arXiv:2506.07288v3 [cs.LG] 1 Aug 2025

EviNet:Towards Open-World Graph Learning via Evidential Reasoning Network

Weijie Guan
skjguan@vt.edu
Virginia Polytechnic Institute and State University
Blacksburg, VA, USA

Haohui Wang
haohuiw@vt.edu
Virginia Polytechnic Institute and State University
Blacksburg, VA, USA

Jian Kang
jian.kang@rochester.edu
University of Rochester
Rochester, NY, USA

Lihui Liu
hw6926@wayne.edu
Wayne State University
Detroit, MI, USA

Dawei Zhou
zhoud@vt.edu
Virginia Polytechnic Institute and State University
Blacksburg, VA, USA

Abstract

Graph learning has been crucial to many real-world tasks, but they are often studied with a closed-world assumption, with all possible labels of data known a priori. To enable effective graph learning in an open and noisy environment, it is critical to inform the model users when the model makes a wrong prediction to in-distribution data of a known class, i.e., misclassification detection or when the model encounters out-of-distribution from novel classes, i.e., out-of-distribution detection. This paper introduces Evidential Reasoning Network (EviNet), a framework that addresses these two challenges by integrating Beta embedding within a subjective logic framework. EviNet includes two key modules: Dissonance Reasoning for misclassification detection and Vacuity Reasoning for out-of-distribution detection. Extensive experiments demonstrate that EviNet outperforms state-of-the-art methods across multiple metrics in the tasks of in-distribution classification, misclassification detection, and out-of-distribution detection. EviNet demonstrates the necessity of uncertainty estimation and logical reasoning for misclassification detection and out-of-distribution detection and paves the way for open-world graph learning. Our code and data are available at https://github.com/SSSKJ/EviNET.

CCS Concepts

• **Computing methodologies** → **Probabilistic reasoning**.

Keywords

Out-of-distribution detection, open-world machine learning

ACM Reference Format:
Weijie Guan, Haohui Wang, Jian Kang, Lihui Liu, and Dawei Zhou. 2025. EviNet:Towards Open-World Graph Learning via Evidential Reasoning Network. 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/10.1145/3711896.3736945

This work is licensed under a Creative Commons Attribution 4.0 International License.
KDD '25, Toronto, ON, Canada
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736945

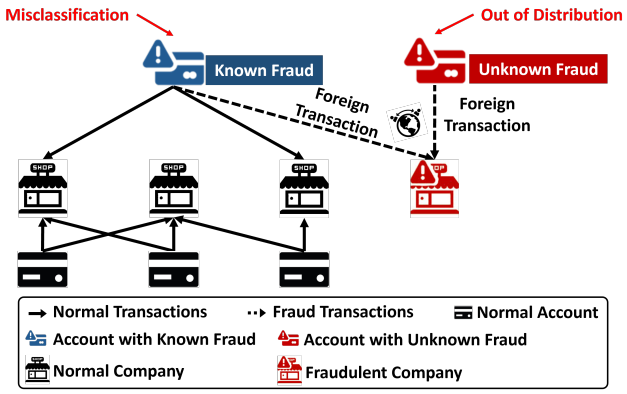


Figure 1: An illustrative figure of financial fraud situations in real-world applications. There are many undiscovered types of fraud, and even known fraudulent activities may incorporate legitimate transactions to mimic normal users and conduct noisy and covert fraud transactions, making detection increasingly challenging.

1 Introduction

Graphs capture relationships among entities in many real-world scenarios, including social networks [15, 31], recommendation systems [35, 37], and financial fraud detection [33, 34]. To date, researchers have developed a wealth of graph learning models, such as graph neural networks, to learn node representations on graphs [11, 17, 29]. These methods often learn the representation of a node by aggregating information from its neighbors and updating the model parameters by its label. An underlying assumption for training these graph learning models is that all possible labels are known a priori in a closed label set during training, which we term learning in *a closed-world environment.*

Yet, the assumption about the closed-world environment might often be violated in real-world applications due to ambiguous data or unknown and novel classes. Graph learning models trained with the closed-world assumption might fail in high-stakes applications such as financial fraud detection [30, 32]. For example, nearly 60% of banks, fintechs, and credit unions faced over $500K in direct fraud losses in 2023. Though financial fraud detection has been

actively studied for years and has shown strong performance in detecting known and commonly happened types of fraud, many types of fraud remain undiscovered (as the account marked as out-of-distribution shown in Figure 1). Meanwhile, there could also be accounts that mostly engage in normal transactions but occasionally reveal fraudulent intentions (as the account marked as misclassification shown in Figure 1), particularly in cross-border money laundering where different regulations across countries create noise in the data that might mislead the model to misclassify data. Thus, it is critical to operationalize a graph learning model in *open and noisy environments* by informing the model users (*C1. In-distribution misclassification detection*) when the graph learning model misclassifies data into one of the known classes or (*C2. Out-of-distribution detection*) when it encounters out-of-distribution (OOD) data from novel classes that do not appear during training.

Recognizing the importance, researchers have developed a collection of models for misclassification detection [8, 23] or OOD detection [12, 20, 36]. However, these existing techniques are often limited to tackling one facet of challenges in learning in open and noisy environments (i.e., either misclassification detection or OOD detection) and fail to address the multifaceted challenges posed by open and noisy environments. A few models, such as GPN [27] and GKDE [40], are capable of detecting both misclassification and OOD instances. However, GPN [27] relies heavily on the approximate posterior distributions of known classes to make predictions, which might make it difficult to detect subtle misclassifications. Moreover, GPN quantifies uncertainty based on the posterior distribution of known classes. It might not generate meaningful representations for detecting novel classes that are out-of-distribution. Regarding GKDE [40], it is a subjective-logic-based framework with a fixed prior weight that is equal to the number of known classes and with a variance-minimization regularization that minimizes the evidence with respect to a class if a data point does not belong to that class. This essentially leads to overconfidence in both misclassification and out-of-distribution detection [5]. Thus, it is critical to understand how to achieve effective misclassification detection and out-of-distribution detection while maintaining strong performance when classifying in-distribution data.

To this end, we propose a subjective-logic-guided framework named Evidential Reasoning Network (EviNet). EviNet learns node embeddings using the Beta embedding to enable logical reasoning capability and provides support regions for known classes and an implicit support region for novel classes. Based on that, EviNet quantifies uncertainty for misclassification detection and out-of-distribution detection with two key modules: (M1) Dissonance Reasoning that detects misclassified data and (M2) Vacuity Reasoning to detect out-of-distribution data from novel classes. Specifically, (M1) Dissonance Reasoning employs logical reasoning to generate class embeddings for known classes and computes a dissonance score using both class embeddings and node embeddings. A high dissonance score refers to a node that has a high probability of being classified into more than one class, which indicates potential misclassification. (M2) Vacuity Reasoning considers the potential novel class and derives representations for potential novel classes by logical disjunction and negation of the known class embeddings in Dissonance Reasoning. Then, it detects out-of-distribution data from novel classes by computing a vacuity score using node embeddings, class embeddings of known classes, and class embedding for novel classes. By combining these two modules, EviNet addresses the challenge of overconfidence and provides more reliable uncertainty estimates for both misclassification and OOD detection. We show that EviNet demonstrates strong performance in both misclassification and out-of-distribution detection while maintaining competitive performance for in-distribution classification. Our main contributions are as follows:

- **Problem.** We study graph learning in an open and noisy environment by addressing misclassification detection and OOD detection simultaneously.
- **Algorithm.** We propose EviNet which employs Beta embedding and subjective logic , providing support regions for known classes and an implicit support region for novel classes to estimate uncertainty for both misclassification and OOD detection.
- **Evaluation.** We systematically evaluate the performance of EviNet in in-distribution classification, misclassification detection, and OOD detection. Extensive experiments show the efficacy of EviNet in all three tasks.

The rest of this paper is organized as follows. We provide background information and problem definition in Section 2, followed by the proposed framework in Section 3. In Section 4, we discuss the experimental setup and results, followed by a literature review in Section 5. Finally, we conclude the paper in Section 6.

2 Preliminary

In this section, we first present key symbols and notations in this work and then offer a preliminary review of subjective logic and Beta embedding, which is core to our proposed EviNet.

Notations. In general, we represent a vector, a matrix, and a set with a bold lowercase letter ($\boldsymbol{b}$), bold uppercase letter ($\mathbf{X}$), and calligraphic letter ($\mathcal{V}$). In this work, we denote a graph as $(\mathbf{A}, \mathbf{X}, \mathcal{Y})$ or $(\mathbf{A}, \mathbf{X})$ without labels, where $\mathbf{A}$ is its adjacency matrix determined by a set of n nodes $\mathcal{V}$ and a set of m edges $\mathcal{E}$, $\mathbf{X}$ is the node feature matrix, and $\mathcal{Y}$ is the label set. Table 4 in Appendix A summarizes key symbols and their descriptions.

