# Long-Tailed Recognition of Evidential Experts for Graph-level Classification

Wei Ju
Sichuan University
Chengdu, Sichuan, China
juwei@scu.edu.cn

Siyu Yi*
Sichuan University
Chengdu, Sichuan, China
siyuyi@scu.edu.cn

Zhengyang Mao†
Peking University
Beijing, China
zhengyang.mao@stu.pku.edu.cn

Yifang Qin†
Peking University
Beijing, China
qinyifang@pku.edu.cn

Yifan Wang
University of International Business and Economics, Beijing, China
yifanwang@uibe.edu.cn

Zhiping Xiao
University of Washington
Seattle, WA, USA
patxiao@uw.edu

Yiwei Fu
Peking University
Beijing, China
fuyw@stu.pku.edu.cn

Ziyue Qiao
Great Bay University
Dongguan, Guangdong, China
ziyuejoe@gmail.com

Ming Zhang*†
Peking University
Beijing, China
mzhang_cs@pku.edu.cn

## Abstract

Graph-level classification involves analyzing the property of the whole graph, which is typically solved by using graph neural networks (GNNs). Existing efforts generally assume a balanced class distribution. However, real-world data often exhibit long-tailed distributions, i.e., tail classes have significantly fewer samples than head classes, and thus directly applying GNNs is eventually biased toward the head classes, resulting in limited generalization over the tail classes. Moreover, the predictions of existing algorithms are usually not trustworthy, and the trained classifiers remain ignorant to their predictive confidence. Towards this end, in this paper we develop a principled framework called GraphEVER for long-tailed graph-level classification. Technically, GraphEVER incorporates the beliefs of multiple experts and leverages the idea of subjective logic within the Dempster-Shafer Evidence Theory (DST). It can provide the evidence and uncertainty estimation for each expert, where the evidence is parameterized by a Dirichlet distribution to model class probability distribution, and the uncertainty is quantified via a well-defined theoretical framework. In this way, diverse experts can be integrated under DST to endow the classifier with both reliability and robustness. Moreover, we propose an evidence-based routing mechanism to dynamically assign experts, such that the tail classes can receive more attention, while the head classes can reduce redundant engaged experts, further cutting down the computational cost and improving the efficiency. Extensive experiments on seven datasets verify the superiority of our proposed framework.

*Corresponding authors
†National Key Laboratory for Multimedia Information Processing, School of Computer Science, Beijing Key Laboratory of Software and Hardware Cooperative Artificial Intelligence Systems, Peking University.


*WWW '26, Dubai, United Arab Emirates*

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792521

## CCS Concepts

• **Information systems** → **Data mining**; • **Mathematics of computing** → **Graph theory**.

## Keywords

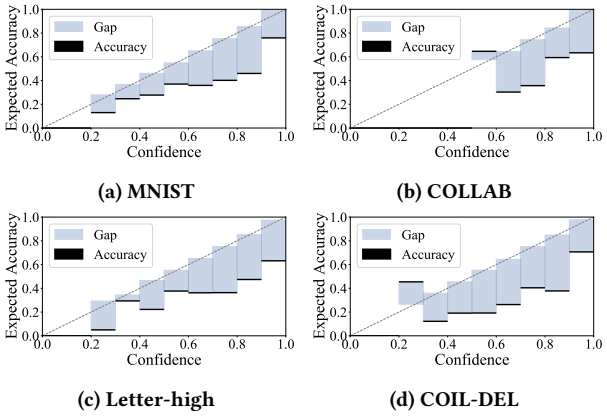
Graph Classification, Imbalanced Learning, Multi-expert Learning

**ACM Reference Format:**
Wei Ju, Siyu Yi, Zhengyang Mao, Yifang Qin, Yifan Wang, Zhiping Xiao, Yiwei Fu, Ziyue Qiao, and Ming Zhang. 2026. Long-Tailed Recognition of Evidential Experts for Graph-level Classification. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3774904.3792521

## 1 Introduction

Graph-structured data, as a kind of ubiquitous data structure, have attracted significant attention in various promising applications including social media, biochemistry, academic citation and transportation. Mining on graphs allows us to discover latent patterns and has hence received widespread interests, covering a variety of tasks and domains. Among them, graph-level classification is a fundamental problem in graph data mining, which aims to analyze and predict the property of an entire graph, such as predicting the target properties of molecules [17, 30] and analyzing the biological functionality of compounds [23].

Recently, graph neural networks (GNNs) [15, 22, 25, 50] have propelled the development of graph-level classification, which have revealed impressive performance via incorporating the attributive and structural information. However, the extensive existing GNN methods generally assume that the number of the labels is in a balanced situation, *i.e.*, the class distribution is balanced. In contrast, many datasets in real-world scenarios naturally exhibit highly-skewed class distributions [13], where a majority of classes (tail classes) contain a small number of labeled graphs, while few classes (head classes) contain abundant labeled graphs. Inevitably, directly applying GNNs on these class-imbalanced datasets is likely to occur




(a) MNIST (b) COLLAB

(c) Letter-high (d) COIL-DEL

**Figure 1: Reliability diagrams for GNNs trained on long-tailed data. The diagrams plot accuracy against confidence, where a perfect diagonal indicates ideal calibration, and any gap from this diagonal reflects model miscalibration.**

the notorious *prediction bias* phenomenon [67], where extremely imbalanced classes bias the GNN classifier towards head classes, and tend to overfit the tail classes, resulting in poor generalization over the tail classes. It thus raises a meaningful question: *How do existing GNNs address severe class imbalances?*

