Evidence-Based Out-of-Distribution Detection on Multi-Label Graphs

Ruomeng Ding* Xujiang Zhao† Chen Zhao‡ Minglai Shao* Zhengzhang Chen†
Haifeng Chen†

Abstract

The Out-of-Distribution (OOD) problem in graph-structured data is becoming increasingly important in various areas of research and applications, including social network recommendation [38], protein function detection[10, 22], *etc.* Furthermore, owing to the inherent multi-label properties of nodes, multi-label OOD detection remains more challenging than in multi-class scenarios. A lack of uncertainty modeling in multi-label classification methods prevents the separation of OOD nodes from in-distribution (ID) nodes. Existing uncertainty-based OOD detection methods on graphs are not applicable for multi-label scenarios because they are designed for multi-class settings. Therefore, node-level OOD detection on multi-label graphs becomes desirable but rarely touched. In this paper, we propose a novel Evidence-Based Out-of-Distribution Detection method on multi-label graphs. The evidence for multiple labels, which indicates the amount of support to suggest that a sample should be classified into a specific class, is predicted by Multi-Label Evidential Graph Neural Networks (ML-EGNNs). The joint belief is designed for multi-label opinions fusion by a comultiplication operator. Additionally, we introduce a Kernel-based Node Positive Evidence Estimation (KNPE) method to reduce errors in quantifying positive evidence. Experimental results prove both the effectiveness and efficiency of our model for multi-label OOD detection on 7 multi-label benchmarks.

1 Introduction

Many real-world application scenarios can be represented by multi-label graphs, including social networks, academic cooperation network, and protein-protein-interaction networks[39, 2, 40]. In multi-label graphs, nodes inherently own multiple labels and only part of the nodes are labeled. Further, some unlabeled nodes can be out-of-distribution (OOD) because their labels didn't appear in labeled nodes. As shown in Fig 1, in a protein-protein-interaction (PPI) network, Function 3, 4, and 5 are unseen for Labeled Protein A, B and C. A multi-class classification method classifies OOD Unlabeled Protein F and G into one or more In-Distribution Functions(like Function 1 and Function 2). The model's inability to detect unknown functions highlights the need to investigate the out-of-distribution (OOD) detection problem in multi-label graphs. A key challenge is that current OOD detection methods often fail to integrate information across multiple labels. By effectively identifying these OOD instances, we can uncover unknown protein functions, which is crucial for advancing our understanding of the human body and developing new medicines.

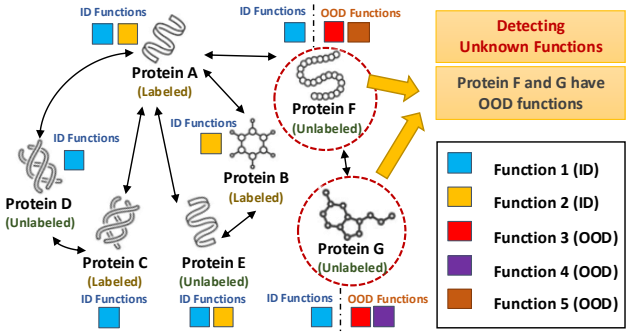


Figure 1: In a Protein-Protein Interaction network, nodes represent proteins, edges connect pairs of interacting proteins, and labels indicate the various functions of these proteins. The network consists of three types of nodes: In-Distribution Labeled Proteins A, B, and C for training; In-Distribution Unlabeled Proteins D and E; and Out-of-Distribution Unlabeled Proteins F and G. During the training process, Functions 3, 4, and 5 remain unseen by the model.

Recently, some semi-supervised learning methods have been proposed for multi-label node classification on graphs[33, 47, 2], with the purpose of predicting user interests in social networks or identifying functions of proteins in PPI networks. However, these methods cannot distinguish OOD nodes from in-distribution (ID) nodes. Due to the lack of uncertainty modeling, they will confidently tag an OOD node only with ID classes

*Tianjin University
†NEC Laboratories America
‡Baylor University

from training data without giving useful estimates of their predictive uncertainty[25]. By effectively distinguishing OOD nodes, we can identify users with potential interests for better recommendation. In addition, drug discovery usually relies on limited labeled data, whereas testing needs to be done on a wider variety of candidates, including some OOD samples[22]. Thus, multi-label out-of-distribution detection is becoming a crucial and inevitable problem for graphs.

Some OOD detection methods[28, 14, 8] based on uncertainty estimation[11, 20, 24] are only available for multi-class graphs. In multi-class setting, each sample only has one label. While, in multi-label setting, each sample may have more than one label. There are some OOD detection methods[21, 17] may be suitable for multi-label OOD settings. However, they may not be effective when dealing with graph data. Besides, there are some evidence-based methods[46, 34] proposed for OOD detection on multi-class graphs with a Dirichlet distribution as conjugate prior[30]. Such methods are not applicable for multi-label graphs. That is because classification probabilities in multi-label setting follow binomial distributions, not a categorical distribution, whose prior is the Beta distribution but not the Dirichlet distribution. Moreover, evaluating metrics including Shannon entropy, the negative log likelihood (NLL), vacuity (*derived from a lack of evidence*) and dissonance (*derived from conflicting evidence*)[12, 16], which are designed for multi-class uncertainty quantification hence not applicable on multi-label graphs. Under multi-label settings, those metrics may incorrectly regard some ID nodes as OOD samples. For instance, in Fig **??**, both Protein D and Protein E have ID unlabeled functions. However, Protein D does not own Function 2 like Protein E. Thus, vacuity designates Protein D as an OOD protein characterized by insufficient information, while dissonance identifies Protein E as an OOD protein exhibiting conflicting evidence.

To address aforementioned problems, we propose a novel evidence based OOD detection method on multi-label graphs. Based on Subjective Logic[17], *Evidence* is the amount of support collected from data to suggest that a sample should (or should not) be classified into a specific class. Under multi-label setting, for each ID class, we define *positive evidence* as a measure of the confidence to classify a sample into this class. While *negative evidence* is used to quantify the objections.

Specifically, we introduce Multi-Label Evidential Graph Neural Networks (ML-EGNNs), from which the positive and negative evidence are used to estimate the predictive uncertainty. Under the Beta prior, ML-EGNNs have a specific loss function, *Beta loss*, which is minimized subject to network parameters using backprop. To address the combination of evidence from multiple classes, we term *joint belief* for multi-label samples based on the comultiplication of binomial opinions[17]. Besides, a Kernel-based Node Positive Evidence Estimation (KNPE) method is provided, using structural information and collecting prior positive evidence from training nodes, to help detect multi-label out-of-distribution nodes. Moreover, to maintain a reliable performance on ID classification, the separate belief of different classes is treated as a basis for class probabilities, which is both effective and efficiency. In summary, the contribution of this paper is three-fold:

- We propose a novel problem of out-of-distribution (OOD) detection on the multi-label graph and develop a novel evidential method for node-level OOD detection. To the best of our knowledge, this is the first study to detect OOD nodes with multiple labels on graphs.
- We introduce Multi-Label Evidential Graph Neural Networks (ML-EGNNs) with Beta loss to predict uncertainty for multiple classes. Besides, we define *joint belief* for multi-label opinions fusion. Additionally, we develop a Kernel-based Node Positive Evidence Estimation (KNPE) method to reduce errors in quantifying positive evidence.
- Experimental results show both the effectiveness and efficiency of our model on multi-label OOD detection.

2 Preliminaries

2.1 Subjective Logic (SL) Subjective logic (SL) is a probabilistic logic that incorporates epistemic uncertainty and source trust [17]. Epistemic uncertainty assesses whether input data falls within the observed distribution [18]. In a multi-class setting, a multinomial opinion of a random variable y is represented as $\omega = (\mathbf{b}, u, \mathbf{a})$ with domain $\mathcal{C} = \{1, \ldots, K\}$ [15, 46], where $\mathbf{b}$ represents belief mass distribution, u indicates uncertainty due to lack of evidence, and $\mathbf{a}$ denotes the base rate distribution. The term *evidence* reflects how much data supports a specific classification [46]. For a K-class setting, the probability mass $\mathbf{p} = [p_1, p_2, \ldots, p_K]$ is assumed to follow a Dirichlet distribution characterized by a K-dimensional Dirichlet strength vector $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$. However, this approach does not apply to multi-label settings, where classifications adhere to multiple binomial distributions. To address this, we introduce the Beta distribution, the conjugate prior of the binomial distribution, which can provide binary evidence for each class:

(2.1)
$$\text{Beta}(p \mid \alpha, \beta) = \begin{cases} \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}, & \text{for } p \in [0,1] \\ 0, & \text{otherwise} \end{cases}$$