Subjective Logic. It is a probabilistic logic framework to manage belief and uncertainty and make inferences under incomplete and uncertain information [14]. Subjective opinions in subjective logic quantifies the beliefs about the truth of propositions under uncertainty, which could be a binomial opinion, multinomial opinion, or hyper opinion. We consider multinomial opinion because it considers beliefs for multiple possible outcomes and matches with the multi-class classification problem in our setting. Specifically, for a multi-class classification with label set $\mathcal{Y} = \{1, \ldots, K\}$, a multinomial opinion over $\mathcal{Y}$ is a triplet $\omega = (\boldsymbol{b}, u, \boldsymbol{a})$, where $\boldsymbol{b} = [b_1, \ldots, b_K]^T$ is the belief mass distribution over $\mathcal{Y}$, u is the uncertainty mass that represents vacuity of evidence, and $\boldsymbol{a} = [a_1, \ldots, a_K]^T$ is the base rate distribution over $\mathcal{Y}$. The multinomial opinion can be further represented as a Dirichlet distribution of the projected probability distribution $\boldsymbol{p} = [p_1, \ldots, p_K]$. Specifically, the expected probability for class $k \in \mathcal{Y}$ is

$$\mathrm{E}\left[p_k\right] = \frac{\xi_k}{\sum_{i=1}^{K} \xi_i} = \frac{e_k + a_k W}{W + \sum_{i=1}^{K} e_i}, \tag{1}$$

where $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_K\}$ is a vector of evidence parameters, e_k is the evidence for class k, and W is the prior weight representing uncertain evidence. To map the observed evidence to a multinomial opinion, the belief and uncertainty are computed as

$$b_k = \frac{e_k}{\sum_{i=1}^{K} \xi_i} = \frac{e_k}{W + \sum_{i=1}^{K} e_i}, \quad u = \frac{W}{\sum_{i=1}^{K} \xi_i} = \frac{W}{W + \sum_{i=1}^{K} e_i}, \quad (2)$$

such that the projected probability for class k is $p_k = b_k + a_k u$. Given the multinomial opinion $\omega = (\boldsymbol{b}, u, \boldsymbol{a})$, we can further understand the belief vacuity resulted from lack of knowledge as

$$\text{Vac}(\omega) = u = \frac{W}{W + \sum_{i=1}^{K} e_i} \quad (3)$$

and the dissonance, which reflects conflicting evidence, as

$$\text{Diss}(\omega) = \sum_{i=1}^{K} \left(\frac{b_i \sum_{j \neq i} b_j \, \text{Bal}(b_j, b_i)}{\sum_{j \neq i} b_j} \right) \quad (4)$$

where $\text{Bal}(b_j, b_i) = 1 - \left| b_j - b_i \right| / (b_j + b_i)$.

Beta Embedding. It is originally designed for logical reasoning on knowledge graphs [24]. Beta embedding is a probabilistic embedding framework to model uncertainty and perform logical operations within the embedding space. Specifically, a Beta embedding in a d-dimensional space is represented by a collection of independent Beta distributions $\{(\alpha_1, \beta_1), \ldots, (\alpha_d, \beta_d)\}$, where each pair of parameters (α, β) defines a Beta distribution. Then neural logical operators such as projection, intersection, and negation are applied by equipping these Beta distributions with neural networks.

Problem Definition. Graph Learning in Open and Noisy Environments is an important problem in open environments. A reliable model should not only output the classification prediction for samples but also assess how likely it is that a given sample is misclassified or belongs to novel classes. Most existing methods focus only on OOD detection [1, 10, 13, 39] or misclassification detection [8, 28]. This limited focus reduces the model's overall effectiveness and robustness in real-world applications. Building a reliable model for real-world environments requires a comprehensive approach that can jointly address both OOD detection and misclassification detection, ensuring that the model can handle the uncertainties inherent in these environments. Based on the above considerations, the problem addressed in this paper is formally defined as follows:

Problem 1. Graph learning in an open and noisy environment.
Given: An undirected attributed graph $(\mathbf{A}, \mathbf{X}, \mathcal{Y})$. **Find:** A model $f(\cdot)$ that, for each node, outputs (1) the predicted label, (2) a misclassification score indicating the likelihood that the prediction is misclassified, and (3) an out-of-distribution score indicating the likelihood that the node is out-of-distribution.

3 Proposed Method

In this section, we introduce EviNet, a unified framework to address both (*C1*) in-distribution misclassification detection and (*C2*) OOD detection. The core idea of EviNet is to estimate uncertainty by representing nodes and classes as Beta embeddings and reasoning with subjective logic. In particular, we first introduce the overall learning framework of EviNet. Then, we discuss two key modules in EviNet, including (M1) Dissonance Reasoning for misclassification detection and (M2) Vacuity Reasoning for out-of-distribution detection. Finally, we present an end-to-end optimization framework for training EviNet.

3.1 Framework Overview

An overview of our proposed framework is presented in Figure 2, which consists of two key modules: **(M1) Dissonance Reasoning** and **(M2) Vacuity Reasoning**. The Dissonance Reasoning module tackles (*C1*) in-distribution misclassification detection by generating class embeddings for known (in-distribution) classes from nodes and computing a dissonance score, which indicates the conflicting evidence towards multiple known classes, using the node embeddings and class embeddings.

The Vacuity Reasoning module builds on the generated class embeddings to address (*C2*) OOD detection. The core idea is to infer a class embedding for the novel (OOD) classes via logical disjunction and negation and compute the vacuity score, which indicates the lack of evidence with respect to the novel classes, using the node embeddings and the class embeddings for known classes and novel classes.

What is central to EviNet is the node representations that maintain structural and attributive information and are appropriate for logical reasoning. Classic node representation learning approaches [9, 16] often map a node to a data point in a low-dimensional space that could reconstruct the graph. Though widely used, these approaches cannot effectively represent support regions of classes and reason under uncertainty. To address these limitations, in EviNet, we adopt Beta embedding [24] that represents each node as a set of Beta distributions and conceptualizes a node as a set containing itself only. A prominent advantage of Beta embedding is its closure property under various logical operations, such as disjunction and negation, which are two core logical operations to reason for both known and novel classes.

3.2 M1. Detecting Misclassification via Dissonance Reasoning

To detect misclassification, it is important to understand to what extent the model is uncertain in its predictions. EviNet builds upon subjective logic to quantify uncertainty [40], because dissonance in subjective logic reflects the contradicting beliefs about a model prediction. However, the subjective logic-based model may suffer from overconfidence [5]. One possible reason is the lack of consideration for context information between samples and classes in traditional subjective logic-based approaches.

To address this limitation, the key idea of the Dissonance Reasoning module is two-fold: First, we encode nodes as Beta embeddings and reason for the class embeddings using logical disjunction. Second, we generate class-specific graphs to produce context-aware node embeddings, which are further used to estimate dissonance uncertainty.

More specifically, let $f(\mathbf{A}, \mathbf{x}|\theta)$ represent an encoder that outputs a Beta embedding for the node in graph $\mathbf{A}$ with feature $\mathbf{x}$ with θ being the parameters of the encoder. Then a d-dimensional Beta embedding of a given node i can be presented as $\mathcal{N}_i = (\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) = \{(\alpha_{i1}, \beta_{i1}), \ldots, (\alpha_{id}, \beta_{id})\} = \text{Softplus}(f(\mathbf{A}, x_i|\theta))$, where $(\alpha_{ij}, \beta_{ij})$ is the pair of parameters for the j-th Beta distribution, and Softplus

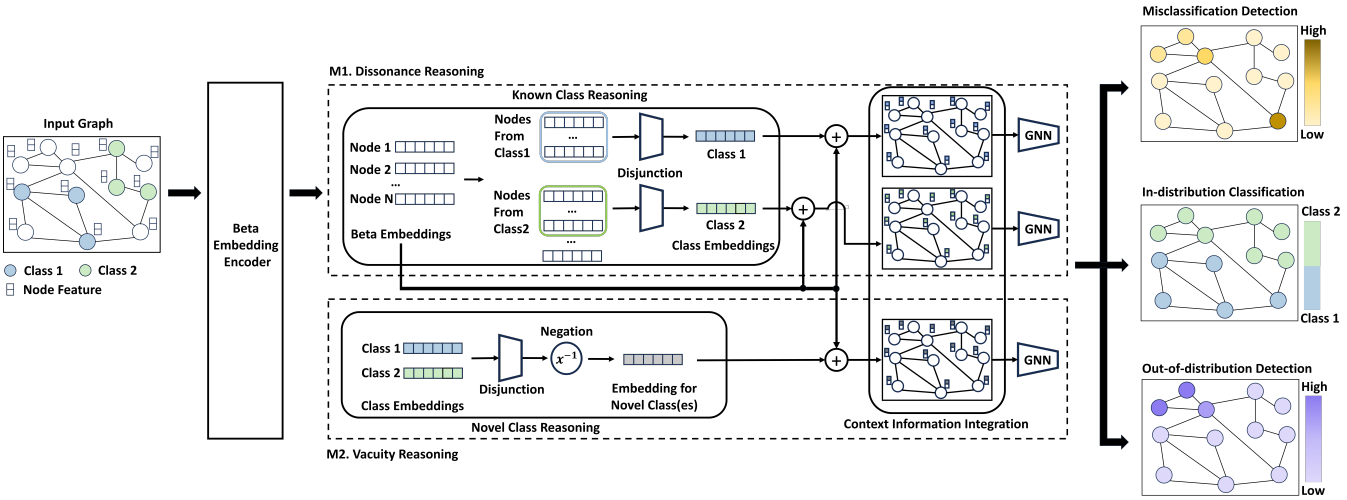


Figure 2: An overview of EvɪNet. The framework processes graph-structured data through two core modules: (1) Dissonance Reasoning (M1); (2) Vacuity Reasoning (M2). The final outputs include in-distribution classification, misclassification detection scores, and OOD detection scores. For simplicity, we consider a binary classification scenario in this example, in which Class 1 is shown in blue, Class 2 is shown in green, and Novel Class(es) is shown in grey.