Actually, there are a large number of proposed methods in vision domains to address class imbalances [64, 65], which can be categorized into three groups: *re-sampling*, *re-weighting*, and *ensembling learning*. Re-sampling strategies [4, 10, 66] aim at constructing synthetic training data to achieve a more balanced distribution. Re-weighting strategies [2, 6, 54, 62] adjust the portion of training loss of different classes by assigning weights to optimize the training objective. Different from the former, ensembling learning [24, 52, 55] combines multiple classifiers in a multi-expert framework to obtain reliable and robust predictions.

Although long-tailed learning has made relatively mature progress in the field of computer vision, for graph machine learning, we have identified three key challenges: **(i) Existing research on long-tailed graph-level classification has been scarcely explored.** Although some algorithms specifically designed for node-level classification on graphs have been proposed [29, 38, 39, 48, 49, 63], long-tailed learning for graph-level classification is another very practical and important issue. **(ii) Inability to estimate the uncertainty of the predictions**. GNNs contribute to the success of predictive accuracy. However, their predictions are not necessarily trustworthy. Studies have proved that traditional neural networks easily lead to over-confidence [9, 37]. Figure 1 presents the reliability diagrams for GNNs trained on long-tailed data. As illustrated in the diagrams, GNNs exhibit over-confidence in long-tailed scenarios, where predictions have high confidence but low accuracy. Hence, the model should know the predictive confidence in its judgment instead of making an incorrect one. **(iii) Most existing long-tailed algorithms often induce excessive computational resources**. Actually, re-sampling strategies typically address the class imbalance via augmenting plenty of synthetic training data, while ensembling learning strategies generally hold the assumption that each classifier should undergo training on all available samples. For example, G$^2$GNN [53] employs a topology-enhanced upsampling technique to generate balanced samples, leveraging a large number of pairwise samples for self-consistency regularization, which significantly increases the training burden. These limitations often result in redundant computational resources. Consequently, there is a compelling need to present an approach capable of estimating prediction uncertainty or confidence while concurrently alleviating excessive computational resources.

In this paper, we develop a novel framework named **EV**idential **E**xpe**R**ts (GraphEVER) for long-tailed graph-level classification. The primary concept involves jointly predicting the classes and estimating the uncertainty of the predictions in a multi-expert framework. To achieve this goal effectively, our GraphEVER is built upon the decision support of the multiple experts to acquire the beliefs from a theory of evidence perspective [8, 14], where each expert uses the subjective logic to provide the evidence and uncertainty estimation based on the Dempster-Shafer Evidence Theory. Technically, the evidence captures the class probability distribution, parameterized using a Dirichlet distribution. And the uncertainty is quantified via a well-defined theoretical framework to express the opinion of "*I do not know*". In this way, we can well measure the predictive confidence of the trained classifier. Moreover, to cut down the redundant computational resources, we develop an evidence-based routing mechanism to dynamically assign experts, such that tail classes are capable of receiving more attention to enhance the generalization, while head classes utilize fewer experts to maintain competitive performance, further reducing computational costs. The key contributions of this survey are highlighted below:

- This paper tackles the less-explored domain of long-tailed graph-level classification. We pioneer the incorporation of evidential uncertainty to measure the predictive confidence in this task.
- We propose a novel framework integrating evidence-based uncertainty quantification into a multi-expert paradigm, and dynamically routing evidence to cut redundant expert engagement.
- Through comprehensive experiments on seven benchmark datasets, we establish the superior performance of our proposed method when compared to competitive baselines.

## 2 Preliminary

Consider a set of $N$ graphs $\mathcal{G} = \{G_i\}_{i=1}^N$, let $G_i = \{(\mathcal{V}_i, \mathcal{E}_i, y_i)\}$ denote a graph with its corresponding label $y_i \in \{1, 2, \ldots, K\}$, where $\mathcal{V}_i$ is the set of nodes, $\mathcal{E}_i$ represents the set of edges, and $K$ stands for the total number of classes. In addition, let $n_i$ represent the count of graphs in the $i$-th class ($i = 1, 2, \ldots, K$), with the assumption that $n_1 \geq n_2 \geq \ldots \geq n_K$. Then we introduce the definitions of long-tailed datasets and long-tailed graph-level classification.

*Definition 2.1* (***Long-tailed Datasets***). For a given dataset, when its classes are arranged in descending order based on their cardinalities, it qualifies as a long-tailed dataset if the distribution of the sorted classes adheres to Zipf's law [36], i.e.,

$$n_i = n_1 \times i^{-\mu}, \tag{1}$$

where $\mu$ serves as a hyper-parameter governing the degree of class imbalance. The **imbalance factor** (**IF**) quantifies this imbalance and is defined as $n_1/n_K$.