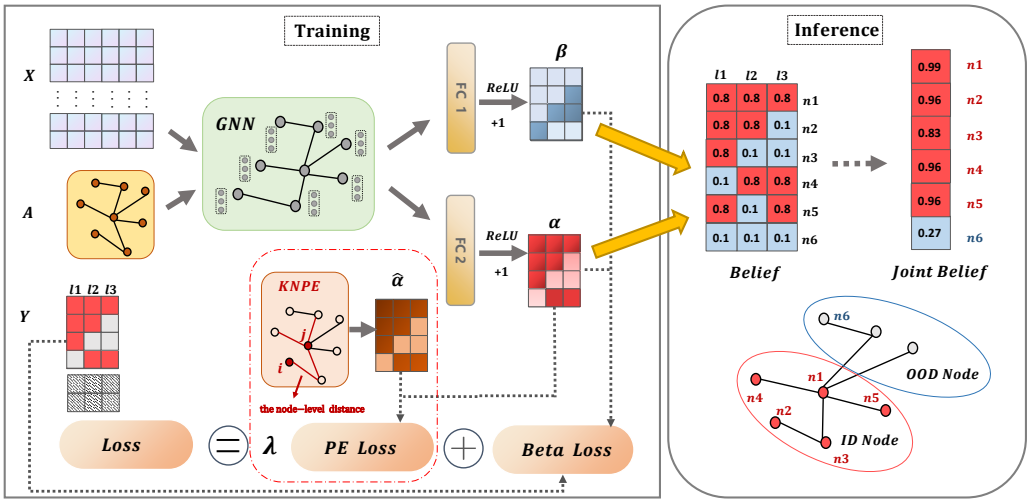


Figure 2: Overall framework of our proposed method ML-EGNNs for training and inference.

where the probability mass $p \in [0,1]$ is assumed to follow a Beta distribution parameterised by a 2 dimensional strength vector $[\alpha, \beta]$. $B(\alpha, \beta)$ is a 2 dimensional Beta function. Each binomial classification ω holds a binomial opinion:

$$\omega = (b, d, u, a) \tag{2.2}$$

with domain $\mathcal{C} = \{0,1\}$, where b indicates belief mass distribution, d indicates disbelief mass distribution, u indicates uncertainty with a lack of evidence, and a indicates base rate distribution. Let $\mathbf{e} = \{e_{pos}, e_{neg}\}$ be the evidence for one binomial classification, where the positive evidence $e_{pos} \geq 0$ and the negative evidence $e_{neg} \geq 0$. The Beta strength $[\alpha, \beta]$ are linked by the following $\alpha = e_{pos} + aW$ and $\beta = e_{neg} + aW$, where W is the weight of uncertain evidence. With loss of generality, the weight W is set to 2 and considering the assumption of the subjective opinion that $a = 1/2$, we have the Beta strength $\alpha = e_{pos} + 1$, $\beta = e_{neg} + 1$. The total strength of the Beta is defined as $S = \alpha + \beta$. Then the Beta evidence can be mapped to the subjective opinion by setting the following equality's:

$$b = \frac{\alpha - 1}{\alpha + \beta}, \quad d = \frac{\beta - 1}{\alpha + \beta}, \quad u = \frac{2}{S} = \frac{2}{\alpha + \beta}. \tag{2.3}$$

2.2 Graph Neural Networks (GNNs) Graph neural networks (GNNs) provide a feasible way to extend deep learning methods into the non-Euclidean domain including graphs and manifolds[42]. For each node, GNN aims to learn an embedding containing information about its neighborhood and itself. The embedding h_i is a vectors of node i[48]:

$$\begin{aligned} h_i &= f(x_i, h_{nei[v]}, x_{nei[v]}), \\ o_i &= g(h_i, x_i), \end{aligned} \tag{2.4}$$

where f represents the local transition function, $h_{nei[v]}$ and $x_{nei[v]}$ are the embeddings and the features of neighbors of node i. Notable models of aggregators include GCN[7, 19], GAT[36], and GraphSAGE[13]. An end-to-end framework can be established by stacking graph convolutional layers, fully connected layers, and an activation function.

3 Methodology

3.1 Notations and Problem Formulation Given a multi-label graph $\mathcal{G} = (\mathbb{V}, \mathbb{E}, \mathbf{A}, \mathbf{X}, \mathbf{Y}_{\mathbb{L}})$ consisting of a set of nodes $\mathbb{V} = \{1, ..., N\}$ and a set of edges $\mathbb{E} \subset \mathbb{V} \times \mathbb{V}$, where the connections in $\mathcal{G}$ can be represented by the adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$. $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, ..., \mathbf{x}_N^T]$ is the node feature matrix. $\mathbf{Y}_{\mathbb{L}} = \{\mathbf{y_i} | i \in \mathbb{L}\}$ are the labels of the training nodes $\mathbb{L} \subset \mathbb{V}$. $\mathbf{y}_i = [0,1]^K$ is the class label of node i, where K is the number of in-distribution classes. Following the semi-supervised learning pattern, among all the nodes, $\mathbb{L}$ are labeled nodes while the remaining $\mathbb{U} = \mathbb{V} \backslash \mathbb{L}$ are unlabeled. $\mathbb{U} = \mathbb{U}_{ID} + \mathbb{U}_{OOD}$, where $\mathbb{U}_{ID}$ denotes unlabeled ID nodes and $\mathbb{U}_{OOD}$ denotes unlabeled OOD nodes. Here we only consider $\mathbb{U}_{OOD}$ as nodes which do not have any labels in K known classes. **We aim to predict**: (1) **class probabilities** of $\mathbb{U}$: $\mathbf{p}_{\mathbb{U}} = \{\mathbf{p}_i \in [0,1]^K | i \in \mathbb{U}\}$ for classifications; (2) **belief estimates**: the joint belief of $\mathbb{U}$, $\mathbf{b}_{\mathbb{U}} = \{\mathbf{b}_i \in [0,1] | i \in \mathbb{U}\}$, where $\mathbf{b}_i$ indicates the confidence in dividing node i into known classes. We provide the detailed notations and their descriptions of this paper in Appendix A.

3.2 Multi-Label Evidential Graph Neural Networks (ML-EGNNs) Further, a multi-label classification opinion $\mathbf{\Omega}$ can be formulated as a combination of K binomial classification opinions $\{\omega_1, ..., \omega_k, ..., \omega_K\}$ [3]. Each binomial classification ω_k holds a binomial opinion $\omega_k = (b_k, d_k, u_k, a_k)$ with domain $\mathcal{C}_k = \{0,1\}$, b_k indicates positive belief mass distribution, d_k indi-

cates negative belief mass distribution, u_k indicates uncertainty with a lack of evidence, and a_k indicates base rate distribution.

Multi-Label Evidence Estimation. Compared with classical neural networks, Evidential Neural Networks (ENNs)[30, 15] do not have a softmax layer, but use an activation layer (e.g., ReLU) to make sure that the output is non-negative. To be specific, as shown in Fig 2, Multi-Label Evidential Graph Neural Networks (ML-EGNNs) are built by stacking graph convolutional layers and two fully connected layers (FCs) and ReLU layers, which are taken as the positive and negative evidence vectors for Beta distribution respectively.

Given sample i, let $f_{pos}(\mathbf{X}, \mathbf{A}|\theta)$ and $f_{neg}(\mathbf{X}, \mathbf{A}|\theta)$ represent the positive and negative evidence vectors predicted by ML-EGNNs, where $\mathbf{X}$ is the input node feature matrix, $\mathbf{A}$ is the adjacency matrix, and θ represents the network parameters. Then, the two parameters $\boldsymbol{\alpha}_i = [\alpha_{i1}, ..., \alpha_{ik}, ..., \alpha_{iK}]$ and $\boldsymbol{\beta}_i = [\beta_{i1}, ..., \beta_{ik}, ..., \beta_{iK}]$ of Beta distribution for node i:

$$\begin{aligned} \boldsymbol{\alpha}_i &= f_{pos}(\mathbf{X}, \mathbf{A}|\theta) + \mathbf{1}, \\ \boldsymbol{\beta}_i &= f_{neg}(\mathbf{X}, \mathbf{A}|\theta) + \mathbf{1}. \end{aligned} \tag{3.5}$$

where k indicates the k-th class of total K classes. For the classification task, the class probabilities are the softmax outputs of $f_{pos}(\mathbf{X}, \mathbf{A}|\theta)$.