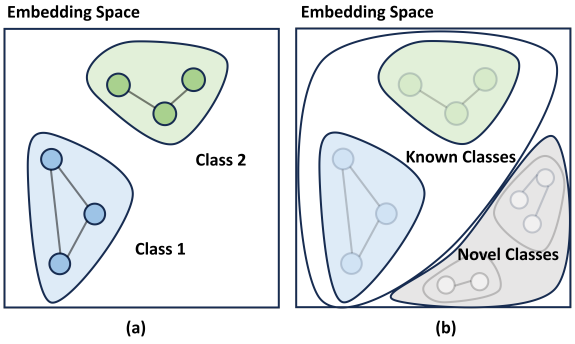


Figure 3: An illustrative figure for M1. Dissonance Reasoning (a) and M2. Vacuity Reasoning (b) with two known classes. In (a), the support regions of class 1 and class 2 are generated by applying disjunction on training samples. In (b), with the support regions of class 1 and class 2, the support region involving all known classes can also be generated with disjunction and by applying negation on it, the inexplicit support region of including novel classes can be derived.

maps each parameter in the Beta embedding to its proper range. Here, we use the graph convolution layer in GCN as our encoder.

Given the node embeddings, we define the disjunction operation to derive class embeddings for known classes. Given a set of n elements $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_n\} = \{(\boldsymbol{\alpha}_1, \boldsymbol{\beta}_1), \ldots, (\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)\}$, in which each element can be either a class or a node in the form of Beta embedding, the disjunction operation $\cup_{i=1}^{n} \mathcal{Z}_i$ is defined as

$$\begin{aligned} &\text{DISJUNCTION}\left(\{(\boldsymbol{\alpha}_1, \boldsymbol{\beta}_1), (\boldsymbol{\alpha}_2, \boldsymbol{\beta}_2), \ldots, (\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)\}\right) \\ &= \text{Softplus}\left(h_2\left(\left(\frac{\sum_{i=1}^{n} h_1([\boldsymbol{\alpha}_i || \boldsymbol{\beta}_i])}{n}\right) \cdot \boldsymbol{w} + bias\right)\right), \end{aligned} \tag{5}$$

where $\boldsymbol{w}$ is the trainable weight vector, and h_1 and h_2 are two projectors, and *bias* is a trainable bias parameter. The reason for applying two projectors h_1 and h_2 is that Beta embedding is naturally a set of Beta distributions with bounded support. Simply aggregating these distributions or their parameters does not align with their statistical properties. Thus, we use two projectors to first project Beta embeddings into a different embedding space where aggregation can be applied and then project the aggregated embedding back to the embedding space of the Beta embedding to obtain the proposition (in the form of Beta embedding) after disjunction.

We then represent each known class as a disjunction of training nodes from this class. Mathematically, given a set of n training nodes $\{\mathcal{N}_{1k}, \ldots, \mathcal{N}_{nk}\}$ from class k with $\mathcal{N}_{ik}$ being the Beta embedding of an arbitrary node i, the class embedding for class k is defined as

$$C_k = \text{DISJUNCTION}\left(\{\mathcal{N}_{1k}, \ldots, \mathcal{N}_{nk}\}\right). \tag{6}$$

Figure 3(a) presents an illustrative example of the class embeddings reasoned from the disjunction operation, i.e., the blue and green shaded areas represent class embeddings of two known classes (blue and green). Then, we integrate context information to avoid overconfidence by using a class-specific graph convolutional network (GCN) to compute the evidence that supports node i belonging to class k. Mathematically, for any known class $k \in 1, \ldots, K$ from a set of K known classes, the evidence of node i belonging to class k is computed as

$$e_{ik} = \left[GCN_k\left(\mathbf{A}, \mathbf{X}'_k\right)\right]_i, \tag{7}$$

where $[\cdot]_i$ denotes the i-th element of the output vector, GCN_k is a class-specific GCN for class k, and $\mathbf{X}_k$ is a specially constructed feature matrix that concatenates the node embeddings $\{\mathcal{N}_{1k}, \ldots, \mathcal{N}_{nk}\}$ with the class embeddings C_k of class k. The context information integration allows each node to generate a context-aware embedding for every class and uses it to compute the dissonance score of

the node belonging to a specific class. Specifically, we compute the dissonance score for node i as

$$\text{Dissonance}_i = \sum_{k=1}^{K} \left(\frac{b_{ik} \sum_{j \neq k} b_{ij} \, \text{Bal}\left(b_{ij}, b_{ik}\right)}{\sum_{j \neq k} b_{ij}} \right), \tag{8}$$

where $b_{ik} = e_{ik}/S_i$, $S_i = \sum_{k=1}^{K} e_{ik} + W_i$, and $\text{Bal}\left(b_k, b_j\right) = 1 - \left|b_k - b_j\right| / \left(b_k + b_j\right)$ is the relative mass balance between a pair of masses b_k and b_j. A key difference between the subjective logic in EviNet and the traditional subjective logic approaches is that W_i is a trainable prior weight that accounts for potentially novel classes (Eq. (12)). We defer detailed computation of W_i to Section 3.3.

3.3 M2. Characterizing Novel Classes via Vacuity Reasoning

Traditional OOD detection techniques typically do not model the representations of novel OOD class(es) explicitly. It might be suboptimal when the boundary between in-distribution data and out-of-distribution data is not clearly defined in high-dimensional space. In contrast, our proposed EviNet introduces an explicit support region [2] for known classes and an implicit support region for novel classes through logical reasoning, which could yield better capability for OOD detection. Specifically, to represent the support regions for known and novel classes, we define the support region of all known classes as the disjunction of class embeddings obtained via Dissonance Reasoning

$$C_{\text{Known}} = \cup_{k=1}^{K} C_k = \textbf{DISJUNCTION}(\{C_1, \ldots, C_K\}). \tag{9}$$

Then the support region of novel class(es) C_{Nov} is defined as the logical negation of the support region of all known classes

$$C_{\text{Nov}} = \overline{\cup_{k=1}^{K} C_k}, \tag{10}$$

where the negation operation is defined as

$$\textbf{NEGATION}(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) = (\frac{1}{\boldsymbol{\alpha}_i}, \frac{1}{\boldsymbol{\beta}_i}). \tag{11}$$

The intuition is that if a node belongs to a novel class, it certainly should not belong to any of the known classes. As shown in Figure 3(b), this approach offers an implicit support region of the novel class(es).

Similar to the context information integration in (M1) Dissonance Reasoning, we compute a context-aware prior weight W_i for any node i in the graph

$$W_i = [GCN_{\text{Nov}}(\mathbf{A}, \mathbf{X}_{\text{Nov}})]_i, \tag{12}$$

where $[\cdot]_i$ denotes the i-th element of the output vector, and $\mathbf{X}_{\text{Nov}}$ is a specially constructed feature matrix that concatenates the node embeddings with the class embedding of the novel class C_{Nov}. Then we compute the vacuity score for node i, which indicates how likely the node i is to belong to novel class(es), as

$$\text{Vacuity}_i = \frac{W_i}{S_i}, \tag{13}$$

where $S_i = \sum_{k=1}^{K} e_{ik} + W_i$, and e_{ik} is the evidence that supports node i belonging to class k (Eq.(7)).

Remark: Traditional subjective logic frameworks preset the prior weight W to the number of classes, which could lead to overconfidence [5]. For example, high confidence could be assigned to misclassified samples, and the model could make high-confidence predictions that classify out-of-distribution data to in-distribution known classes [5]. The main problem is that the fixed prior weight disallows the model to adjust its uncertainty based on the observed evidence, which causes inaccurate uncertainty estimates. EviNet overcomes this limitation by treating the prior weight W_i for each node as a learnable parameter for better capabilities in both misclassification detection tasks and OOD detection.

