\} d\sigma^2 \\
&= \int_{\sigma=0}^{\infty} \frac{\beta^\alpha\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/\upsilon}\Gamma(\alpha)}\exp\left\{-\frac{2\beta+\frac{\upsilon(\mathbf{X}_{ih}-\gamma)^2}{1+\upsilon}}{2\sigma^2}\right\} 2\sigma d\sigma \\
&= \frac{\Gamma(1/2+\alpha)}{\Gamma(\alpha)}\sqrt{\frac{\upsilon}{\pi}}\left(2\beta(1+\upsilon)\right)^\alpha\left(\upsilon(\mathbf{X}_{ih}-\gamma)^2+2\beta(1+\upsilon)\right)^{-(\frac{1}{2}+\alpha)},
\end{aligned}
$$

which is equivalent to:

$$
p(\mathbf{X}_{ih}|\mathbf{m}) = \text{St}\left(\mathbf{X}_{ih};\gamma,\frac{\beta(1+\upsilon)}{\upsilon\,\alpha},2\alpha\right).
$$

$\text{St}\left(y;\mu_{\text{St}},\sigma^2_{\text{St}},\upsilon_{St}\right)$ is the Student-t distribution evaluated at y with location parameter μ_{St}, scale parameter σ^2_{St}, and υ_{St} degrees of freedom. Afterwards, we can compute the negative log likelihood loss, $\mathcal{L}_{ih}$, for reconstruction $\mathbf{X}_{ih}$ as:

$$
\begin{aligned}
\mathcal{L}_{ih} &= -\log p(\mathbf{X}_{ih}|\mathbf{m}) \\
&= -\log\left(\text{St}\left(\mathbf{X}_{ih};\gamma,\frac{\beta(1+\upsilon)}{\upsilon\,\alpha},2\alpha\right)\right),
\end{aligned}
$$

which has the following derivation:

$$\begin{aligned}\mathcal{L}_{ih} =& \tfrac{1}{2}\log\left(\tfrac{\pi}{v}\right) \\ &- \alpha\log(\Omega) + \left(\alpha+\tfrac{1}{2}\right)\log((y-\gamma)^2 v + \Omega) \\ &+ \log\left(\tfrac{\Gamma(\alpha)}{\Gamma(\alpha+\frac{1}{2})}\right)\end{aligned}$$

where $\Omega = 2\beta(1+v)$.

B Baseline Methods

- **LOF** [9]: Local Outlier Factor measures node isolation by comparing its density to k-nearest neighbors using node features.
- **IF** [4]: Isolation Forest uses decision trees, scoring anomalies by their proximity to the root.
- **MLPAE** [56]: Multi-Layer Perceptron Autoencoder reconstructs node features, using reconstruction loss as the anomaly score.
- **SCAN** [61]: Structural Clustering Algorithm for Networks identifies clusters and labels structurally distinct nodes as anomalies.
- **Radar** [39]: Radar detects anomalies using attribute residuals and structural coherence, with reconstruction residuals as the score.
- **GCNAE** [34]: Graph Convolutional Network Autoencoder reconstructs node features and graph structure, using reconstruction error for anomaly detection.
- **DOMINANT** [19]: DOMINANT uses a two-layer GCN to reconstruct features and structure, with combined reconstruction error as the anomaly score.
- **DONE** [5]: DONE optimizes node embeddings and anomaly scores using MLPs in a unified framework.
- **AnomalyDAE** [21]: AnomalyDAE reconstructs structure and attributes using a structural autoencoder and attribute decoder.
- **GAAN** [13]: Graph Anomaly Adversarial Network uses a GAN framework, combining detection confidence and reconstruction loss for anomaly scoring.
- **GUIDE** [64]: GUIDE uses motif-based degree vectors and mirrors DONE for anomaly detection.
- **CONAD** [62]: CONAD applies graph augmentation and contrastive learning for anomaly detection.
- **G3AD** [6]: G3AD introduces an adaptive caching module to guard the GNNs from solely reconstructing the observed data that contains anomalies.
- **MuSE** [32]: MuSE uses the multifaceted summaries of reconstruction errors as indicators for anomaly detection.
- **OCGIN** [67]: OCGIN is an end-to-end graph outlier detection model that addresses the "performance flip" phenomenon by leveraging factors such as density disparity and overlapping support.
- **GAD-NR** [55]: Graph Anomaly Detection with Neighborhood Reconstruction enhances GAE by reconstructing a node's full neighborhood, using reconstruction loss to detect anomalies.