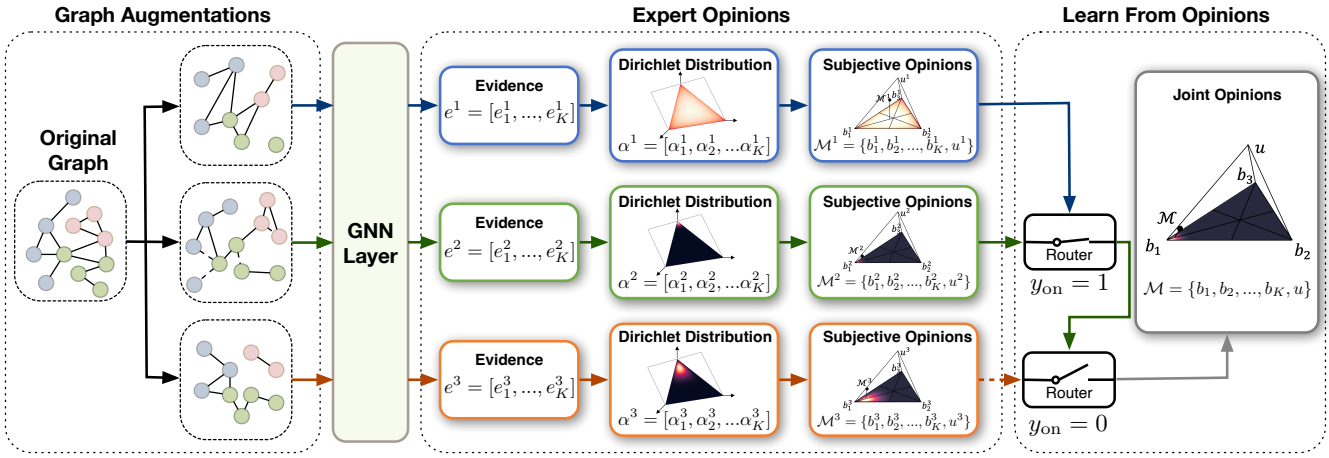



**Figure 2: Overview of the proposed Evidential Experts (GraphEVER) framework for graph-level classification.**

*Definition 2.2 (**Long-tailed Graph-level Classification**).* For a long-tailed graph dataset denoted as $\mathcal{G} = \{G_i, y_i\}_{i=1}^{N}$, the objective of long-tailed graph-level classification involves training an unbiased classifier $\mathcal{F}$, denoted by the mapping: $\mathcal{F} : \mathcal{F}(G_i) \rightarrow y_i$, such that the trained classifier can work well for graphs in both head and tail classes and generalize well on a balanced test dataset.

**GNN Classifier.** To derive an effective probability assignment for each input graph, we begin by leveraging the GNNs [18] to extract accurate information of feature attributes and structural topology. Specifically, let $f_\theta(\cdot)$ denote the GNN classifier with the network parameters $\theta$, consisting of a $L$-layer GNN encoder and a multilayer perceptron (MLP) classifier, the propagation rule of in the $l$-th layer of GNN encoder can be expressed as:

$$\mathbf{h}_v^{(l)} = \mathcal{U}_\theta^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathcal{A}_\theta^{(l)}\left(\left\{\mathbf{h}_u^{(l-1)}\right\}_{u\in\mathcal{N}(v)}\right)\right), \tag{2}$$

where $\mathbf{h}_v^{(l)}$ represents the node $v$'s representation at the $l$-th layer, and $\mathcal{N}(v)$ denotes the set of first-order neighbor nodes of node $v$. The functions $\mathcal{U}_\theta^{(l)}$ and $\mathcal{A}_\theta^{(l)}$ are associated with the updating and aggregation processes at the $l$-th layer. Hereafter, we can make further pooling operation [59] for the output node representations after $L$ iterations, and summarize the graph representation $\mathbf{h}$ for graph $G$ using the *READOUT* function:

$$\mathbf{h} = \mathit{READOUT}(\{\mathbf{h}_v^{(L)} : v \in \mathcal{V}\}), \tag{3}$$

Hence, the obtained graph representation proves valuable for subsequent graph-level classification tasks. Formally, we express the probability assignment for graph $G$ as follows:

$$f_\theta(G) = \mathit{softmax}(\mathrm{MLP}(\mathbf{h})). \tag{4}$$

However, the *softmax* function, while commonly used for estimating class probabilities in graph samples, has limitations. It tends to provide a point estimate, leading to potential over-confidence. Besides, the *softmax* function fails to provide the associated uncertainty. Therefore, our primary focus is on enhancing the accuracy of the GNN classifier and concurrently addressing the challenge of quantifying its uncertainty.

## 3 Methodology

In this paper, we introduce our framework GraphEVER for long-tailed graph-level classification. An illustration of the framework is presented in Figure 2. Next, we will introduce the evidence-based uncertainty module and the dynamic routing module. Finally, the training algorithm to optimize the model is explained.

### 3.1 Evidence-based Uncertainty Module

It has been shown that traditional neural networks easily suffer from over-confidence [9, 37]. This phenomenon is further exacerbated in the long-tailed classification, especially important in tail classes with limited training samples, which shows the great uncertainty in the learning process.

In this section, we detail evidential deep learning to quantify the evidence confidence of each expert. This approach not only models the probability for each class but also captures the overarching uncertainty of predictions. Then we reduce the uncertainty of prediction by combining the subjective opinions of multiple experts through evidence theory. Finally, we elaborate on the training details of the learning objective.