Training Loss. With N training samples and K different classes, a multi-label evidential neural network is trained by minimizing the Beta loss:

$$\begin{aligned} \mathcal{L}_{Beta} &= \sum_{i=1}^{N}\sum_{k=1}^{K} \int \left[\mathbf{BCE}\left(y_{ik}, p_{ik}\right)\right] B\left(\alpha_{ik}, \beta_{ik}\right) dp_{ik} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \int \left[-y_{ik}\log\left(p_{ik}\right) - \left(1-y_{ik}\right)\log\left(1-p_{ik}\right)\right] B\left(\alpha_{ik}, \beta_{ik}\right) dp_{ik} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \left[-y_{ik}\mathbb{E}\left[\log\left(p_{ik}\right)\right] - \left(1-y_{ik}\right)\mathbb{E}\left[\log\left(1-p_{ik}\right)\right]\right], \end{aligned} \tag{3.6}$$

where $B(\alpha_{ik}, \beta_{ik})$ is a 2 dimensional Beta function. $\mathbf{BCE}(\cdot)$ denotes the Binary Cross Entropy Loss. p_{ik} represents the predicted probability of sample i belonging to class k by model. y_{ik} represents the ground truth for sample i with label k, *i.e.*, $y_{ik} = 1$ means the training node i belongs to class k, otherwise $y_{ik} = 0$. We use $\mathbb{E}[\cdot]$ to represent $\mathbb{E}_{p_{ik}\sim\mathbf{Beta}}[\cdot]$. To be specific,

$$\mathbb{E}_{p_{ik}\sim\mathbf{Beta}}\left[\log\left(p_{ik}\right)\right] = \psi(\alpha_{ik}) - \psi(\alpha_{ik}+\beta_{ik}), \tag{3.7}$$

$$\mathbb{E}_{p_{ik}\sim\mathbf{Beta}}\left[\log\left(1-p_{ik}\right)\right] = \psi(\beta_{ik}) - \psi(\alpha_{ik}+\beta_{ik}), \tag{3.8}$$

where we use $\Gamma(\cdot)$ represents the Gamma function. The derivation of Eq.(3.7) and Eq.(3.8) can be found in Appendix E. Thus, the Beta loss term $\mathcal{L}_{Beta}$ is:

$$\begin{aligned} \mathcal{L}_{Beta} = \sum_{j=1}^{N}\sum_{i=1}^{K} &\left[y_{ij}\left(\psi\left(\alpha_{ij}+\beta_{ij}\right) - \psi\left(\alpha_{ij}\right)\right)\right. \\ &\left. + \left(1-y_{ij}\right)\left(\psi\left(\alpha_{ij}+\beta_{ij}\right) - \psi\left(\beta_{ij}\right)\right)\right], \end{aligned} \tag{3.9}$$

where $\psi(\cdot)$ denotes the Digamma function. Besides, as the belief and disbelief of label k for sample i, we have:

$$b_{ik} = \frac{\alpha_{ik}-1}{\alpha_{ik}+\beta_{ik}}, \qquad d_{ik} = \frac{\beta_{ik}-1}{\alpha_{ik}+\beta_{ik}}. \tag{3.10}$$

So far, for in-distribution multi-label classification, we set the positive belief as the probability of class i for sample j, *i.e.*, $\frac{\alpha_{ik}-1}{\alpha_{ik}+\beta_{ik}}$, without additional time consuming.

Importance of Multi-Label Positive Evidence. Here, we discuss the connections and differences between multi-class and multi-label OOD detection. In the multi-class OOD scenario, there is some evidence for each class, leading to vacuity uncertainty as defined by Eq. (2.3). In contrast, for multi-label OOD detection, we predict the Beta distribution for each class, where the ideal Beta distribution for an OOD example—belonging to no ID class—will have zero positive evidence and large negative evidence. This results in small vacuity uncertainty, making it challenging to distinguish between ID and OOD samples. Additionally, most ID nodes belong to only a few classes, causing significant negative evidence in other classes. This complicates differentiation based on negative evidence. Unlike ID nodes, OOD nodes have zero positive evidence, which may aid in detecting multi-label OOD samples. Given the importance of positive evidence in multi-label OOD detection, we propose KNPE and multi-label opinion fusion techniques in the following sections to enhance positive evidence estimation during both training and inference.

3.3 Kernel-based Node Positive Evidence Estimation (KNPE)

The idea of the KNPE is to estimate prior Beta distribution parameter for each node based on the labels of other training nodes and node-level distance. To be specific, we focus on the estimation the prior information of multi-label evidence. For each pair of training nodes i and j, calculate the node-level distance d_{ij}, *i.e.*, the shortest path between nodes i and j. Then the Gaussian kernel function is used to estimate the positive distribution effect between nodes i and j:

$$g\left(d_{ij}\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right), \tag{3.11}$$

where σ is the bandwidth parameter. The contribution of positive evidence estimation for node i from labeled

Table 1: Details of 7 benchmark multi-label graph-structured datasets.

Dataset	Domain	Node	Edge	Label	$\|\mathbf{V}\|$	$\|\mathbf{E}\|$	$\|\mathbf{Y}\|$	$\|\mathbf{X}\|$	$\|\mathbf{Y}_{id}\|$	$\|\mathbf{Y}_{ood}\|$	$\|\mathbf{N}_{id}\|$	$\|\mathbf{N}_{ood}\|$
DBLP	Citation	Author	Co-authorship	Research areas	28,702	68,335	4	300	3	1	21, 553	4,539
Facebook	Social	User	Contacts	Groups	792	14,024	17	319	14	3	524	243
BlogCatalog	Social	User	Contacts	Topic Categories	10,312	333,983	39	128	25	14	8,513	1,037
Flickr	Social	User	Contacts	Interest Groups	80,513	5,899,882	195	128	150	45	57,185	14,775
PPI	Biology	Protein	Interaction	Funtions	56,944	409,358	121	50	100	21	1,748	33
Movielens	Movie	Movie	Co-director	Genres	7,805	55,832	20	5,000	11	9	4,338	998
Yeast	Biology	Gene	Interaction	Funtions	681	910	13	200	5	8	138	13

Table 2: The performance for multi-label OOD detection in terms of AUC (mean ± std).

Backbone	Method	AUC						
		DBLP	Facebook	BlogCatalog	Flickr	PPI	Movielens	Yeast
GCN	Backbone	0.518 ± 0.006	0.523 ± 0.012	0.423 ± 0.013	0.450 ± 0.006	0.491 ± 0.023	0.537 ± 0.003	0.698 ± 0.021
	Dropout	0.634 ± 0.002	0.503 ± 0.009	0.536 ± 0.010	0.500 ± 0.007	0.608 ± 0.035	0.484 ± 0.001	0.530 ± 0.018
	Ensemble	0.643 ± 0.002	0.507 ± 0.006	0.504 ± 0.004	0.500 ± 0.007	0.569 ± 0.001	0.489 ± 0.003	0.583 ± 0.033
	Mahalanobis	0.508 ± 0.009	0.603 ± 0.081	0.501 ± 0.005	0.522 ± 0.015	0.518 ± 0.050	0.520 ± 0.003	0.501 ± 0.069
	JointEnergy	0.645 ± 0.005	0.613 ± 0.023	0.527 ± 0.016	0.504 ± 0.009	0.530 ± 0.034	0.490± 0.008	0.524 ± 0.053
	Ours	**0.655 ± 0.004**	**0.846 ± 0.048**	**0.612 ± 0.021**	**0.552 ± 0.010**	**0.668 ± 0.052**	**0.556 ± 0.007**	**0.746 ± 0.021**
GAT	Backbone	0.422 ± 0.002	0.425 ± 0.003	0.464 ± 0.001	0.497 ± 0.004	0.522 ± 0.087	0.469 ± 0.001	0.646 ± 0.016
	Dropout	0.759 ± 0.001	0.913 ± 0.021	**0.612 ± 0.027**	0.484 ± 0.008	0.571 ± 0.111	0.552 ± 0.002	0.542 ± 0.061
	Ensemble	0.757 ± 0.003	0.920 ± 0.008	0.577 ± 0.002	0.486 ± 0.003	0.591 ± 0.006	0.562 ± 0.004	0.588 ± 0.073
	Mahalanobis	0.537 ± 0.026	0.661 ± 0.081	0.541 ± 0.010	0.502 ± 0.004	0.517 ± 0.038	0.519 ± 0.018	0.542 ± 0.050
	JointEnergy	0.758 ± 0.003	0.908 ± 0.016	0.568 ± 0.027	0.500 ± 0.007	0.512 ± 0.013	0.545 ± 0.010	0.557 ± 0.036
	Ours	**0.811 ± 0.008**	**0.922 ± 0.028**	0.565 ± 0.028	**0.512 ± 0.002**	**0.598 ± 0.002**	**0.628 ± 0.014**	**0.763 ± 0.005**
GraphSAGE	Backbone	0.489 ± 0.006	0.326 ± 0.041	0.501 ± 0.001	0.500 ± 0.006	0.457 ± 0.001	0.430 ± 0.001	0.641 ± 0.023
	Dropout	0.768 ± 0.001	**0.957 ± 0.007**	0.698 ± 0.001	0.492 ± 0.008	0.806 ± 0.018	0.609 ± 0.003	0.637 ± 0.065
	Ensemble	0.762 ± 00013	0.956 ± 0.005	0.697 ± 0.006	0.492 ± 0.005	**0.808 ± 0.034**	0.604 ± 0.003	0.612 ± 0.020
	Mahalanobis	0.400 ± 0.016	0.612 ± 0.091	0.410 ± 0.004	0.502 ± 0.003	0.523 ± 0.044	0.452 ± 0.023	0.551± 0.018
	JointEnergy	0.765 ± 0.002	0.901 ± 0.026	**0.700 ± 0.003**	0.499 ± 0.003	0.721 ± 0.013	0.592 ± 0.011	0.513 ± 0.046
	Ours	**0.796 ± 0.001**	0.937 ± 0.028	0.615 ± 0.021	**0.528 ± 0.008**	0.762 ± 0.006	**0.623 ± 0.004**	**0.741 ± 0.003**
-	MLGW	0.566 ± 0.004	0.497 ± 0.031	0.502 ± 0.002	0.495 ± 0.010	0.617 ± 0.010	0.532 ± 0.004	0.538 ± 0.042
	LANC	0.494 ± 0.049	0.681 ± 0.008	0.478 ± 0.009	0.507 ± 0.009	0.449 ± 0.056	0.481 ± 0.002	0.568 ± 0.014
	MLGD	0.512 ± 0.003	0.689 ± 0.007	0.508 ± 0.024	0.511 ± 0.011	0.627 ± 0.005	0.517 ± 0.008	0.615 ± 0.014

node j is $\boldsymbol{h_{ij}}(\mathbf{y}_i, d_{ij}) = [h_{ij}^1, h_{ij}^2, ..., h_{ij}^k, ..., h_{ij}^K]$. And h_{ij}^k is obtained by:

$$h_{ij}^k = \begin{cases} 0 & y_{jk} = 0, \\ g\left(d_{ij}\right) & y_{jk} = 1, \end{cases} \tag{3.12}$$

where $\mathbf{y}_j = [y_{j1}, ..., y_{jk}, ..., y_{jK}] = [0,1]^K$ represents the ID labels of training node j. The prior positive parameter is estimated as:

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{i}} = \sum_{j \in \mathbb{L}} \boldsymbol{h_{ij}}(\mathbf{y}_j, d_{ij}) + \mathbf{1}, \tag{3.13}$$

where $\mathbb{L}$ is the set of labeled nodes. Since the multi-label positive evidence is more importance in multi-label OOD detection, we only estimate the prior positive evidence in this section. During the training process, we minimize $\mathcal{L}_{PE} = \sum_{i=1}^{N} \hat{\boldsymbol{\alpha}}_{\boldsymbol{i}} \log \frac{\hat{\boldsymbol{\alpha}}_{\boldsymbol{i}}}{\boldsymbol{\alpha}_{\boldsymbol{i}}}$. The total loss function we use to optimize the model is:

$$\mathcal{L}_{total} = \mathcal{L}_{Beta} + \lambda \cdot \mathcal{L}_{PE}, \tag{3.14}$$

where λ denotes a trade-off parameter with $\mathcal{L}_{PE}$.

3.4 Multi-Label Opinions Fusion After obtaining separate beliefs of multiple labels, we need to combine these opinions and quantify a integrate opinion, *i.e.*, Opinions Fusion. Note that, if a sample belongs to any label we already know, then it is an ID sample. Only samples that do not belong to any known category should be classified as OOD samples. Hence, naive operations like summing up all the beliefs are inapplicable for multi-label setting.

Multi-Label Joint Belief. Inspired by the multiplication in Subjective Logic[17], a multi-label opinion $\boldsymbol{\Omega} = \omega_1 \vee \omega_2 \vee \cdots \vee \omega_K$. Based on that, the multi-label joint belief over all classes is defined as:

$$\mathbf{b} = b_1 \vee b_2 \vee \cdots \vee b_K. \tag{3.15}$$

Let $\mathcal{C}_m = \{0,1\}$ and $\mathcal{C}_n = \{0,1\}$ be two different class domain. $\omega_m = (b_m, d_m, u_m, a_m)$ and $\omega_n = (b_n, d_n, u_n, a_n)$ are binomial opinions on $\mathcal{C}_m$ and $\mathcal{C}_n$. The joint opinion $\omega_{m \vee n} = \omega_m \vee \omega_n$ can be formulated as:

$$\begin{aligned} b_{m\vee n} &= b_m + b_n - b_m b_n, \\ d_{m\vee n} &= d_m d_n + \frac{a_m\left(1-a_n\right) d_m u_n + \left(1-a_m\right) a_n u_m d_n}{a_m + a_n - a_m a_n}, \\ u_{m\vee n} &= u_m u_n + \frac{a_n d_m u_n + a_m u_m d_n}{a_m + a_n - a_m a_n}, \\ a_{m\vee n} &= a_m + a_n - a_m a_n, \end{aligned} \tag{3.16}$$

where the joint belief $\mathbf{b}$ can be calculated by Eq.(3.16) iteratively. As shown in Fig 2 (Inference), only samples which do not belong to any known classes will have a relative low joint belief, which can effectively differentiate them from in-distribution sample. Thus, we use the joint belief to distinguish whether a sample is out-of-distribution. With a higher joint belief, we shall be more confident to consider a sample as in-distribution

Table 3: The performance for multi-label OOD detection in terms of AUPR (mean ± std).

Backbone	Method	AUPR						
		DBLP	Facebook	BlogCatalog	Flickr	PPI	Movielens	Yeast
GCN	Backbone	0.553 ± 0.009	0.519 ± 0.016	0.454 ± 0.008	0.500 ± 0.007	0.589 ± 0.011	0.546 ± 0.003	0.690 ± 0.027
	Dropout	0.609 ± 0.001	0.475 ± 0.005	0.519 ± 0.010	0.501 ± 0.013	0.567 ± 0.026	0.485± ± 0.001	0.602 ± 0.027
	Ensemble	0.614 ± 0.001	0.560 ± 0.025	0.505 ± 0.004	0.512 ± 0.005	0.534 ± 0.008	0.488 ± 0.002	0.604 ± 0.086
	Mahalanobis	0.524 ± 0.003	0.575 ± 0.093	0.499 ± 0.003	0.508 ± 0.009	0.560 ± 0.057	0.520 ± 0.003	0.576 ± 0.073
	JointEnergy	0.659 ± 0.004	0.547 ± 0.011	0.541 ± 0.023	0.492 ± 0.006	0.570 ± 0.036	0.483 ± 0.006	0.564 ± 0.045
	Ours	**0.681 ± 0.003**	**0.875 ± 0.071**	**0.657 ± 0.015**	**0.565 ± 0.005**	**0.703 ± 0.062**	**0.547 ± 0.003**	**0.781 ± 0.027**
GAT	Backbone	0.535 ± 0.001	0.315 ± 0.002	0.534 ± 0.001	0.488 ± 0.007	0.591 ± 0.056	0.505 ± 0.001	0.739 ± 0.005
	Dropout	0.734 ± 0.003	0.909 ± 0.026	0.563 ± 0.027	0.497 ± 0.010	0.584 ± 0.058	0.536 ± 0.002	0.524 ± 0.043
	Ensemble	0.734 ± 0.001	**0.937 ± 0005**	0.539 ± 0.001	0.500 ± 0.008	0.585 ± 0.011	0.543 ± 0.004	0.597 ± 0.072
	Mahalanobis	0.537 ± 0.029	0.661 ± 0.081	0.553 ± 0.019	0.501 ± 0.004	0.568 ± 0.034	0.533 ± 0.009	0.556 ± 0.030
	JointEnergy	0.779 ± 0.003	0.894 ± 0.025	0.611 ± 0.017	0.477 ± 0.003	0.634 ± 0.011	0.530 ± 0.011	0.531 ± 0.065
	Ours	**0.813 ± 0.002**	0.936 ± 0.029	**0.613 ± 0.029**	**0.510 ± 0.003**	**0.664 ± 0.001**	**0.638 ± 0.010**	**0.789 ± 0.002**
GraphSAGE	Backbone	0.523 ± 0.005	0.421 ± 0.028	0.386 ± 0.001	0.504 ± 0.011	0.461 ± 0.002	0.480 ± 0.001	0.710 ± 0.012
	Dropout	0.748 ± 0.002	0.940 ± 0.017	0.663 ± 0.003	0.483 ± 0.028	0.790 ± 0.015	0.590 ± 0.005	0.560 ± 0.044
	Ensemble	0.739 ± 0.001	**0.951 ± 0.004**	0.662 ± 0.002	0.484 ± 0.009	0.785 ± 0.035	0.585 ± 0.003	0.601 ± 0.001
	Mahalanobis	0.437 ± 0.007	0.615 ± 0.053	0.445 ± 0.005	0.503 ± 0.003	0.576 ± 0.052	0.478 ± 0.012	0.520 ± 0.028
	JointEnergy	0.776 ± 0.001	0.912 ± 0.028	**0.723 ± 0.005**	0.529 ± 0.001	0.756 ± 0.017	0.592 ± 0.012	0.522 ± 0.048
	Ours	**0.796 ± 0.001**	0.942 ± 0.026	0.647 ± 0.018	**0.536 ± 0.013**	**0.803 ± 0.005**	**0.634 ± 0.007**	**0.784 ± 0.005**
-	MLGW	0.511 ± 0.010	0.499 ± 0.009	0.505 ± 0.003	0.498 ± 0.005	0.610 ± 0.042	0.522 ± 0.011	0.581 ± 0.067
	LANC	0.518 ± 0.026	0.651 ± 0.002	0.478 ± 0.002	0.499 ± 0.012	0.481 ± 0.016	0.505 ± 0.002	0.623 ± 0.050
	MLGD	0.524 ± 0.003	0.624 ± 0.001	0.502 ± 0.004	0.500 ± 0.010	0.603 ± 0.007	0.516 ± 0.003	0.690 ± 0.018

sample. For example, in Fig 2, our model assigns high joint belief for n_2 and n_3 with 0.96 and 0.83 respectively. n_2 and n_3 have similar higher joint belief as n_1 because all of them have at least one ID label. On the other hand, nodes like n_6 which do not have any ID labels will be assigned a low joint belief, *i.e.*, 0.27.