Besides the dissonance score and vacuity score, we also compute the probability that node i belongs to class k for in-distribution classification as

$$p_{ik} = b_{ik} + a_k \frac{W_i}{S_i}, \tag{14}$$

where $a_k = 1/K$ is the base rate for any class k in the set of K known classes following the common practice.

3.4 Optimization

Our training aims to ensure the effective learning of robust Beta embeddings for nodes, the disjunction operation for logical reasoning, and the estimation of class-specific evidence based on this reasoning, ultimately providing a reliable estimation of uncertainty. The whole training is separated into two phases. In the first phase, we focus on ensuring training the encoder for encoding node into Beta embedding and disjunction operation using the idea that the Beta embedding of a node aligns closely with its corresponding class embedding while being distinct from embeddings of other classes. We achieve this by minimizing the KL Divergence between a node's Beta embedding $\mathcal{N}$ and its corresponding class-level Beta embedding C and maximizing the KL Divergence between other class-level Beta embeddings. Since the encoder is responsible for generating node embeddings and the disjunction operation derives class embeddings from these node embeddings, this method can effectively train both the encoder and the disjunction operation. This loss function BL in this phase for node i with label y_i can be formalized as:

$$\begin{aligned} BL_i = &- \log \sigma(\gamma - \text{Dist}(\mathcal{N}_i; C_{y_i})) \\ &- \sum_{\substack{k=1 \\ k \neq y_i}}^{K} \frac{1}{K} \log \sigma\left(\text{Dist}\left(\mathcal{N}_i; C_k\right) - \gamma\right), \end{aligned} \tag{15}$$

where $\mathcal{N}_i$ is the Beta embedding for node i belongs to the class y_i with class-level Beta embedding C_{y_i}, γ denotes the margin, and Dist is defined as

$$\text{Dist}(\mathcal{N}; C) = \sum_{j=1}^{d} \text{KL}\left(\left(\alpha_{\mathcal{N}j}, \beta_{\mathcal{N}j}\right); \left(\alpha_{Cj}, \beta_{Cj}\right)\right). \tag{16}$$

Here $\left(\alpha_{\mathcal{N}j}, \beta_{\mathcal{N}j}\right)$ or $\left(\alpha_{Cj}, \beta_{Cj}\right)$ represents the j-th beta distribution and d is the dimension of a Beta embedding. We use $\text{Dist}(\mathcal{N}; C)$ rather than $\text{Dist}(C; \mathcal{N})$ to ensure that class embeddings effectively "cover" the modes of all node embeddings belonging to that class. By including the embeddings of other classes (including novel classes) as negative samples, we optimize the model to distinguish the target class from others.

The second phase of the training focuses on training the evidence estimator to estimate the evidence supports that a given node

belongs to a specific class for both prediction and uncertainty estimation. To achieve that, we incorporate an expected cross-entropy loss based on the Dirichlet distribution $D(p_{ik}|\xi_{ik})$ as the loss function where p_{ik} is the predicted probability indicating that sample i belongs to class k described in Eq. 14 and ξ_{ik} is the total strength of the belief of class k for sample i. For a given sample i, the loss function DL in phase 2 is expressed as

$$\begin{aligned} DL_i &= \int \left[\sum_{k=1}^{K} -y_{ik} \log\left(p_{ik}\right) \right] \frac{1}{B\left(\xi_i\right)} \prod_{k=1}^{K} p_{ik}^{\xi_{ik}-1} d\mathbf{p}_i \\ &= \sum_{k=1}^{K} y_{ik} \left(\psi\left(S_i\right) - \psi\left(\xi_{ik}\right)\right), \end{aligned} \tag{17}$$

where ψ is the digamma function, $S_i = \sum_{k=1}^{K} e_{ik} + W_i$ and $\xi_{ik} = e_{ik} + a_k W_i$. a_k is the base rate for class k defined in subjective logic and set as $\frac{1}{K}$ for every class following the common practice.

Unlike traditional subjective-logic-based frameworks that often include a variance-minimization regularization term, which can lead to overconfidence in both misclassification and OOD detection (as shown in [5]), we avoid this in our approach.

To enhance the model's performance further, we adopt an alternating training strategy. This approach iteratively refines both node Beta embeddings and the disjunction operation while improving the evidence estimator and uncertainty estimation. By alternating between these two phases, the model gains a more comprehensive understanding of the data, leading to improved performance.

4 Experiments

In this section, we evaluate the performance of EviNet on five benchmark datasets to understand the performance of EviNet in in-distribution classification, misclassification detection, and out-of-distribution detection. EviNet exhibits superior performance compared to various state-of-the-art baselines (Section 4.2). Moreover, we conduct ablation studies (Section 4.3) to demonstrate the necessity of each module in EviNet and report the parameter sensitivity, scalability and complexity in Section 4.4, Appendix E and Appendix F to show that EviNet is scalable and achieves a convincing performance with minimal tuning efforts.

4.1 Experimental Settings

Datasets. We consider node classification and evaluate EviNet on five benchmark datasets, including Wiki-CS [22], Amazon-Photo, Amazon-Computer, Coauthor-CS, and Coauthor-Physics [25]. For in-distribution data, we follow the common practice [17] and use random splits with 1:1:8 for training/validation/testing. To create an OOD set for out-of-distribution detection, we follow a label leave-out strategy by considering nodes with certain classes as in-distribution and nodes with other labels as out-of-distribution [21]. Detailed statistics of datasets are summarized in Appendix B.

Baseline Methods. We compare EviNet with nine baseline methods in three categories, which include methods that are designed for OOD detection specifically, to misclassification specifically, and to both OOD detection and misclassification detection.

- **Methods for OOD detection** include (1) MaxLogit [12], which uses the maximum logit as the OOD score, (2) ODIN [20], which improves OOD detection with small perturbations to the input data and temperature scaling, (3) Mahalanobis [18], which uses a confidence score based on the Mahalanobis distance for OOD detection, (4) Energy [21] which detects OOD data by an energy-based discriminator and (5) GNNSafe [36], which is a graph-specific method and also detects OOD data by an energy-based discriminator. To evaluate the performance of misclassification detection of these methods, we use their OOD score as the misclassification score for misclassification detection.
- **Methods for misclassification detection** include (1) CRL [23], which improves confidence estimates by optimizing the correctness ranking loss, and (2) Doctor [8], which uses the probabilistic output by the model to determine whether we should trust model prediction or not. To evaluate the performance of OOD detection of these methods, we use their misclassification score as the OOD score for OOD detection.
- **Methods for both OOD detection and misclassification detection** include (1) GPN [27], which estimates the uncertainty of model predictions with Bayesian posterior updates, and (2) GKDE [40], which estimates node uncertainty by leveraging a Dirichlet prior based on graph-based kernel distribution estimation. Following existing settings [27, 40], we use aleatoric uncertainty estimate for misclassification detection and epistemic uncertainty estimate for OOD detection.

Evaluation Metrics. We evaluate the performance of EviNet for in-distribution classification, misclassification detection, and OOD detection. For in-distribution classification, we calculate the classification accuracy (Acc) of in-distribution test samples. For misclassification detection, we calculate the area under the risk coverage curve (AURC), which measures the error rate using samples whose confidence is higher than a pre-defined confidence threshold [7]. For OOD detection, we calculate the false positive rate at 95% true positive rate (FPR95), which quantifies the probability that an OOD example is predicted as an in-distribution when the true positive rate is 95%, and the area under the receiver operating characteristic curve (AUROC). We perform 5 runs and report the mean and standard deviation of each metric in our experiments.

Implementation Details. We set the encoder in EviNet to be two GCN layers, each of which has 64 hidden channels and is followed by a batch normalization layer and a softplus function. For a fair comparison, we apply the same encoder to MaxLogit, ODIN, CRL, Doctor, Mahalanobis, Energy, and GNNSafe. The main reasons are twofold: (1) MaxLogit, ODIN, CRL, Doctor, Mahalanobis, and Energy are not specifically designed for graph data, and (2) GNNSafe is a generic framework that can be applied to any encoder. For GPN and GKDE, we follow their publicly released implementations and use hyperparameters reported in the original paper. Regarding model selection, we select the model with the best overall performance on the validation set, where the overall performance is calculated by Acc + AUROC − 10 ∗ AURC. More details on hyperparameter settings for the learning rate, dropout rate, and γ are deferred to Appendix C.