*3.1.1* ***Estimating Uncertainty for Each Expert.*** In probabilistic frameworks, the Dempster–Shafer Theory of Evidence (DST), or evidence theory, emerges as an extension of Bayesian principles with a focus on subjective probabilities [8]. DST employs belief functions to distribute belief masses across mutually exclusive sets of potential states, such as diverse class labels present in a sample. This framework enables the expression of uncertainty in predictions by acknowledging "*I do not know*" as a valid opinion [14, 42]. Subjective logic serves as the formalization tool for belief assignments and establishes a theoretical framework for acquiring the probabilities (belief masses) of different classes while addressing the overall uncertainty inherent in predictions.

Formally, for a classification task involving $K$ categories, we establish $K$ distinct and mutually exclusive singletons (such as class labels). Each singleton, denoted as $k = 1, 2, \ldots, K$, is assigned a belief mass represented by $b_k$, and there is an additional mass termed the overall uncertainty mass denoted as $u$. These $K+1$ mass

values are strictly non-negative and collectively sum to one. This definition can be expressed as follows:

$$u + \sum_{k=1}^{K} b_k = 1, \tag{5}$$

where $b_k \geq 0$ for $k = 1, \ldots, K$ and $u \geq 0$. For classification, the variables $b_k$ and $u$ are interpreted as the probability associated with the $k$-th class and the overall uncertainty. Additionally, let $e_k \geq 0$ denote the evidence corresponding to the belief mass $b_k$ for a given singleton $k$. Then, the belief mass $b_k$ and uncertainty $u$ can be defined as:

$$b_k = \frac{e_k}{S}, \text{ and } \quad u = \frac{K}{S}, \tag{6}$$

where $S = \sum_{i=1}^{K}(e_i+1)$. For a more precise quantification of evidence, subjective logic associates the assignment of belief mass with a Dirichlet distribution, characterized by parameters $\alpha_k = e_k + 1$. Consequently, a subjective opinion can be constructed using the Dirichlet distribution parameters, expressed as $b_k = (\alpha_k - 1)/S$, where $S = \sum_{i=1}^{K} \alpha_i$ denotes the Dirichlet strength. The Dirichlet distribution can be formally defined as follows:

*Definition 3.1 (**Dirichlet Distribution**).* It arises as an extension of the Beta distribution to the multivariate domain and is parameterized by its $K$ concentration parameters $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$. The probability density function for the vector $\boldsymbol{p}$ is defined as follows:

$$D(\boldsymbol{p} \mid \boldsymbol{\alpha}) = \begin{cases} \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} p_i^{\alpha_i - 1} & \text{for } \boldsymbol{p} \in \mathcal{S}_K, \\ 0 & \text{otherwise .} \end{cases} \tag{7}$$

Here, $\mathcal{S}_K$ represents the $K$-dimensional unit simplex, defined as:

$$\mathcal{S}_K = \left\{ \boldsymbol{p} \mid \sum_{i=1}^{K} p_i = 1 \text{ and } 0 \leq p_1, \ldots, p_K \leq 1 \right\}, \tag{8}$$

and $B(\boldsymbol{\alpha})$ represents the K-dimensional multinomial beta function. As a practical tool, Dirichlet distribution can typically be regarded as the conjugate prior of the multinomial distribution [1].

Therefore, given a sample, the parameters of the Dirichlet distribution used for classification can be interpreted as evidence $e_k$ for $k$-th class, and a Dirichlet distribution parametrized over evidence models second-order probabilities and uncertainty [14]. Specifically, we leverage the output of the GNN to construct the multinomial opinions. Formally, when presented with a graph sample $G$, the GNN classifier predicts the evidence vector for each class, denoted as $\boldsymbol{e} = [e_1, \ldots, e_K]$. This prediction is achieved by substituting the *softmax* layer with a non-negative activation function, such as *ReLU*, as specified in Eq. (4). Consequently, the Dirichlet distribution parameters become $\boldsymbol{\alpha} = f_\theta(G) + 1$, and the mean $\boldsymbol{\alpha}/S$ can be computed as an estimation of the class probabilities.

In practice, to enhance the diversity of experts, we introduce graph augmentations into our framework to encourage the discrepancy. Data augmentation plays a crucial role in generating diverse and meaningful data by applying specific transformations without altering the inherent semantics [5]. In particular, in graph domains, GraphCL [60] introduces four types of graph transformations via augmenting topological and attributive information of the graphs, which are beneficial for enhancing the diversity of experts. Specifically, for each expert, the graph $G$ undergoes stochastic graph augmentations denoted as $\mathcal{T}(\cdot|G)$. This process results in a semantically preserved augmented graph, denoted as $\hat{G}$, achieved by randomly selecting one of four augmentation strategies mentioned above. Then the parameters of the Dirichlet distribution can be recalculated as $\boldsymbol{\alpha} = f_\theta(\hat{G}) + 1$. In this way, for each expert, we can measure the evidence-based uncertainty directly from the outputs of GNNs, and such evidence theoretically addresses the problem of over-confidence, which is especially effective for long-tailed recognition.