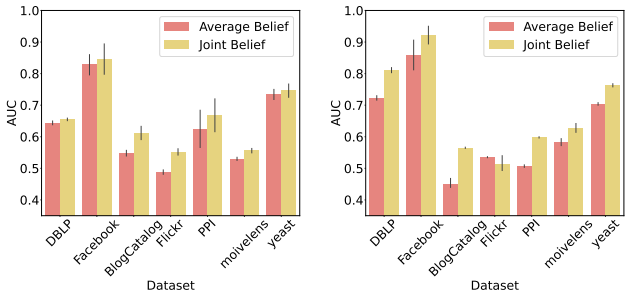
4 Experiments

4.1 Datasets The data used to validate our model are required to be graph-structured and multi-labeled. We collect 7 public available benchmark datasets to perform our experiments including DBLP[2], Facebook[47], BlogCatalog[4], Flickr[35], PPI[43], Movielens[47], and Yeast[6]. The major details of the datasets are listed in Table 1. $|\mathbf{V}|$, $|\mathbf{E}|$ and $|\mathbf{Y}|$ represent the number of nodes, the number of edges, and the number of labels, respectively. $|\mathbf{X}|$ denote the dimensions of node features. $|\mathbf{Y}_{id}|$ and $|\mathbf{Y}_{ood}|$ denote the number of ID classes and OOD classes, respectively. $|\mathbf{N}_{id}|$ and $|\mathbf{N}_{ood}|$ denote the number of ID nodes and OOD nodes, respectively. Detailed information of each dataset is in Appendix B.

4.2 Experimental Setting and Baselines Different from multi-class, for the multi-label, an input is considered an OOD only if it does not contain any ID labels[37]. For OOD sample, its label set should have no intersection with the training label set and therefore should not be predicted by the model. For multi-label OOD detection, specific to different datasets, we select some classes as OOD classes and trained the models based on training nodes which **only** own labels of the other classes, *i.e.*, ID classes. The numbers of ID nodes and OOD nodes are listed in Table 1: $|\mathbf{N}_{id}|$ and $|\mathbf{N}_{ood}|$. For testing nodes, we randomly select the same number of ID testing nodes as OOD nodes from the whole ID nodes-set. For example, for DBLP, OOD testing nodes number (unlabeled) is $4,539$, ID testing nodes number (unlabeled) is $4,539$, and ID training nodes number (labeled) is $21,533 - 4,539 = 16,994$. We do not need any labeled OOD data for model training. To summarize, there are **3** kinds of nodes: **ID training nodes, ID testing nodes and OOD testing nodes**.

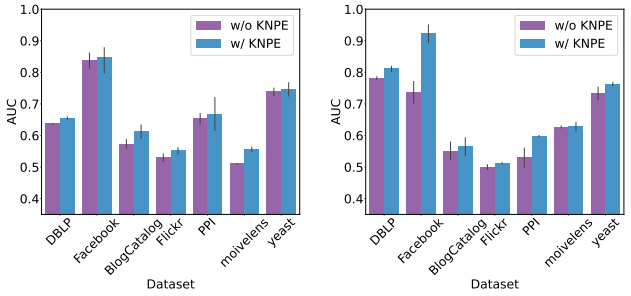
The effectiveness of our method is validated using 3 GNN models as backbone: GCN[19], GAT[36] and GraphSAGE[13]. We compare our method with three state-of-the-art multi-label classification methods, MLGW[2], LANC[47] and MLGD[33]. Two traditional OOD detection methods, MC-Dropout (Dropout)[11][29] and Deep Ensembles (Ensemble)[20], which can be applied on graphs are compared with our method. One feature-based method Mahalanobis[21] and one output-based method JointEnergy[37], both can be derived post hoc from a trained model. Details of each baseline can be found in Appendix C. Our model configurations are in Appendix D.

4.3 Multi-Label OOD Detection For multi-label OOD detection, TABLE 2 and TABLE 3 show the performance of each comparing method (mean ± std) in terms of AUC and AUPR, respectively. For each backbone, the top-1 model is bolded. The results show that our method improve the performance of multi-label OOD detection over all 3 backbones. To be specific, for multi-label OOD detection AUC, our method improves 10.8% over backbone GCN, 17.9% over GAT, and 16.6% over GraphSAGE on the average of 7 benchmarks. That is because all the backbones are optimized by BCE loss with softmax layers forehead. Without the constraint of Beta prior and ReLU layers to output evidence, it is difficult to distinguish OOD nodes effectively only according to the prediction probability. For the multi-label classification methods, MLGW, LANC and MLGD, our



(a) Ablation - Belief - GCN (b) Ablation - Belief - GAT

Figure 3: Ablation Study for joint belief on multi-label OOD detection (AUC).

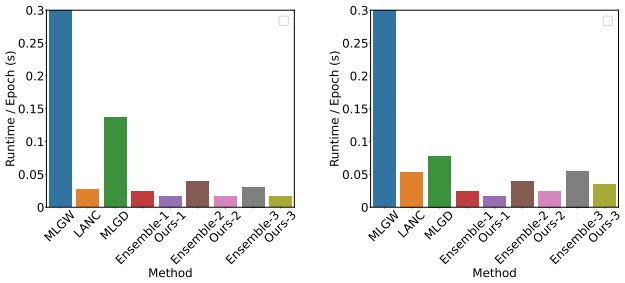


(a) Ablation - KNPE - GCN (b) Ablation - KNPE - GAT

Figure 4: Ablation Study for KNPE on multi-label OOD detection (AUC).

method also outperforms them on OOD detection for all 7 datasets with an average of 14.7% increase. Although those multi-label classification methods have considered the existence of multiple labels and the association characteristics of different labels. They are not designed for OOD setting with a lack of evaluating uncertainty. Therefore, the performance of these classification methods in multi-lable OOD detection is basically the same as that of backbones.

Moreover, compared to Dropout and Ensemble, our method has better and more stable performance, though it is slightly inferior on Facebook and BlogCatalog with GAT and GraphSAGE as backbones. We think this is acceptable due to the characteristics of different datasets and the stable performance of our method on the whole. Dropout and Ensemble are widely used for OOD detection. Nevertheless, these methods can be applied on graphs. They still have the defects of being unable to model multi-label problems. Our method outperforms Mahalanobis method on all the benchmarks. JointEnergy, which is designed for multi-label setting, performs well on some of the datasets like DBLP. Generally, our method works better on multiple datasets and different backbones which proves the effectiveness and the generalization ability of our model on different benchmarks.



(a) Dataset: DBLP (b) Dataset: Movielens

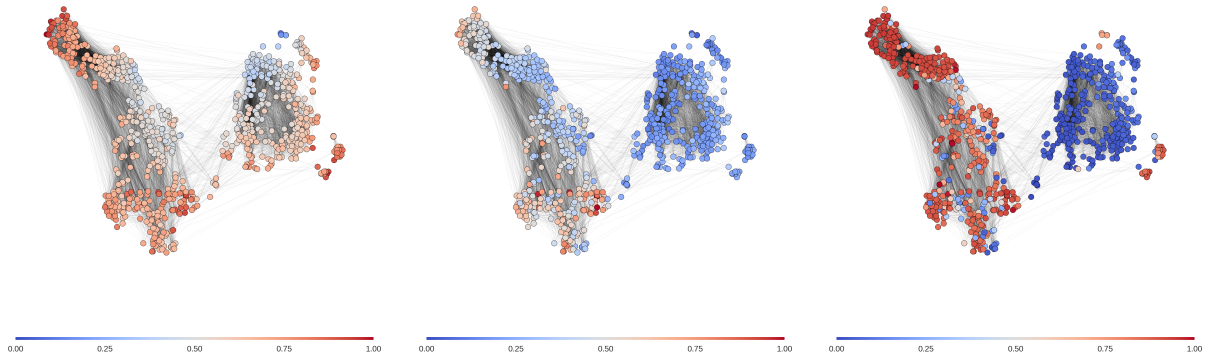
Figure 5: Runtime comparison between different methods. For backbones, -1 represents GCN, -2 represents GAT and -3 represents GraphSAGE.

4.4 Ablation Study We conduct additional experiments to demonstrate the contributions of our two key technical components, joint belief and KNPE.

4.4.1 Joint Belief To evaluate the effectiveness of joint belief, we perform a ablation study on multi-label OOD detection. To be specific, we replace the joint belief with a simple averaging belief $\frac{\sum_{k=1}^{K} b_{ik}}{K}$ for each testing node i. As shown in Fig 3, we compare joint belief with average belief for the performance on backbone GCN and GAT in terms of AUC. The standard deviation of the results are indicated by the vertical lines on the column charts. Generally, compared to Average Belief, the applying of joint belief improve the performance of models over different backbones. It confirms the validity of joint belief to combine multiple belief and form the final fusion opinion.

4.4.2 KNPE To further measure the effect of KNPE, we conduct experiments both with and without KNPE on multi-label OOD detection. For those without KNPE, we only use Beta loss to update our model and joint belief to predict OOD nodes. We compare our full method against a version without KNPE under AUC and AUPR on different benchmarks. As shown in Fig 4, the KNPE component enhance OOD detection over different backbones.