4.2 Effectiveness Results

Table 1 presents the effectiveness of EviNet in the tasks of in-distribution classification, misclassification detection, and out-of-distribution. In general, we have the following observations: **(1)**

Table 1: Effectiveness of in-distribution classification (IDC), misclassification detection (MD), and out-of-distribution detection (OODD) on Amazon-photo, Amazon-computer, Coauthor-cs, Coauthor-physics, and Wiki-cs. We report the mean and standard deviation for each method on each dataset over five runs. AURC is multiplied by 1000, all other metrics are in percentage. Lower is better for ↓, and higher is better for ↑. Bold and red indicate best, and underlined and green indicate second best.

Dataset	Metric		ODIN	MaxLogit	Mahalanobis	Energy	GNNSafe	CRL	DOCTOR	GKDE	GPN	EviNet
Amazon -photo	IDC	Acc (↑)	92.59 ± 1.92	94.75 ± 0.32	90.07 ± 4.15	94.06 ± 0.91	94.35 ± 0.43	93.18 ± 0.78	**94.90 ± 0.36**	94.27 ± 1.18	91.21 ± 2.98	94.76 ± 0.24
	MD	AURC (↓)	25.48 ± 6.79	14.03 ± 1.20	129.50 ± 46.33	16.56 ± 2.13	10.78 ± 0.85	14.70 ± 2.53	13.64 ± 2.35	12.76 ± 3.41	20.97 ± 10.74	**6.99 ± 0.36**
	OODD	FPR95 (↓)	42.84 ± 23.18	7.69 ± 2.36	69.87 ± 8.42	6.53 ± 2.20	**4.43 ± 2.91**	15.81 ± 6.03	13.81 ± 4.78	37.45 ± 25.45	53.99 ± 31.03	4.79 ± 2.08
		AUROC (↑)	91.44 ± 4.82	98.04 ± 0.43	70.45 ± 4.83	98.20 ± 0.47	97.88 ± 0.31	97.07 ± 0.67	97.22 ± 0.66	91.74 ± 4.76	85.96 ± 8.31	**98.50 ± 0.34**
Amazon -computer	IDC	Acc (↑)	74.18 ± 19.20	86.79 ± 1.74	76.62 ± 10.61	86.05 ± 2.11	87.55 ± 1.38	84.73 ± 0.72	86.84 ± 1.15	83.58 ± 4.75	78.20 ± 14.49	**87.73 ± 0.20**
	MD	AURC (↓)	183.05 ± 202.77	61.46 ± 5.07	272.28 ± 48.16	66.51 ± 7.99	58.31 ± 4.95	50.65 ± 2.03	52.04 ± 6.71	73.25 ± 42.03	80.97 ± 42.42	**31.42 ± 1.88**
	OODD	FPR95 (↓)	54.43 ± 21.23	37.04 ± 10.95	84.04 ± 4.56	35.91 ± 10.24	34.17 ± 7.60	40.84 ± 4.23	38.98 ± 7.60	70.01 ± 16.37	74.99 ± 19.39	**22.01 ± 3.27**
		AUROC (↑)	81.58 ± 15.67	90.99 ± 2.87	65.63 ± 2.58	91.18 ± 1.43	91.20 ± 2.48	90.24 ± 1.31	90.17 ± 2.45	79.61 ± 7.59	79.94 ± 11.73	**95.01 ± 0.76**
Coauthor -cs	IDC	Acc (↑)	91.87 ± 2.20	94.31 ± 0.43	91.07 ± 3.08	93.69 ± 1.20	94.31 ± 0.38	92.00 ± 1.52	93.98 ± 0.69	**95.00 ± 0.15**	85.57 ± 6.92	93.88 ± 0.14
	MD	AURC (↓)	18.30 ± 6.29	15.56 ± 1.80	104.60 ± 34.84	17.02 ± 2.23	19.25 ± 1.11	16.47 ± 3.50	11.65 ± 4.20	16.71 ± 0.50	35.30 ± 29.28	**8.53 ± 0.17**
	OODD	FPR95 (↓)	31.55 ± 7.26	19.00 ± 4.56	88.53 ± 4.15	17.98 ± 2.67	12.40 ± 2.01	39.38 ± 12.90	27.83 ± 8.01	14.76 ± 0.57	65.15 ± 20.10	**5.82 ± 0.47**
		AUROC (↑)	93.21 ± 1.20	95.37 ± 1.17	59.23 ± 6.13	95.77 ± 0.52	96.96 ± 0.33	91.79 ± 2.76	94.02 ± 0.77	96.80 ± 0.25	84.39 ± 5.97	**98.53 ± 0.10**
Coauthor -physics	IDC	Acc (↑)	96.27 ± 0.50	97.59 ± 0.18	94.90 ± 2.13	97.71 ± 0.12	97.69 ± 0.18	96.25 ± 0.73	97.56 ± 0.10	**97.94 ± 0.08**	87.57 ± 12.92	97.62 ± 0.03
	MD	AURC (↓)	5.80 ± 1.77	2.97 ± 0.54	66.98 ± 25.08	2.54 ± 0.57	3.86 ± 0.73	4.06 ± 0.81	2.48 ± 0.47	2.11 ± 0.38	19.10 ± 20.30	**1.66 ± 0.13**
	OODD	FPR95 (↓)	18.08 ± 2.17	8.56 ± 1.36	93.30 ± 7.33	8.31 ± 1.23	7.49 ± 1.06	18.59 ± 2.29	11.24 ± 1.43	8.51 ± 0.52	33.70 ± 25.72	**3.30 ± 0.32**
		AUROC (↑)	95.99 ± 0.43	97.40 ± 0.35	49.55 ± 9.83	97.49 ± 0.44	97.78 ± 0.28	95.88 ± 0.45	96.99 ± 0.47	98.04 ± 0.13	93.37 ± 4.31	**99.18 ± 0.11**
Wiki-cs	IDC	Acc (↑)	68.38 ± 6.75	82.25 ± 0.59	76.91 ± 2.76	82.63 ± 0.66	**82.72 ± 1.35**	82.00 ± 0.35	82.58 ± 0.27	82.00 ± 1.30	79.84 ± 3.55	82.32 ± 1.01
	MD	AURC (↓)	184.52 ± 105.42	62.96 ± 4.33	227.81 ± 28.97	77.84 ± 7.60	81.00 ± 11.33	58.88 ± 3.70	60.85 ± 3.21	56.93 ± 5.66	60.58 ± 14.72	**49.27 ± 4.33**
	OODD	FPR95 (↓)	84.87 ± 4.73	66.34 ± 2.04	91.28 ± 3.44	63.60 ± 6.37	70.43 ± 2.16	70.24 ± 4.55	66.91 ± 2.61	71.91 ± 2.61	85.23 ± 2.38	**55.20 ± 7.52**
		AUROC (↑)	68.99 ± 9.63	84.15 ± 0.57	54.59 ± 1.43	83.01 ± 3.39	84.73 ± 1.66	83.72 ± 1.11	83.78 ± 0.77	81.77 ± 1.39	68.71 ± 2.49	**86.95 ± 3.51**

In-distribution classification. In terms of in-distribution classification accuracy, EviNet demonstrates consistently competitive performance across all datasets. More specifically, EviNet attains the highest accuracy at 87.73%, slightly surpassing GNNSafe (87.55%) on Amazon-computer; The accuracy of EviNet closely approaches the top-performing baseline method GKDE (97.62% for EviNet vs. 97.94% for GKDE) on Coauthor-physics. **(2) Misclassification detection.** EviNet excels in misclassification detection, consistently achieving the lowest AURC scores across all datasets. For instance, on Amazon-photo, EviNet could achieve a significantly lower AURC (6.99) compared to the second-best method, GNNSafe (10.78). This consistent trend across datasets underscores the ability of EviNet for misclassification detection. **(3) Out-of-distribution detection.** EviNet consistently delivers the state-of-the-art performance in both FPR95 and AUROC. For instance, on Amazon-computer, EviNet significantly outperforms other methods, achieving an FPR95 of 22.01% (vs. 34.17% for GNNSafe, the second-best method) and an AUROC of 95.01% (vs. 91.20% for GNNSafe, the second-best method). This strong performance highlights the ability of EviNet to detect out-of-distribution data effectively. **(4) Overall balanced performance.** While EviNet might not always be the top performer in specific tasks on every dataset, its ability to strike an effective balance across tasks leads to consistently strong overall results. For instance, even though the accuracy of EviNet on Amazon-photo is 0.15% lower than the accuracy of DOCTOR (the best-performing method) , it more than compensates for this with a 48.75% lower AURC and a 1.32% higher AUROC. These results demonstrate that EviNet effectively balances the trade-offs between accuracy, misclassification detection, and OOD detection, leading to strong overall performance across multiple tasks. Appendix D presents additional experimental results to further support our findings.

Table 2: Ablation study of EviNet on Amazon-photo by including (✓) or excluding (M1) the dissonance reasoning module, (M2) the vacuity reasoning module, and (CI) context information integration in the tasks of in-distribution classification (IDC), misclassification detection (MD) and out-of-distribution detection (OODD). We report the mean and standard deviation of each variant over five runs. AURC is multiplied by 1000, all other metrics are in percentage. Lower is better for ↓ and higher is better for ↑. Bold and red indicate best.