*3.1.2* ***Combining Multiple Experts with DST.*** Having acquired evidence and uncertainty for each expert, each one may still have its inherent prediction preference. How to combine all available evidence to eliminate bias and yield more trustworthy predictions is a nontrivial problem. Actually, one reliable solution is the ensembling learning through multiple experts, and Dempster-Shafer Evidence Theory (DST) offers a viable approach to fuse evidence from diverse sources effectively [43], thereby mitigating uncertainty, defined as:

*Definition 3.2.* (**Dempster's combination rule for two experts**) The joint mass $\mathcal{M} = \left\{\{b_k\}_{k=1}^{K}, u\right\}$ is combined from the probability mass assignments of two experts $\mathcal{M}^1 = \left\{\{b_k^1\}_{k=1}^{K}, u^1\right\}$ and $\mathcal{M}^2 = \left\{\{b_k^2\}_{k=1}^{K}, u^2\right\}$ in the following manner:

$$\mathcal{M} = \mathcal{M}^1 \oplus \mathcal{M}^2. \tag{9}$$

More specifically, the calculation rule is formulated as:

$$b_k = \frac{1}{1-C}(b_k^1 b_k^2 + b_k^1 u^2 + b_k^2 u^1), u = \frac{1}{1-C} u^1 u^2. \tag{10}$$

Here, $C = \sum_{i \neq j} b_i^1 b_j^2$ represents a conflict factor quantifying the level of inconsistency between the belief masses provided by two experts. The normalization term is defined as the scale factor $\frac{1}{1-C}$.

In a multi-expert framework involving $M$ experts, the combination of beliefs from various sources can be performed sequentially using Dempster's rule of combination, which can be formulated as:

$$\mathcal{M} = \mathcal{M}^1 \oplus \mathcal{M}^2 \oplus \cdots \mathcal{M}^M. \tag{11}$$

In this way, we can obtain the joint mass $\mathcal{M} = \left\{\{b_k\}_{k=1}^{K}, u\right\}$ under DST. Accordingly, the corresponding collective evidence from multiple experts and the parameters of the Dirichlet distribution can be determined using Eq. (6):

$$S = \frac{K}{u}, e_k = b_k \times S, \text{ and } \alpha_k = e_k + 1. \tag{12}$$

Based on the above combination rule, the derived joint evidence $\boldsymbol{e}$ from multiple experts and the associated parameters of joint Dirichlet distribution $\boldsymbol{\alpha}$ can be induced to yield the final probability for each class, along with the overall uncertainty in the prediction.

*3.1.3* ***Learning to Form Opinions.*** Here we explore the training strategy that enables the model to learn evidence for each expert [24, 42]. Actually, by replacing the *softmax* layer with a non-negative activation function, such as *ReLU*, the model outputs can be treated as the evidence vector $\boldsymbol{e}$. This, in turn, allows for the determination of the Dirichlet parameters $\boldsymbol{\alpha}$. These results can be integrated via subjective logic to reflect the opinion of the expert, namely, the confidence and uncertainty of the prediction.

Technically, for a graph sample $G_i$ along with its ground-truth class label $y_i$ represented as a one-hot encoded vector. We first construct the Dirichlet distribution $D(\boldsymbol{p}_i|\boldsymbol{\alpha}_i)$, which is a prior on

the multinomial likelihood $Multi(\boldsymbol{y}_i|\boldsymbol{p}_i)$. Then we adopt the Type II Maximum Likelihood Estimation to formulate our loss function:

$$\begin{aligned}\mathcal{L}_i &= -\log\left(\int \prod_{j=1}^{K} p_{ij}^{y_{ij}} \frac{1}{B(\boldsymbol{\alpha}_i)} \prod_{j=1}^{K} p_{ij}^{\alpha_{ij}-1} \mathrm{d}\boldsymbol{p}_i\right) \\ &= \sum_{j=1}^{K} y_{ij}\Big(\log(S_i) - \log(\alpha_{ij})\Big).\end{aligned} \tag{13}$$

The above loss indicates that the evidence of the ground-truth class is encouraged to be higher than other class labels. However, it cannot guarantee that incorrect labels would produce less evidence. To this end, we introduce a mechanism to drive the total evidence to approach zero for a graph sample that cannot be accurately classified. Formally, for each expert, we incorporate the KL-divergence $KL(\cdot)$ into the loss function:

$$\mathcal{L}_{exp} = \sum_{i=1}^{N} \mathcal{L}_i + \lambda_t \sum_{i=1}^{N} KL\Big(D(\boldsymbol{p}_i|\widetilde{\boldsymbol{\alpha}}_i)\,\Big\|\,D(\boldsymbol{p}_i|\mathbf{1})\Big). \tag{14}$$

where $\lambda_t = \min(1, t/T) \in [0,1]$, and $t$ denotes the current training epoch, $T$ refers to the annealing epoch. $D(\boldsymbol{p}_i|\mathbf{1})$ represents the uniform Dirichlet distribution, with $\mathbf{1}$ as the parameter vector consisting of $K$ ones. Furthermore, we define $\widetilde{\boldsymbol{\alpha}}_i = \boldsymbol{y}_i + (1-\boldsymbol{y}_i) \odot \boldsymbol{\alpha}_i$, where $\odot$ signifies the Hadamard product. $\widetilde{\boldsymbol{\alpha}}_i$ is the Dirichlet distribution parameter which ensures that the evidence for the ground-truth class is not mistakenly considered as zero. Specifically, the KL-divergence term can be formulated as follows:

$$\begin{aligned}KL\left[D\left(\mathbf{p}_i|\tilde{\boldsymbol{\alpha}}_i\right)\|D\left(\mathbf{p}_i|\mathbf{1}\right)\right] &= \log\left(\frac{\Gamma\left(\sum_{k=1}^{K}\tilde{\alpha}_{ik}\right)}{\Gamma(K)\prod_{k=1}^{K}\Gamma\left(\tilde{\alpha}_{ik}\right)}\right) \\ &+ \sum_{k=1}^{K}\left(\tilde{\alpha}_{ik}-1\right)\left[\psi\left(\tilde{\alpha}_{ik}\right) - \psi\left(\sum_{j=1}^{K}\tilde{\alpha}_{ij}\right)\right],\end{aligned} \tag{15}$$

where $\Gamma(\cdot)$ and $\psi(\cdot)$ are the gamma function and digamma function.

Additionally, to make the evidence contained by multiple experts as diverse as possible, an intuitive approach is to compare different experts pairwise, hoping that their prediction results are as different as possible. However, this brings a risk: if two experts make completely opposite predictions, such conflict would contradict our goal of achieving more accurate predictions through a multi-expert framework, which is unreasonable. To address this, we utilize normalized Dirichlet parameters $\alpha_{ik}^m/S_i^m$ to form Dirichlet distribution $D(\boldsymbol{p}_i|\boldsymbol{\alpha}_i^m)$, and then the diversity of experts can be achieved through KL-divergence as:

$$\mathcal{L}_{div} = -\frac{1}{M}\sum_{m=1}^{M} KL\left(P\left(\boldsymbol{p}_i \mid \boldsymbol{\alpha}_i^m\right)\|P\left(\boldsymbol{p}_i \mid \overline{\boldsymbol{\alpha}}_i\right)\right) \tag{16}$$

where $\overline{\boldsymbol{\alpha}}_i = \sum_{m=1}^{M}\boldsymbol{\alpha}_i^m/M$ denotes the averaged Dirichlet parameters, and $M$ is the number of experts. Through this strategy, the fused evidence distribution obtained through parameter averaging is more reasonably analogous to a clustering center. We aim for each expert to be appropriately distant from the clustering center, maintaining their diversity without encouraging them to optimize in completely opposite directions. Instead, we hope that while maintaining their differences, they will be trained through the previously mentioned supervised loss Eq. (14) to maintain consistency in their prediction results as much as possible. This ensures that the predictions from the multi-expert fusion framework are superior to those of individual experts.

Finally, the overall loss function of the joint training objective in our multi-expert framework can be formulated as follows:

$$\mathcal{L} = \sum_{i=1}^{M}\mathcal{L}_{exp} + \mathcal{L}_{div}. \tag{17}$$

## 3.2 Dynamic Routing Module

From the above module we know that for all class labels, we use all the experts to estimate the evidence and uncertainty of the prediction. In other words, for each class, all experts are involved even for some easily predictable classes. We argue that for many easy samples (usually in head classes), we do not need all the experts engaged, which would induce redundant computational resources and affect the efficiency of the inference process [52].

To this end, we develop a routing mechanism to dynamically assign experts, such that the tail classes can receive more attention, while the head classes can reduce redundant engaged experts. Specifically, we train a router to allocate these experts in the order needed. For example, if the first $m$ experts are in charge of a graph sample $G$, our model produces the joint evidence of each class from the first to the $m$-th expert with Dempster's combination rule, and can be used to predict for classification. Then the router makes a binary decision $y_{\text{on}}$ regarding the allocation of the $m+1$-th expert. If the $m$-th expert makes an incorrect prediction, but one of the rest $M-m$ experts classifies correctly, the router should output $y_{\text{on}}=1$ and otherwise $y_{\text{on}}=0$. In this way, to well adapt to long-tailed recognition, head classes should assign fewer experts, while tail classes will allocate as many experts as possible.

We achieve this by proposing an evidence-based binary classifier to learn each router. Specifically, for the $m$-th expert, we concatenate its evidence vector $\boldsymbol{e}^m$ with the top-$s$ ranked joint evidence vector $\boldsymbol{e}$ from the first to $m$-th expert derived from the joint mass $\mathcal{M} = \left\{\{b_k\}_{j=1}^{k}, u\right\}$ under DST according to Eq. (12), where $\mathcal{M} = \mathcal{M}^1 \oplus \mathcal{M}^2 \oplus \cdots \mathcal{M}^m$. Then we project the double-long vector to a scalar by a fully connected layer $\mathbf{W}^{(m)}$. Importantly, this layer is independent between routers. Finally, we apply the Sigmoid function $S(x) = \frac{1}{1+e^{-x}}$ to obtain a real activation value in [0,1]:

$$r(x) = S(\mathbf{W}^{(m)}[\boldsymbol{e}^m || \boldsymbol{e}_{\text{top-}s\text{-components}}]), \tag{18}$$

and this routing mechanism dynamically controls the switch of the router, and can be optimized by a binary cross-entropy loss:

$$\mathcal{L}_{\text{rout}} = -y_{\text{on}}\log\left(r(x)\right) - (1-y_{\text{on}})\log\left(1-r(x)\right). \tag{19}$$

By this means, routers dynamically control the number of engaged experts. During inference, we adopt a simple thresholding mechanism (0.5): if $r(x) < 0.5$, the classifier uses the current joint evidence for the final decision, otherwise it proceeds to the next expert. Compared to directly averaging all expert outputs in the multi-expert framework, conflicts among experts may yield inaccurate predictions. Moreover, this situation may result in the allocation of more experts than necessary, which does not effectively reduce computational costs or improve inference efficiency. Our evidence-based dynamic routing mechanism with theoretically guaranteed DST fusion rules more reasonably reflects the prediction tendencies after multi-expert fusion and more efficiently estimate the number of experts to be allocated, minimizing computational resources to the greatest extent. We summarize the optimization algorithm for GraphEVER in Algorithm 1 in Appendix A.