4.5 Efficiency Analysis In addition, we compare the average runtime of our method and MLGW, LANC, MLGD, and Ensemble (for 3 backbones) to verify the efficiency of our method. As shown in Fig 5, MLGW is more time-consuming than others due to the process of graph walks conducted by multiple label-specific agents. Our method is faster than LANC and MLGD because the accession of ML-EGNNs do not significantly increase the number of model parameters compared to backbones. In addition, our method is faster than Ensemble and takes half or less time. This is common sense because the ensemble method inherently requires



(a) Probability (b) Average Belief (c) Joint Belief

Figure 6: Graph embedding representations of the Facebook dataset with different quantitative methods on multi-label OOD detection experiment using GraphSAGE. The color of nodes denotes the belief value predicted by model. Red means higher belief to distinguish a node to be in-distribution while blue represents the opposite. (a) Backbone optimized by BCE loss; (b) ML-EGNNs with average belief; (c) ML-EGNNS with joint belief.

more training times.

4.6 Visualization In Fig 6, we present the t-SNE visualization of embeddings from Facebook obtained using GraphSAGE. The ID nodes cluster on the left, while the OOD nodes are on the right. In Fig 6 (a), we use the average probabilities of multiple classes as belief values, resulting in high, uniform belief across almost all nodes. Fig 6 (b) employs average belief $\frac{\sum_{k=1}^{K} b_{ik}}{K}$, with the OOD cluster shown in blue, performing better due to ML-EGNNs optimized by Beta loss. However, some ID nodes still receive low belief levels incorrectly. In Fig 6 (c), using joint belief effectively distinguishes ID from OOD nodes, with distinct red and blue colors indicating valid OOD detection results. Overall, our ML-EGNNs with joint belief achieve the best visualization among the three schemes.

5 Related Work

5.1 Multi-Label Classification on Graphs Due to the non-Euclidean datatype of graphs[33], multi-label classification on graphs is more challenging than Euclidean data like images[5][23]. MLGW[2] is the first work focusing on the multi-label node classification task, in the form of simultaneous graph walks. MINE[31] and ML-GCN[32] model node-node network and label-label network to enhance the node representation learning. LANC[47] is a label attentive neighborhood convolution model which leverages structure, attribute and label information simultaneously. MLGD[33] generates both the node embedding and the label embedding together via a deep probabilistic model to capture higher-order multi-label correlations. Besides, MLGNC[45] proposes a synthetic multi-label graph generator with tunable properties for multi-label node classification benchmark. Despite this, these methods fail to distinguish OOD samples from ID samples, as they lack uncertainty modeling and confidently classify OOD nodes into ID classes based solely on training data.

5.2 Out-of-Distribution Detection on Graphs Despite discussions on multi-label OOD detection in images, as highlighted in JointEnergy[1], there are limited studies on OOD detection in graphs. This topic is closely related to the estimation of uncertainty in semi-supervised node classification[1, 34]. One way is to introduce Bayesian-based (Dropout)[11] methods or Ensemble methods[20] on graphs, then apply entropy[18] or NLL to measure the uncertainty of samples and detect OOD samples[28, 14, 8, 26]. Another line of research in prediction uncertainty modeling is to employ prior distributions on model parameters based on Subjective Logic and Belief Theory. S-BGCN-T-K[46] parameterized a Dirichlet conjugate prior combing with Graph-Based Kernel and Teacher Network. GPN[34] performs a Bayesian update over the class predictions based on density estimation and diffusion. GNNSAFE[41] utilizes energy-based belief propagation and introduces an auxiliary regularization term, which serving as outlier exposure. These methods are not suitable for multi-label graphs, where classification probabilities follow multiple binomial distributions rather than a categorical distribution.

6 Conclusion

In this work, we first propose and formulate the multi-label OOD detection problem on graphs. To address this problem, we introduce a novel evidential method, Multi-Label Evidential Graph Neural Networks (ML-EGNNs) , to predict uncertainty for multiple classes. Our interpretation of joint belief combining multiple classes incorporates the idea of multiplication in Subjective Logic (SL). Besides, a Kernel-based Node Pos-

itive Evidence Estimation (KNPE) method is applied for estimating prior evidence. Experimental results prove both the effectiveness and efficiency of our proposed method ML-EGNNs on detecting OOD samples in multi-label graphs. For this work, we considers OOD nodes which only contain OOD labels. In the future, we will leverage detection on nodes that contain both ID labels and OOD labels under multi-label setting, which is a more challenging and untouched issue.

7 Acknowledgments

This work is supported by NSFC program (No. 62272338).

References

[1] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297.

[2] Uchenna Akujuobi et al. "Collaborative graph walk for semi-supervised multi-label node classification". In: *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2019, pp. 1–10.

[3] André CPLF de Carvalho and Alex A Freitas. "A tutorial on multi-label classification techniques". In: *Foundations of computational intelligence volume 5* (2009), pp. 177–195.

[4] Haochen Chen et al. "Harp: Hierarchical representation learning for networks". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[5] Zhao-Min Chen et al. "Multi-label image recognition with graph convolutional networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5177–5186.

[6] Jie Cheng et al. "KDD Cup 2001 report". In: *ACM SIGKDD Explorations Newsletter* 3.2 (2002), pp. 47–64.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". In: *Advances in neural information processing systems* 29 (2016).

[8] Pantelis Elinas, Edwin V Bonilla, and Louis Tiao. "Variational inference for graph convolutional networks in the absence of graph data and adversarial settings". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18648–18660.

[9] Matthias Fey and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428* (2019).

[10] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. "Exploring the limits of out-of-distribution detection". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 7068–7081.

[11] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[12] Tilmann Gneiting and Adrian E Raftery. "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American statistical Association* 102.477 (2007), pp. 359–378.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in neural information processing systems* 30 (2017).

[14] Arman Hasanzadeh et al. "Bayesian graph neural networks with adaptive connection sampling". In: *International conference on machine learning*. PMLR. 2020, pp. 4094–4104.

[15] Yibo Hu et al. "Multidimensional uncertainty-aware evidential neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 9. 2021, pp. 7815–7822.

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

[17] AUDUN. JSANG. *Subjective Logic: A formalism for reasoning under uncertainty*. Springer, 2018.

[18] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems* 30 (2017).

[19] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

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

[21] Kimin Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in neural information processing systems* 31 (2018).

[22] Haoyang Li et al. "Ood-gnn: Out-of-distribution generalized graph neural network". In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[23] Qing Li et al. "Learning label correlations for multi-label image recognition with graph networks". In: *Pattern Recognition Letters* 138 (2020), pp. 378–384.

[24] Weitang Liu et al. "Energy-based out-of-distribution detection". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21464–21475.

[25] Yaniv Ovadia et al. "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift". In: *Advances in neural information processing systems* 32 (2019).

[26] Soumyasundar Pal, Florence Regol, and Mark Coates. "Bayesian graph convolutional neural networks using node copying". In: *arXiv preprint arXiv:1911.04965* (2019).

[27] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[28] Yu Rong et al. "Dropedge: Towards deep graph convolutional networks on node classification". In: *arXiv preprint arXiv:1907.10903* (2019).

[29] Seongok Ryu, Yongchan Kwon, and Woo Youn Kim. "Uncertainty quantification of molecular property prediction with Bayesian neural networks". In: *arXiv preprint arXiv:1903.08375* (2019).

[30] Murat Sensoy, Lance Kaplan, and Melih Kandemir. "Evidential deep learning to quantify classification uncertainty". In: *Advances in neural information processing systems* 31 (2018).

[31] Min Shi, Yufei Tang, and Xingquan Zhu. "Mlne: Multi-label network embedding". In: *IEEE transactions on neural networks and learning systems* 31.9 (2019), pp. 3682–3695.

[32] Min Shi et al. "Multi-label graph convolutional network representation learning". In: *IEEE Transactions on Big Data* (2020).

[33] Zixing Song et al. "Semi-supervised Multi-label Learning for Graph-structured Data". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, pp. 1723–1733.

[34] Maximilian Stadler et al. "Graph posterior network: Bayesian predictive uncertainty for node classification". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18033–18048.

[35] Lei Tang and Huan Liu. "Relational learning via latent social dimensions". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009, pp. 817–826.

[36] Petar Velickovic et al. "Graph attention networks". In: *stat* 1050 (2017), p. 20.

[37] Haoran Wang et al. "Can multi-label classification networks know what they don't know?" In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29074–29087.

[38] Wenjie Wang et al. "Causal Representation Learning for Out-of-Distribution Recommendation". In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 3562–3571.

[39] Xi Wang and Gita Sukthankar. "Multi-label relational neighbor classification using social context features". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 464–472.

[40] Qingyao Wu et al. "Semi-supervised multi-label collective classification ensemble for functional genomics". In: *BMC genomics* 15.9 (2014), pp. 1–14.

[41] Qitian Wu et al. "Energy-based Out-of-Distribution Detection for Graph Neural Networks". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=zoz7Ze4STUL.

[42] Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[43] Hanqing Zeng et al. "Graphsaint: Graph sampling based inductive learning method". In: *arXiv preprint arXiv:1907.04931* (2019).

[44] Min-Ling Zhang and Zhi-Hua Zhou. "A review on multi-label learning algorithms". In: *IEEE transactions on knowledge and data engineering* 26.8 (2013), pp. 1819–1837.

[45] Tianqi Zhao et al. "Multi-label Node Classification On Graph-Structured Data". In: *Transactions on Machine Learning Research* (2023). ISSN: 2835-8856. URL: https://openreview.net/forum?id=EZhkV2BjDP.