For baseline methods, we used publicly available implementations and followed the parameter settings recommended in their original papers. Our reproduced results closely match those reported in their respective publications.

For fair comparison across all methods, we performed grid search over hyperparameters including:

- Learning rate $\in \{0.0001, 0.001, 0.01\}$.
- Weight decay $\in \{0, 0.0001, 0.001\}$.
- Model-specific parameters following their respective papers.

C Dataset Overview

The details of these datasets are shown in Table 7.

- **Weibo** [68]: A directed user interaction network from Tencent-Weibo. Anomalous users are identified by temporal post patterns. Features include post location and bag-of-words content representation.
- **Reddit** [37]: A network of user-subreddit interactions on Reddit. Banned users are anomalies. Features are LIWC-based vectors from user and subreddit posts.
- **Disney** [48] and **Books** [57]: Co-purchase networks for movies (Disney) and books (Books). Anomalies are based on student votes (Disney) or Amazon failure tags (Books). Features include price, review count, and ratings.
- **Enron** [57]: An email interaction network. Spam-sending email addresses are anomalies. Features include average email length, recipient count, and email time intervals.

Table 7: Dataset statistics.

Statistic	Weibo	Reddit	Disney	Books	Enron
#Nodes	8405	10984	124	1418	13533
#Edges	407963	168016	335	3695	176987
#Features	400	64	28	21	18
#Degree	48.5	15.3	2.7	2.6	13.1
#Anomalies	868	366	6	28	5
Anomaly Ratio	10.3%	3.3%	4.8%	2.0%	0.04%

D Details of Netowrk Implemetation

To ensure compatibility with various GAE-based anomaly detection methods, we model f_{θ_1} and f_{θ_2} using lightweight Multi-Layer Perceptrons (MLPs). These networks are designed to output parameters that align with the prior definitions of their respective distributions. For $f_{\theta_1}(\cdot)$, we use the following architecture:

$$\boldsymbol{\gamma} = \mathbf{W}_3 \cdot \mathrm{Tanh}(\mathbf{W}_2^{\gamma} \cdot \mathrm{Tanh}(\mathbf{W}_1^{\gamma}\mathbf{Z} + \mathbf{b}_1^{\gamma}) + \mathbf{b}_2^{\gamma}) + \mathbf{b}_3^{\gamma}, \tag{17}$$

$$\boldsymbol{\nu} = \mathrm{ReLU}(\mathbf{W}_2^{\nu} \cdot \mathrm{ReLU}(\mathbf{W}_1^{\nu}\mathbf{Z} + \mathbf{b}_1^{\nu}) + \mathbf{b}_2^{\nu}), \tag{18}$$

$$\boldsymbol{\alpha} = \mathrm{ReLU}(\mathbf{W}_2^{\alpha} \cdot \mathrm{ReLU}(\mathbf{W}_1^{\alpha}\mathbf{Z} + \mathbf{b}_1^{\alpha}) + \mathbf{b}_2^{\alpha}) + 1, \tag{19}$$

$$\boldsymbol{\beta} = \mathrm{ReLU}(\mathbf{W}_2^{\beta} \cdot \mathrm{ReLU}(\mathbf{W}_1^{\beta}\mathbf{Z} + \mathbf{b}_1^{\beta}) + \mathbf{b}_2^{\beta}). \tag{20}$$

For $f_{\theta_2}(\cdot)$, the evidence parameters ε_{ij} and $\bar{\varepsilon}_{ij}$ for the existence and non-existence of edge e_{ij} are computed as:

$$[\varepsilon_{ij}, \bar{\varepsilon}_{ij}] = \mathrm{ReLU}(\mathbf{W}_2^{\varepsilon} \cdot \mathrm{ReLU}(\mathbf{W}_1^{\varepsilon} \cdot \mathrm{concat}([\mathbf{Z}_i, \mathbf{Z}_j]) + \mathbf{b}_1^{\varepsilon}) + \mathbf{b}_2^{\varepsilon}) + 1, \tag{21}$$

where $\mathrm{concat}([\mathbf{Z}_i, \mathbf{Z}_j])$ represents the concatenation of the latent embeddings of nodes v_i and v_j, and the ReLU activation ensures the non-negativity of the evidence. The final "+1" term ensures consistency with the definition of the Beta parameters $\varepsilon_{ij} = \mathbf{E}_{ij} + 1$ and $\bar{\varepsilon}_{ij} = \bar{\mathbf{E}}_{ij} + 1$.