**Table 1: Long-tailed graph classification accuracy (%) on seven benchmark datasets, varying the degree of imbalance by two IFs. The top classification results are highlighted in boldface and the second-best results are indicated with an underline.**

| Model | COLLAB | | Synthie | | ENZYMES | | MSRC_9 | | Letter-high | | Letter-low | | COIL-DEL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IF=10 | IF=20 | IF=15 | IF=30 | IF=15 | IF=30 | IF=5 | IF=10 | IF=25 | IF=50 | IF=25 | IF=50 | IF=10 | IF=20 |
| GraphSAGE | 63.07 | 53.33 | 34.74 | 30.25 | 30.66 | 25.16 | 82.00 | 79.10 | 51.06 | 42.16 | 86.00 | 84.32 | 38.80 | 31.32 |
| Up-sampling | 72.33 | 70.25 | 35.25 | 33.50 | 32.33 | 28.50 | 83.20 | 78.50 | 53.62 | 44.20 | 88.48 | 86.72 | 39.20 | 26.96 |
| CB loss | 68.78 | 65.85 | 34.75 | 30.75 | 32.19 | 26.83 | 81.50 | 76.50 | 53.76 | 45.06 | 87.46 | 85.44 | 41.72 | 32.34 |
| LACE loss | 68.33 | 64.77 | 33.25 | 30.85 | 31.16 | 25.50 | 80.50 | 80.20 | 47.46 | 38.94 | 87.89 | 84.69 | 41.96 | 32.18 |
| Augmentation | 72.85 | 71.14 | 39.37 | 35.37 | 32.08 | 26.75 | 85.00 | 78.75 | 49.28 | 42.36 | 88.32 | 86.40 | 38.18 | 30.80 |
| $G^2GNN_n$ | 73.94 | 71.89 | 38.08 | 27.94 | 35.00 | 29.17 | 88.57 | 85.04 | 58.91 | 51.12 | 89.49 | 87.98 | 38.32 | 27.98 |
| $G^2GNN_e$ | 74.50 | 72.76 | 40.19 | 37.53 | 35.83 | 29.50 | 90.28 | 86.25 | 58.85 | 49.96 | 89.84 | 87.80 | 39.18 | 31.06 |
| GraphCL | 69.33 | 67.36 | 40.25 | 36.25 | 36.66 | 29.83 | 88.37 | 84.69 | 57.34 | 48.93 | 89.28 | 87.89 | 42.02 | 33.19 |
| SupCon | 69.25 | 67.14 | 40.34 | 37.25 | 37.08 | 30.67 | 89.44 | 85.01 | 57.29 | 48.93 | 89.12 | 87.36 | 42.93 | 34.20 |
| GraphEVER | 77.13 | 75.07 | **41.63** | **38.25** | 38.17 | **32.25** | **90.50** | **87.00** | 63.40 | 54.86 | **92.90** | **92.12** | **45.52** | **36.28** |
| $GraphEVER_{rout}$ | **77.33** | **75.24** | 41.45 | 37.50 | **38.33** | 32.16 | 88.75 | 86.50 | **63.73** | **55.37** | 92.75 | 91.94 | 45.00 | 35.86 |
| Improve ↑ | +3.80% | +3.41% | +3.20% | +1.92% | +3.37% | +5.15% | +0.24% | +0.87% | +8.18% | +8.31% | +3.41% | +4.71% | +6.03% | +6.08% |

# 4 Experiment

## 4.1 Experimental Setup

*4.1.1 Datasets.* We evaluate our GraphEVER and baselines on seven publicly graph datasets in various fields, including (a) vision dataset: MSRC_9 [35], Letter-high [40], Letter-low [40], COIL-DEL [40], (b) social networks: COLLAB [56], (c) synthetic dataset: Synthie [34], and (d) bioinformatics dataset: ENZYMES [41].

In the implementation, training datasets are transformed into long-tailed formats with varying IFs to strictly adhere to Zipf's law, whereas the validation and test datasets are kept balanced.

*4.1.2 Compared Baselines.* To highlight the efficacy of our proposed approach, we benchmark the GraphEVER against several leading baselines. We categorize these baseline methods into four distinct groups for a comprehensive comparison: (a) Techniques for data re-balancing: up-sampling [3]; (b) Cost-sensitive learning methods: CB loss [6] and LACE loss [33]; (c) Information augmentation methods: graph augmentation [61] and $G^2GNN$ [53]; (d) Contrastive learning-based methods: graph contrastive learning (GraphCL) [60] and supervised contrastive learning (SupCon) [21].

For our GraphEVER, the detailed model and parameter settings is provided in the Appendix C.