[46] Xujiang Zhao et al. "Uncertainty aware semi-supervised learning on graph data". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12827–12836.

[47] Cangqi Zhou et al. "Multi-label graph node classification with label attentive neighborhood convolution". In: *Expert Systems with Applications* 180 (2021), p. 115063.

[48] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *AI open* 1 (2020), pp. 57–81.

A Notations

Here we provide the notations and their descriptions of this paper in Table 4.

Table 4: Notations and Their Descriptions.

Notation	Description
$\mathcal{G}$	a multi-label graph
$\mathbb{V}$	the set of nodes
$\mathbb{E}$	the set of edges
$\mathbf{A}$	the adjacency matrix
$\mathbf{X}$	the node feature matrix
k	the k-th in-distribution class
$\mathbb{L}$	the set of labeled nodes
$\mathbf{Y}_{\mathbb{L}}$	the labels of the training nodes
$\mathbb{U}$	the set of unlabeled nodes
$\mathbb{U}_{ID}$	the set of unlabeled ID nodes
$\mathbb{U}_{OOD}$	the set of unlabeled OOD nodes
N	the number of nodes
K	the number of in-distribution classes
$\mathbf{p}_{\mathbb{U}}$	the class probabilities of $\mathbb{U}$
$\mathbf{b}_{\mathbb{U}}$	the joint belief of $\mathbb{U}$
$\mathbf{\Omega}$	a multi-label opinion
ω_k	the k-th binomial opinion in $\mathbf{\Omega}$
$\mathcal{C}_k$	domain of class k
$\mathbf{b}$	the joint belief of $\mathbf{\Omega}$
b_k	positive belief mass distribution of class k
d_k	negative belief mass distribution of class k
u_k	uncertainty with a lack of evidence of class k
a_k	base rate distribution of class k
W	the weight of uncertain evidence
$\mathbf{y}_i$	the class label of node i
$\mathbf{p}_i$	the class probabilities of node i
$\mathbf{b}_i$	the joint belief of node i
$\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$	the parameters of Beta distribution for node i
θ	the model parameters
$f_{pos}(\mathbf{X}, \mathbf{A}\|\theta)$	the positive evidence predicted by model
$f_{neg}(\mathbf{X}, \mathbf{A}\|\theta)$	the negative evidence predicted by model
d_{ij}	the shortest path between nodes i and j
$\boldsymbol{h_{ij}}$	the contribution of positive evidence for j from i
$\hat{\boldsymbol{\alpha}}_{\boldsymbol{i}}$	the estimated prior parameter $\boldsymbol{\alpha_i}$ of Beta distribution

B Dataset Setting

Graph-structured and multi-labeled data are required to validate our model. To conduct our experiments, we collect 7 public benchmark datasets. For each dataset, labels are divided into ID labels and OOD labels randomly, or based on the data characteristic if necessary.

DBLP[2]. It is a multi-label citation dataset[4]. Each node represents an author, and each edge represents the existence of a co-authorship relation between two nodes. There are 4 labels, and each represents a research

area: database (DB - ID = 0), data mining (DM - ID = 1), artificial intelligence (AI - ID = 2) and information retrieval (IR - ID = 3). We set label 0-2 as ID labels, label 3 as OOD label. The concatenated title of a paper published by an author is the attribute associated with their node (author).

Facebook[47]. There are several users (nodes) in this social network[5], and their circles are treated as labels. There are 21 labels. A node can be assigned to multiple circles. A node's attributes include several personal social tags, such as education status, employment status, and others. We select one typical 1684 ego network for our experiments. For OOD detection setting, the first 17 labels are treated as ID labels, while the last 3 labels are OOD labels.

BlogCatalog[4]. Each node represents a blogger on the BlogCatalog website[6]. And an edge is connected if two bloggers are friends with each other. The labels represent the categories where a blogger publishes. The original dataset contains no node attributes. For node attributes, we apply the embedding vectors which are generated by Node2Vec[7]. There are 39 labels. For OOD detection setting, the first 25 labels are treated as ID labels, while the last 14 labels are OOD labels.

Flickr[35]. It is a photo-sharing network[6] between users, where labels represent users' interest groups, such as The Sea and Travel Photography. Each user could have one or multiple groups of interest from the same labeling space of 195 labels in total. For OOD detection setting, the first 150 labels are treated as ID labels, while the last 45 labels are OOD labels. For node attributes, we apply the embedding vectors which are generated by Node2Vec[6].

PPI[43]. It is a protein-protein interactions network[6], where labels represent protein roles—in terms of their cellular functions from gene ontology (121 in total). The node attributes are generated by Node2Vec[7] based on positional gene sets, motif gene sets and immunological signatures. For OOD detection setting, labels 1-100 are treated as ID labels, while the last 111-121 labels are OOD labels.

Movielens[47]. Following LANC[47], we construct Movielens by processing Movie-Lens-2 k dataset[2] which consists of personal ratings and tags for movies. A movie is treated as a graph node. A link is created between two movies if they share a common director. The first 5000 tags assigned to movies in the original dataset are used as attributes of movies. The genres of movies are treated as labels. There are 20 genres in total. To be specific, ID labels include: *Adventure*, *Animation*, *Children*, *Comedy*, *Fantasy*, *Western*, *Musical*, *Romance*, *Documentary*, *Drama*, and *IMAX*; OOD labels include: *Crime*, *Thriller*, *Horror*, *Mystery*, *Sci-Fi*, *War*, *Film-Noir*, *Action*, *Thriller*, and *Short*.

Yeast[6]. This dataset is obtained from KDD Cup 2001[3]. It consists of a variety of details about various genes of a particular organism. In our experiments, a gene is considered as a node. The interactions among these genes are used to construct a graph. The functions performed by the proteins encoded by the genes are treated as labels (13 in total). For OOD detection setting, 5 labels are treated as ID labels while the rest labels are OOD labels. Specifically, ID labels are *cellular organization*, *metabolism*, *transcription*, *cellular transport and transport mechanisms*, and *energy*; OOD labels are *cell growth, cell division and DNA synthesis*, *cell rescue, defense, cell death and ageing*, *cellular biogenesis*, *cellular communication/signal transduction*, *ionic homeostasis*, *protein destination*, *protein synthesis*, and *transport facilitation*.

C Baseline Setting

Dropout[11]. By interpreting dropout probabilistic, Monte-Carlo Dropout can obtain model uncertainty from existing deep learning models. We adapt it into GNN backbones to learn probabilistic uncertainty. Measurement of uncertainty is based on the average entropy of multi-label classification prediction. Except for the drop techniques, we used the same hyperparametersas as backbones, and set Monte Carlo sampling times M= 10, dropout rate equal to 0.5.

Ensemble[20]. Following the original paper of Deep Ensemble, we apply the randomization based ensemble approach with random initialization of the model parameters. The ensemble method is treated as a uniformly-weighted mixture model, in which the predictions are combined in the end. We use 5 backbone models in our ensemble. The hyper-parameters are the same as backbones, except for the ensemble techniques. We evaluate the average entropy of multi-label classification predicted probabilities to measure the uncertainty.

Mahalanobis[21]. Mahalanobis method derives OOD indicator scores based on the maximum Mahalanobis distance among all labels. Small controlled noise are added to a test sample to make ID and OOD features more separable. For feature ensemble, we combine the Mahalanobis confidence scores from both the outputs of two GNN layers and the final output features.

[2] http://ir.ii.uam.es/hetrec2011/datasets.htm

[3] http://pages.cs.wisc.edu/~dpage/kddcup2001/

[4] https://github.com/Uchman21/MLGW/tree/master/DBLP

[5] http://snap.stanford.edu/data/ego-Facebook.html

[6] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html

[7] http://ir.ii.uam.es/hetrec2011/datasets.html

Table 5: The performance for multi-label classification in terms of micro-AUC (mean ± std). For classification, we do not present *Mahalanobis* and *JointEnergy* since they are post-hoc OOD detection methods from a training model. Their classification performance will exactly same as backbone model.