Variant	Components			IDC	MD	OODD	
	M1	M2	CI	Acc (↑)	AURC (↓)	FPR95 (↓)	AUROC (↑)
(a)				93.07 ± 0.40	11.56 ± 1.40	44.19 ± 12.60	91.12 ± 2.11
(b)	✓			94.55 ± 0.53	8.50 ± 1.40	49.96 ± 11.88	90.90 ± 1.91
(c)	✓	✓		94.33 ± 0.33	9.10 ± 1.05	19.92 ± 6.65	95.59 ± 0.86
(d)	✓		✓	94.43 ± 0.25	7.07 ± 0.32	8.03 ± 2.33	97.35 ± 0.49
(e)	✓	✓	✓	**94.76 ± 0.24**	**6.99 ± 0.36**	**4.79 ± 2.08**	**98.50 ± 0.34**

4.3 Ablation Study

Table 2 presents five variants of EviNet and their performance on Amazon-photo, including (a) only subjective logic uncertainty estimation with a GCN generating evidence directly, (b) EviNet without Vacuity Reasoning (M2) and context information integration (CI), using MLP for evidence calculation and fixed prior weight W, (c) EviNet without CI, using MLP instead of GCN for evidence calculation, (d) EviNet without M2, using fixed prior weight W for

Table 3: Ablation study of EviNet with (w/) or without (w/o) alternate training (AT) on Amazon-photo in the tasks of in-distribution classification (IDC), misclassification detection (MD), and out-of-distribution detection (OODD). We report the mean and standard deviation of each variant over five runs. AURC is multiplied by 1000, all other metrics are in percentage. Lower is better for ↓, and higher is better for ↑. Bold and red indicate best.

Method	IDC	MD	OODD	
	Acc (↑)	AURC (↓)	FPR95 (↓)	AUROC (↑)
w/o AT	94.50 ± 0.10	**6.90 ± 0.33**	5.64 ± 2.26	98.29 ± 0.33
w/ AT	**94.76 ± 0.24**	6.99 ± 0.36	**4.79 ± 2.08**	**98.50 ± 0.34**

each sample, (e) complete EviNet. From the results, we have several interesting observations: (1) Significant performance improvements are evident in configuration (b) compared with (a), with a 1.59% boost in accuracy, a 26.47% reduction in AURC, and a 0.22% increase in AUROC. Although FPR95 increases slightly by 13.06%, the gains in overall performance demonstrate the effectiveness of Beta embedding in the Dissonance Reasoning module (M1) for generating representations that capture misclassified data; (2) The addition of Vacuity Reasoning (M2) in (c) results in minor decreases in accuracy (0.23%) and a 7.06% increase in AURC compared with (b), yet it leads to substantial improvements in OOD detection. FPR95 decreases by 60.13%, and AUROC improves by 5.16%, indicating that Vacuity Reasoning enhances the performance of OOD detection, despite slight trade-offs in in-distribution classification; (3) The inclusion of context information integration (CI) in (d) yields notable benefits, with accuracy improving by 0.12%, AURC reducing by 16.82%, FPR95 dropping by 83.93%, and AUROC increasing by 7.10% compared with (b). These metrics show that integrating context information significantly enhances the performance in both misclassification and OOD detection; (4) The complete EviNet model demonstrates the synergistic effects of integrating all modules. Compared to the previous best configuration (d), the full model increases accuracy by 0.35%, reduces AURC by 1.13%, lowers FPR95 by 40.35%, and increases AUROC by 1.18%. These results affirm that combining all components leads to optimal performance across in-distribution classification, misclassification detection, and OOD detection tasks.

Moreover, we experiment with evaluating the impact of using the alternate training strategy on the Amazon-photo dataset, as shown in Table 3. With alternate training, we observe improvements in accuracy for in-distribution classification, and notably better performance in out-of-distribution detection as indicated by the lower FPR95 and higher AUROC values. While there is a slight increase in AURC for misclassification detection, the overall benefits in other metrics, especially for OOD detection, suggest that the alternate training strategy contributes to a more robust and balanced model performance when considering both misclassification detection and OOD detection.

4.4 Parameter Sensitivity Analysis

Since the only novel hyperparameter introduced in our model is the margin value γ mentioned in Eq. (15), we study its sensitivity in this section. The analysis results are presented in Figure 4. From the figure, we observe that EviNet exhibits high stability in terms of

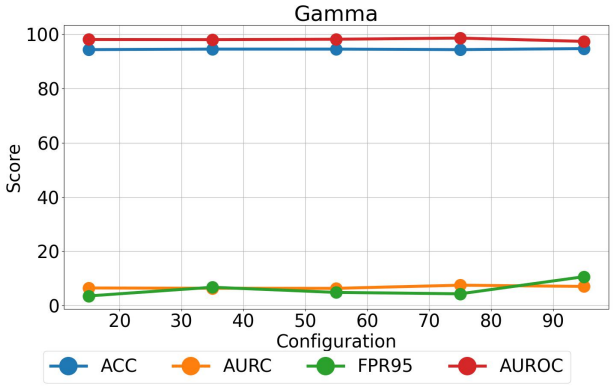


Figure 4: Hyperparameter analysis for γ on Amazon-photo

accuracy (Acc), AURC, FPR95, and AUROC across different values of γ, demonstrating that EviNet is generally robust to γ.

Remark: We also present the analysis of the relationship between the running time and the number of nodes and edges, and the computational overhead and memory usage in Appendix E due to space limitation.

5 Related Work

Out-of-distribution (OOD) detection aims to distinguish between in-distribution samples from known classes and OOD samples from unknown/novel classes, which is critical to several real-world applications. In cybersecurity, it plays a key role in detecting unfamiliar cyber threats [3, 19, 26], while in healthcare, it helps distinguish between common and rare diseases, improving diagnostic accuracy [4, 38]. OOD detection also has notable applications in signal processing, aiding in differentiating known and unknown signal patterns [1, 10, 13, 39]. Representative approaches include ODIN [20], which enhances softmax scores, Mahalanobis-based methods [18], utilizing distance metrics, and Energy-based OOD detection [21], leveraging energy functions in neural networks. These methods show strong performance in preventing high-confidence but wrong predictions for OOD samples. Regarding OOD detection on graphs, techniques like GPN [27] apply Graph Neural Networks (GNNs) to estimate uncertainty to detect OOD data. GNNSafe [36] detects OOD nodes by using an OOD discriminator based on an energy function defined with the logits of graph neural networks.

Misclassification detection, in contrast to OOD detection, primarily addresses the problem of predicting in-distribution samples to a wrong known classes, focusing especially on instances at the boundaries of class categories or those involving noise. [28] presents two computationally efficient methods to estimate uncertainty in Transformer-based models, addressing misclassification detection in named entity recognition and text classification tasks, which are comparable or even better performance to more computationally intensive methods. DOCTOR [8] verifies the predictions made by deep neural networks with consideration of varying level of access to the model to achieve state-of-the-art performance in misclassification without requiring knowledge of the dataset.

Though both misclassification detection and OOD detection are actively studied, there is limited attention to addressing both tasks

on graphs. To our best knowledge, only GPN [27] and GKDE [40] evaluate the performance in both tasks. Our work differs from GPN in that we quantify node uncertainty using Beta embedding with subjective logic, instead of Bayesian uncertainty estimation. Compared to GKDE, we avoid the overconfidence caused by the fixed prior weight and variance-minimizing regularization by introducing a learnable prior weight and the alternate training strategy.

6 Conclusion

In this paper, we introduce the Evidential Reasoning Network (EviNet) to address misclassification and out-of-distribution detection on graphs. EviNet learns node embeddings and class embeddings using Beta embedding to achieve better reasoning ability and to model the support regions of known classes and novel classes. It further applies the subjective logic framework to detect misclassified data by the Dissonance Reasoning module and out-of-distribution data by the Vacuity Reasoning module. Extensive experiments on five widely used benchmark datasets demonstrate that EviNet outperforms state-of-the-art baseline methods in in-distribution classification, misclassification detection, and out-of-distribution detection. This paper highlights the necessity of reasoning and uncertainty quantification on graphs and presents a step forward in graph learning in open and noisy environments.