Backbone	Method	micro-AUC						
		DBLP	Facebook	BlogCatalog	Flickr	PPI	Movielens	Yeast
GCN	Backbone	**0.922 ± 0.001**	0.964 ± 0.006	0.696 ± 0.010	0.574 ± 0.003	**0.963 ± 0.001**	0.862 ± 0.002	0.949 ± 0.003
	Dropout	0.910 ± 0.001	**0.967 ± 0.007**	0.692 ± 0.007	0.566 ± 0.003	0.943 ± 0.004	0.825 ± 0.003	0.935 ± 0.013
	Deep Ensemble	0.917 ± 0.001	0.923 ± 0.014	0.686 ± 0.004	0.731 ± 0.001	0.955 ± 0.001	0.836 ± 0.003	0.925 ± 0.002
	Ours	0.908 ± 0.008	0.940 ± 0.013	**0.806 ± 0.002**	**0.817 ± 0.002**	0.957 ± 0.002	**0.871 ± 0.001**	**0.951 ± 0.002**
GAT	Backbone	0.964 ± 0.001	0.963 ± 0.001	0.740 ± 0.001	0.770 ± 0.004	**0.939 ± 0.004**	**0.881 ± 0.001**	0.931 ± 0.003
	Dropout	**0.972 ± 0.003**	0.967 ± 0.003	**0.785 ± 0.008**	0.788 ± 0.003	0.928 ± 0.005	0.855 ± 0.002	0.920 ± 0.011
	Deep Ensemble	0.969 ± 0.002	**0.973 ± 0.001**	0.770 ± 0.003	**0.791 ± 0.002**	0.926 ± 0.002	0.853 ± 0.001	0.916 ± 0.004
	Ours	0.954 ± 0.002	0.940 ± 0.003	0.768 ± 0.007	0.741 ± 0.004	0.895 ± 0.007	0.817 ± 0.003	**0.936 ± 0.002**
GraphSAGE	Backbone	**0.982 ± 0.001**	0.949 ± 0.002	**0.850 ± 0.001**	0.782 ± 0.002	**0.949 ± 0.002**	**0.894 ± 0.001**	0.917 ± 0.006
	Dropout	0.977 ± 0.001	**0.956 ± 0.002**	0.849 ± 0.006	0.815 ± 0.002	0.929 ± 0.002	0.876 ± 0.003	0.897 ± 0.006
	Deep Ensemble	0.975 ± 0.001	0.955 ± 0.002	0.848 ± 0.004	0.815 ± 0.001	0.926 ± 0.002	0.854 ± 0.005	0.911 ± 0.004
	Ours	0.962 ± 0.001	0.934 ± 0.001	0.837 ± 0.001	**0.887 ± 0.003**	0.933 ± 0.001	0.856 ± 0.001	**0.929 ± 0.002**
-	MLGW	0.904 ± 0.002	0.824 ± 0.001	0.703 ± 0.060	0.503 ± 0.005	0.724 ± 0.046	0.580 ± 0.003	0.760 ± 0.020
	LANC	0.833 ± 0.017	0.862 ± 0.018	0.730 ± 0.001	0.715 ± 0.002	0.890 ± 0.005	0.874 ± 0.006	0.960 ± 0.004
	MLGD	0.636 ± 0.005	0.952 ± 0.007	0.584 ± 0.010	0.721 ± 0.002	0.901± 0.016	0.796 ± 0.006	0.972 ± 0.003

JointEnergy[37]. JointEnergy is a method for OOD detection in multi-label classification networks. It estimates the uncertainty by aggregating energy scores from multiple labels. Following the original paper, we take the summation of label-wise energy scores across all labels as the OOD indicator scores.

MLGW[2]. It integrates information from both labeled and unlabeled nodes into the learning process of node embeddings in attributed graphs under the framework of reinforcement learning. A separate agent is assigned for each label, which walks through the graph and selects which neighboring nodes to maximize the classification gain. The agent's recurrent decisions can be considered the walk path's informative nodes. For experimental setting, We set walk length T = 10 and the number of walks per node M = 5 with the limitation of max_neighbors = 40. For all the datasets, the hidden layer dimension d is 128. The model is trained for 40 epochs or until the model converges with a learning rate of 5e-3. We set the hyperparameters $\gamma = 0.9$, $\alpha = 1$, $\beta = 0.1$ according to the original paper. The bath size is 32.

LANC[47]. It uses a one-dimensional convolutional module to extract node representation vectors based on a label attentive neighborhood convolution. Meanwhile, another label attention module is applied to capture node-label dependencies. This module leverages semantic representations of output for node classification. The size of each node's neighborhood is set to 40 at most. For those nodes with less neighbors, zero padding will be applied to the corresponding rows. Based on the original paper, we set the convolutional kernels sizes to be $[2, 3]$. The dimension of the feature vector of a node after the operations of convolution and pooling is 64. We use uniformly distributed random embedding as the initial vector inputs of labels. The model is trained for 1000 epochs or until the model converges with a learning rate of 1e-3. The batch size is 64.

MLGD. In this method, both the node embedding and the label embedding are generated using a deep probabilistic model with ground-truth label sets as the prior distribution. Using these two types of embeddings, it further assigns a label confidence vector to each node, reflecting its likelihood of existing. We set the output size for graph convolutional layers as 256. The size of the embedding vector for nodes and labels is set to 64 to compare with other methods. The model is trained for 1000 epochs or until the model converges with a learning rate of 1e-3. The other hyper-parameters follow the setting suggested in the original paper.

D Model Configurations

We conduct our experiments with Pytorch[27] and PyG[9] on Ubuntu 20.04 with a CPU as Intel(R) Xeon(R) Gold 6138 (20-core 2.00GHz) and four Nvidia RTX 3090 GPUs (32GB). For all the backbones, parameters are set as follows: 2 convolution layers, interval dimension 128, output dimension 64. Then the outputs will be forwarded to a fully connected layer and a softmax layer. For the part of our method, ML-EGNNs use 2 fully connected layers and ReLU layers to obtain the positive evidence and negative evidence, respectively.

The size of all fully connected layers is set to $[64, |\mathbf{Y}_{id}|]$ for both backbones and our methods. The trade-off parameter λ is set to 2. We apply Adam SGD optimizer with a learning rate of 0.001 for at most 6000 epochs, or until the model converges. Each method is performed on each dataset 5 times independently with the same setting of parameters. And we repeat the splitting process for ID training and ID testing nodes with different random seeds. The resulting outputs are averaged to get final micro-AUC[44] for multi-label classification, AUC and AUPR for multi-label OOD detection.

E Derivations for Beta loss

As mentioned in Eq.(3.6), with N training samples and K different classes, the Beta loss can be written as:

$$\mathcal{L}_{Beta} = \sum_{i=1}^{N}\sum_{k=1}^{K}\left[-y_{ik}\mathbb{E}\left[\log\left(p_{ik}\right)\right] - \left(1-y_{ik}\right)\mathbb{E}\left[\log\left(1-p_{ik}\right)\right]\right], \tag{E.1}$$

where the term $\mathbb{E}_{p_{ik}\sim\mathbf{Beta}}\left[\log\left(p_{ik}\right)\right]$ can be obtained through the following derivation:

$$\begin{aligned}
&\mathbb{E}_{p_{ik}\sim\mathbf{Beta}}\left[\log\left(p_{ik}\right)\right] \\
&= \int_0^1 \log p_{ik} f(p_{ik};\alpha_{ik},\beta_{ik})dp_{ik} \\
&= \int_0^1 \log p_{ik} \frac{p_{ik}^{\alpha_{ik}-1}(1-p_{ik})^{\beta_{ik}-1}}{\mathrm{B}(\alpha_{ik},\beta_{ik})}dp_{ik} \\
&= \frac{1}{\mathrm{B}(\alpha_{ik},\beta_{ik})}\int_0^1 \frac{\partial p_{ik}^{\alpha_{ik}-1}(1-p_{ik})^{\beta_{ik}-1}}{\partial\alpha_{ik}}dp_{ik} \\
&= \frac{1}{\mathrm{B}(\alpha_{ik},\beta_{ik})}\frac{\partial}{\partial\alpha_{ik}}\int_0^1 p_{ik}^{\alpha-1}(1-p_{ik})^{\beta_{ik}-1}dp_{ik} \\
&= \frac{1}{\mathrm{B}(\alpha_{ik},\beta_{ik})}\frac{\partial \mathrm{B}(\alpha_{ik},\beta_{ik})}{\partial\alpha_{ik}} \\
&= \frac{\partial \ln \mathrm{B}(\alpha_{ik},\beta_{ik})}{\partial\alpha_{ik}} \\
&= \frac{\partial \ln\Gamma(\alpha_{ik})}{\partial\alpha_{ik}} - \frac{\partial \ln\Gamma(\alpha_{ik}+\beta_{ik})}{\partial\alpha_{ik}} \\
&= \psi(\alpha_{ik}) - \psi(\alpha_{ik}+\beta_{ik}),
\end{aligned} \tag{E.2}$$

where we use $\Gamma(\cdot)$ represents the Gamma function. $B(\alpha_{ik},\beta_{ik})$ is a 2-dimensional Beta function. **BCE** denotes the Binary Cross Entropy Loss. p_{ik} represents the predicted probability of sample i belonging to class k by model. y_{ik} represents the ground truth for sample i with label k, *i.e.*, $y_{ik} = 1$ means the training node i belongs to class k, otherwise $y_{ik} = 0$. $\mathbb{E}_{p_{ik}\sim\mathbf{Beta}}\left[\log\left(1-p_{ik}\right)\right]$ can be obtained through the same derivation above.

F Multi-Label In-Distribution Classification

To better evaluate the applicability of our method, we compare our ML-EGNNs with baselines on the task of multi-label node classification. We do not present Mahalanobis and JointEnergy since their classification performance will exactly same as backbone. We use all the ID testing nodes to evaluate the results on multi-label ID classification with micro-averaging AUC [44]. In Table 5, for each backbone, the top-1 model is bolded. Our method maintains an comparable in-distribution classification performance when compared with other baselines. For DBLP, Facebook, PPI and Movielens, the classification results obtained by our method are only $2\% - 3\%$ lower than the best method on average. For BlogCatalog, Flickr and Yeast, our methods even outperforms some other baselines. It demonstrates that ML-EGNNs can maintain a good classification performance when effectively detecting OOD samples.