7 Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work is supported by the National Science Foundation under Award No. IIS-2339989 and No. 2406439, DARPA under contract No. HR00112490370 and No. HR001124S0013, U.S. Department of Homeland Security under Grant Award No. 17STCIN00001-08-00, Amazon-Virginia Tech Initiative for Efficient and Robust Machine Learning, Amazon AWS, Google, Cisco, 4-VA, Commonwealth Cyber Initiative, National Surface Transportation Safety Center for Excellence, and Virginia Tech. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

[1] Wasseem N. Ibrahem Al-Obaydy and Shahrel Azmin Suandi. 2020. *Open-Set Face Recognition in Video Surveillance: A Survey.* 425–436. doi:10.1007/978-981-15-2317-5_36

[2] Mariette Awad and Rahul Khanna. 2015. *Support Vector Machines for Classification.* Apress, Berkeley, CA, 39–66. doi:10.1007/978-1-4302-5990-9_3

[3] Gaspard Baye, Priscila Silva, Alexandre Broggi, Lance Fiondella, Nathaniel D. Bastian, and Gokhan Kul. 2023. Performance Analysis of Deep-Learning Based Open Set Recognition Algorithms for Network Intrusion Detection Systems. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium.* 1–6. doi:10.1109/NOMS56928.2023.10154410

[4] Alexander Cao, Diego Klabjan, and Yuan Luo. 2023. Open-set recognition of breast cancer treatments. *Artificial Intelligence in Medicine* 135 (2023), 102451. doi:10.1016/j.artmed.2022.102451

[5] Mengyuan Chen, Junyu Gao, and Changsheng Xu. 2024. R-EDL: Relaxing Nonessential Settings of Evidential Deep Learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024.* OpenReview.net. https://openreview.net/forum?id=Si3YFA641c

[6] P ERDdS and A R&wi. 1959. On random graphs I. *Publ. math. debrecen* 6, 290-297 (1959), 18.

[7] Yonatan Geifman and Ran El-Yaniv. 2017. Selective Classification for Deep Neural Networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4878–4887. https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html

[8] Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. 2021. DOCTOR: A Simple Method for Detecting Misclassification Errors. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 5669–5681. https://proceedings.neurips.cc/paper_files/paper/2021/file/2cb6b10338a7fc4117a80da24b582060-Paper.pdf

[9] 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. doi:10.1145/2939672.2939754

[10] Manuel Günther, Steve Cruz, Ethan M. Rudd, and Terrance E. Boult. 2017. Toward Open-Set Face Recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017.* IEEE Computer Society, 573–582. doi:10.1109/CVPRW.2017.85

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf

[12] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. 2022. Scaling Out-of-Distribution Detection for Real-World Settings. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 8759–8773. https://proceedings.mlr.press/v162/hendrycks22a.html

[13] Hitham Jleed and Martin Bouchard. 2020. Open Set Audio Recognition for Multi-Class Classification With Rejection. *IEEE Access* 8 (2020), 146523–146534. doi:10.1109/ACCESS.2020.3015227

[14] Audun Josang, Jin-Hee Cho, and Feng Chen. 2018. Uncertainty characteristics of subjective opinions. In *2018 21st International Conference on Information Fusion (FUSION).* IEEE, 1998–2005.

[15] Kazi Zainab Khanam, Gautam Srivastava, and Vijay Mago. 2023. The homophily principle in social network analysis: A survey. *Multimedia Tools and Applications* 82, 6 (2023), 8811–8854.

[16] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. arXiv:1611.07308 [stat.ML] https://arxiv.org/abs/1611.07308

[17] 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. https://openreview.net/forum?id=SJU4ayYgl

[18] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 7167–7177. https://proceedings.neurips.cc/paper/2018/hash/abdeb6f575ac5c6676b747bca8d09cc2-Abstract.html

[19] Xue Li, Jinlong Fei, Jiangtao Xie, Ding Li, Heng Jiang, Ruonan Wang, and Zan Qi. 2023. Open Set Recognition for Malware Traffic via Predictive Uncertainty. *Electronics* 12, 2 (2023). https://www.mdpi.com/2079-9292/12/2/323

[20] Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* OpenReview.net. https://openreview.net/forum?id=H1VGkIxRZ

[21] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. Energy-based Out-of-distribution Detection. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/f5496252609c43eb8a3d147ab9b9c006-Abstract.html

[22] Péter Mernyei and Cătălina Cangea. 2022. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks. arXiv:2007.02901 [cs.LG] https://arxiv.org/abs/2007.02901

[23] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. 2020. Confidence-Aware Learning for Deep Neural Networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119).* PMLR, 7034–7044. http://proceedings.mlr.press/v119/moon20a.html

[24] Hongyu Ren and Jure Leskovec. 2020. Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/e43739bba7cdb577e9e3e4e42447f5a5-Abstract.html

[25] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Pitfalls of Graph Neural Network Evaluation. arXiv:1811.05868 [cs.LG] https://arxiv.org/abs/1811.05868

[26] Gun-Yoon Shin, Dong-Wook Kim, and Myung-Mook Han. 2023. Open Set Recognition With Dissimilarity Weight for Unknown Attack Detection. *IEEE Access* 11 (2023), 102381–102390. doi:10.1109/ACCESS.2023.3277871

[27] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. 2021. Graph Posterior Network: Bayesian Predictive Uncertainty for Node Classification. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 18033–18048. https://proceedings.neurips.cc/paper/2021/hash/95b431e51fc53692913da5263c214162-Abstract.html

[28] Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. 2022. Uncertainty Estimation of Transformer Predictions for Misclassification Detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 8237–8252. doi:10.18653/v1/2022.acl-long.566

[29] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[30] Haohui Wang, Weijie Guan, Jianpeng Chen, Zi Wang, and Dawei Zhou. 2024. Towards Heterogeneous Long-tailed Learning: Benchmarking, Metrics, and Toolbox. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/85cd8edc556709341b2ef6c4d5725545-Abstract-Datasets_and_Benchmarks_Track.html

[31] Haohui Wang, Baoyu Jing, Kaize Ding, Yada Zhu, Wei Cheng, Si Zhang, Yonghui Fan, Liqing Zhang, and Dawei Zhou. 2024. Mastering Long-Tail Complexity on Graphs: Characterization, Learning, and Generalization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, Ricardo Baeza-Yates and Francesco Bonchi (Eds.). ACM, 3045–3056. doi:10.1145/3637528.3671880

[32] Haohui Wang, Yuzhen Mao, Yujun Yan, Yaoqing Yang, Jianhui Sun, Kevin Choi, Balaji Veeramani, Alison Hu, Edward Bowen, Tyler Cody, and Dawei Zhou. 2024. EvoluNet: Advancing Dynamic Non-IID Transfer Learning on Graphs. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. https://openreview.net/forum?id=anM1M5aoM8

[33] Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. 2022. A Review on Graph Neural Network Methods in Financial Applications. arXiv:2111.15367 [q-fin.ST]

[34] Tuo Wang, Jian Kang, Yujun Yan, Adithya Kulkarni, and Dawei Zhou. 2025. Non-exchangeable Conformal Prediction for Temporal Graph Neural Networks. In *Proceedings of the 31st ACM SIGKDD International conference on knowledge discovery & data mining*.

[35] Longfeng Wu, Yao Zhou, Jian Kang, and Dawei Zhou. 2025. Bridging Fairness and Uncertainty: Theoretical Insights and Practical Strategies for Equalized Coverage in GNNs. In *THE WEB CONFERENCE 2025*. https://openreview.net/forum?id=lZQp7s69ls

[36] Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. 2023. Energy-based Out-of-Distribution Detection for Graph Neural Networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=zoz7Ze4STUL

[37] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* 55, 5, Article 97 (dec 2022), 37 pages. doi:10.1145/3535101

[38] Jiawen Xu, Matthias Kovatsch, and Sergio Lucia. 2021. Open Set Recognition for Machinery Fault Diagnosis. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*. 1–7. doi:10.1109/INDIN45523.2021.9557572

[39] Chen Zhao, Dawei Du, Anthony Hoogs, and Christopher Funk. 2023. Open Set Action Recognition via Multi-Label Evidential Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 22982–22991.

[40] Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. 2020. Uncertainty Aware Semi-Supervised Learning on Graph Data. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 12827–12836. https://proceedings.neurips.cc/paper_files/paper/2020/file/968c9b4f09cbb7d7925f38aea3484111-Paper.pdf

A Symbols and Notations

Table 4 presents key symbols and their corresponding notations used in the paper.

Table 4: Key symbols and their descriptions.

Symbol	Description
$\mathcal{V}$	A node set for a graph.
$\mathcal{E}$	An edge set for a graph.
$\mathbf{A}$	The adjacency matrix of a graph.
$\mathbf{X}$	The node feature matrix of a graph.
$\mathcal{Y}$	The label set of known classes of a graph.
K	The number of known classes.
$\mathcal{N}$	The node Beta embeddings.
$\mathcal{C}$	The class Beta embeddings.
γ	The margin for training Beta embedding.
d	The number of Beta distributions.
(α, β)	a pair of parameters for a Beta distribution.
ξ	the parameter of Dirichlet distribution.
b_{ik}	the belief indicates sample i belongs to class k.
e_{ik}	the evidence supports sample i belongs to class k.
W_i	the prior weight for sample i defined by subjective logic

Table 5: Dataset statistics. # Classes (5th column) indicates the number of all classes in the dataset that includes both in-distribution classes and out-of-distribution classes, and # OOD Classes (6th column) indicates the number of OOD classes only.

Dataset	# Nodes	# Edges	# Attributes	# Classes	# OOD Classes
Amazon-photo	7,650	238,162	745	8	4
Amazon-computer	13,752	491,722	767	10	3
Coauthor-cs	18,333	163,788	6,805	15	4
Coauthor-physics	34,493	495,924	8,415	5	2
Wiki-cs	11,701	431,726	300	10	3

B Dataset Statistics

Table 5 summarizes statistics for the datasets, including the number of nodes (# Nodes), the number of edges (# Edges), the number of attributes for each node (# Attributes), the number of all classes (# Classes, including known classes and OOD classes), and the number of OOD classes considered in our experiments (# OOD Classes).

C Hyperparameter Settings

EviNet has two training phases (Section 3.4), and we conduct parameter search for each phase on each dataset. In both phases, we conduct separate grid searches for the learning rate (lr) within 0.01, 0.001, 0.0005 and the dropout rate within 0.2, 0.4, 0.6, with different

Table 6: Effectiveness of in-distribution classification (IDC), misclassification detection (MD), and out-of-distribution detection (OODD) on Amazon-photo using models selected by validation accuracy. We report the mean and standard deviation for each method on each dataset over five runs. AURC is multiplied by 1000, all other metrics are in percentage. Lower is better for ↓, and higher is better for ↑. Bold and red indicate best, and underlined and green indicate second best.

Dataset	Metric		ODIN	MaxLogit	Mahalanobis	Energy	GNNSafe	CRL	DOCTOR	GKDE	GPN	EviNet
Amazon -photo	IDC	Acc (↑)	77.29 ± 16.53	95.18 ± 0.24	86.28 ± 13.52	95.06 ± 0.40	**95.32 ± 0.23**	94.26 ± 0.61	95.20 ± 0.12	94.23 ± 1.66	90.98 ± 3.78	94.74 ± 0.23
	MD	AURC (↓)	101.95 ± 113.91	20.16 ± 4.09	167.11 ± 133.28	18.33 ± 3.12	15.12 ± 7.05	13.41 ± 2.60	14.61 ± 3.71	15.88 ± 2.83	23.81 ± 14.87	**7.77 ± 0.39**
		AUROC (↑)	81.19 ± 3.66	77.30 ± 3.19	41.60 ± 5.50	79.33 ± 3.26	80.33 ± 6.67	87.04 ± 2.15	83.92 ± 2.55	**87.33 ± 2.75**	85.53 ± 2.10	78.96 ± 1.88
		AUPR (↑)	91.93 ± 8.31	98.03 ± 0.41	83.48 ± 13.30	98.22 ± 0.32	98.55 ± 0.73	**98.77 ± 0.29**	98.60 ± 0.39	98.63 ± 0.18	97.95 ± 1.23	98.22 ± 0.16
	OODD	FPR95 (↓)	65.12 ± 19.18	22.29 ± 21.06	77.15 ± 9.08	12.09 ± 4.77	5.97 ± 4.22	37.28 ± 17.72	15.33 ± 1.53	52.02 ± 33.53	55.22 ± 26.59	**4.52 ± 1.25**
		AUROC (↑)	81.38 ± 20.14	94.96 ± 4.64	66.92 ± 3.42	97.03 ± 0.87	97.58 ± 0.59	93.88 ± 2.65	96.80 ± 0.33	86.97 ± 8.70	85.69 ± 6.72	**98.39 ± 0.30**
		AUPR (↑)	82.13 ± 19.99	94.22 ± 4.34	58.84 ± 2.49	96.05 ± 1.09	97.81 ± 0.52	93.80 ± 2.61	95.93 ± 0.54	85.69 ± 6.75	85.72 ± 4.18	**98.25 ± 0.25**

Table 7: Hyper-parameters for Each Dataset in Phase 1 (P1) and Phase 2 (P2)

Dataset	lr (P1)	dropout (P1)	γ	lr (P2)	dropout (P2)
Amazon-photo	0.005	0.2	55	0.001	0.6
Amazon-computer	0.0005	0.2	15	0.001	0.4
Coauthor-cs	0.001	0.2	55	0.01	0.6
Coauthor-physics	0.01	0.2	15	0.01	0.6
Wiki-cs	0.0005	0.2	15	0.001	0.4

optimal values selected for each phase based on the validation set. Additionally, in the first phase, we also perform a grid search for γ within 15, 55, 95, 135. Table 7 lists the parameter settings that achieve the optimal performance on each dataset.

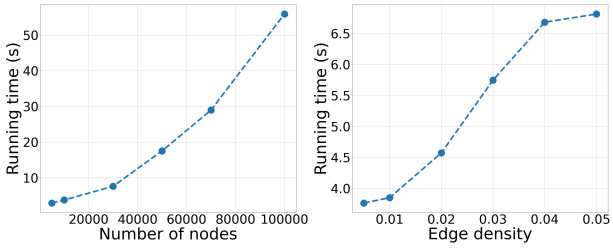
D Additional Experiment Results

Here we present the results with AUROC and AUPR for both misclassification and OOD detection using models selected by validation accuracy on Amazon-photo dataset in Table 6. As we can see, the results align with our main findings and support the robustness of our method. However, we note that AUROC and AUPR can be misleading for misclassification detection, as each model defines positives and negatives differently [5]. These metrics reflect internal ranking quality rather than true prediction reliability. Thus, we use AURC as the primary metric for the misclassification detection task.

E Scalability Analysis

Table 8: Comparison of Training Time, Inference Time, and Memory Usage on ogbn-arxiv

Method	Training Time (s)/epoch	Inference Time (s)/epoch	Memory Usage (MB)
ODIN	0.020	2.66	862
MaxLogit	0.019	2.4	862
GNNSafe	0.032	2.46	1095
CRL	0.020	2.5	863
DOCTOR	0.019	2.5	862
GKDE	21.17	2.4	878
GPN	/	/	/
Ours	5.05	2.5	1400



(a) Running time vs. # of nodes. (b) Running time vs. # of edge density.

Figure 5: Running time vs. the number of nodes and edge density.

Here we analyze the scalability of EviNet, by reporting the running time of EviNet on a series of synthetic graphs with increasing sizes (i.e., the number of nodes and the edge density), we generate the synthetic graphs via the ER algorithm [6]. In Figure 5(a), we fix the edge density to be 0.005 and gradually increase the number of nodes from 5000 to 100000. In Figure 5(b), we fix the number of nodes to 10000 and increase the edge density from 0.005 to 0.05. From Figure 5, we observe that the running time of EviNet scales efficiently with both the number of nodes and edge density. As the number of nodes increases, the running time shows steady growth, with a slight superlinear increase at larger scales. This might be due to the increasing number of logical operations needed for Dissonance Reasoning and Vacuity Reasoning. While regarding the scalability with respect to edge density, even though the edge density increases by 10x, the running time only increases less than 2x, demonstrating the reasonable computational efficiency of EviNet.

What's more, we also evaluated EviNet's scalability on ogbn-arxiv with 169,343 nodes and 1,166,243 edges, which is significantly larger than our main experimental datasets like Amazon-computer (13,752 nodes). The results for EviNet and all baseline models are shown in Table 8. Simple baseline models like ODIN and DOCTOR show minimal overhead with 0.02s training time and 860MB memory usage. EviNet requires higher resources (5.05s training time, 1400MB memory) due to its Beta embedding and context integration components. Notably, while GKDE, which is also a subjective logic-based method, uses less memory but requires much longer training time (21.17s per epoch), GPN was unable to process this

dataset due to out-of-memory issues. While EviNet has higher computational costs than simpler baselines, this trade-off enables its superior performance in both misclassification and OOD detection tasks.

F Computational Complexity Compared with Regular Graph Embedding

We analyze the computational complexity of our proposed model in comparison to standard graph embedding approaches. When the Beta embedding and reasoning modules are removed from EviNet, the remaining architecture reduces to a standard evidential network with a GNN encoder. The complexity of this baseline is given by $O(mH + nH^2 + nKH)$, where m is the number of edges, n is the number of nodes, H is the hidden dimension, and K is the number of classes.

EviNet extends this baseline by incorporating class-aware reasoning through Beta embeddings. This includes class-level disjunction reasoning, context-sensitive class-specific GCNs, and the estimation of dissonance and vacuity. These components contribute an additional complexity of $O(K(mH + nH^2) + nK^2 + KnD)$, where D represents the dimension used in reasoning. The overall complexity thus grows with the number of classes K.

While this introduces additional computational cost, it enables EviNet to explicitly model both dissonance and vacuity uncertainty, resulting in substantially improved performance on misclassification detection and out-of-distribution detection. These capabilities are beyond the reach of standard graph embedding methods or GNNs without explicit reasoning mechanisms.