## 4.2 Overall Evaluation

We assess the performance of GraphEVER along with baselines for long-tailed graph classification. Table 1 presents the results across seven benchmark datasets, each characterized by varying levels of IFs. From these findings, we derive the following insights:

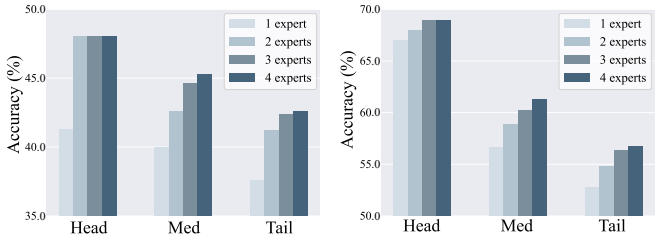
- An in-depth examination of accuracy across seven datasets indicates a significant decline in the performance of all methods as the imbalance in class distribution between head and tail grows. This observation points to a tendency for GNNs to experience a significant drop in effectiveness, leading to lower classification results in scenarios characterized by long-tailed distributions.
- Across the four groups of competitive baseline methods, it is generally observed that information augmentation strategies outperform re-balancing methods on the majority of datasets by leveraging extra information to enhance the representation of tail classes. However, it degrades severely in extreme imbalance situations, like the COIL-DEL dataset, which has an IF of 20 and over 60 classes with fewer than 3 training examples each. On the other hand, baselines based on contrastive learning tend to maintain consistent performance across different datasets.
- Overall, from the quantitative results, it can be observed that our framework GraphEVER and its variant $GraphEVER_{rout}$ achieve the best performance compared to other competitive baselines on all seven datasets with varying levels of class imbalance. In particular, $GraphEVER_{rout}$ outperforms the closest competitor on Letter-high with 8.18% with IF=25 and 8.31% with IF=50, which demonstrates the excellent capability of our framework for estimating the uncertainty and performing trustworthy long-tailed graph classification. However, it should be noted that the variant $GraphEVER_{rout}$ may result in performance degradation under particularly severe long-tailed scenarios as it dynamically adjusts the expert engagement for each sample.

**Table 2: Ablation study of classification accuracy (%) for various variants on the Letter-high and ENZYMES datasets.**

| | $\mathcal{L}_{mle}$ | $\mathcal{L}_{kl}$ | $\mathcal{L}_{div}$ | Aug | Letter-high | ENZYMES |
|---|---|---|---|---|---|---|
| $M_1$ | ✓ | | | | 62.70 | 35.67 |
| $M_2$ | ✓ | ✓ | | | 62.87 | 35.17 |
| $M_3$ | ✓ | | ✓ | | 62.98 | 36.00 |
| $M_4$ | ✓ | ✓ | ✓ | | 63.16 | 36.21 |
| $M_5$ | ✓ | ✓ | ✓ | ✓ | 63.40 | 38.17 |

## 4.3 Ablation and Sensitivity Studies

Here we further investigate the functionality of various components within GraphEVER via ablation studies. Additionally, we explore how GraphEVER's performance is influenced by changes in the hyper-parameter related to the number of experts, denoted as $M$.




(a) COIL-DEL (b) Letter-high

**Figure 3: Performance comparison for the head, medium and tail classes w.r.t. various configurations of expert numbers.**

*4.3.1 Functionality of Objective Components.* Given that the overall loss function of GraphEVER comprises two main components: $\mathcal{L}_{exp}$ ($\mathcal{L}_{mle}$ + $\mathcal{L}_{kl}$), and $\mathcal{L}_{div}$, conducting ablation studies is crucial to assess the impact of each individual component and the graph augmentation strategies (Aug) on overall effectiveness. Specifically, we explore four different variants designed as follows:

- $M_1$: Our base model, in which each expert is trained utilizing only the loss function $\mathcal{L}_{mle} = \sum_{i=1}^{N} \mathcal{L}_i$, as defined by Type II Maximum Likelihood Estimation in Eq. (13).
- $M_2$: It is a variant in which each expert is optimized utilizing both $\mathcal{L}_{mle}$ and the KL-divergence term $\mathcal{L}_{kl}$, *i.e.*, the total $\mathcal{L}_{exp}$.
- $M_3$: It is another variant, in which $\mathcal{L}_{mle}$ is combined with $\mathcal{L}_{div}$ in order to diversify each expert.
- $M_4$: The model that trains with the overall loss function of the joint training objective in our multi-expert framework.
- $M_5$: Our complete model, which combines all three objective components along with the graph augmentations.

We choose two representative datasets, Letter-high and ENZYMES to conduct the experiments. The results presented in Table 2 reveal that: (i) the incorporation of $\mathcal{L}_{kl}$ is beneficial in achieving more reliable uncertainty estimation, as it ensures that less evidence is generated for incorrect labels. (ii) The application of $\mathcal{L}_{div}$ in the optimization of the multi-expert framework leads to a remarkable improvement in overall performance, due to the diverse opinions formed by each individual expert. (iii) Moreover, the utilization of graph augmentations plays a crucial role in diversifying the views observed by each expert and encouraging discrepancy, leading to enhanced graph-level representations for classification.

*4.3.2 Influence of Expert Number.* We explore the effect of varying expert numbers on accuracy across head, medium, and tail class segments within the COIL-DEL and Letter-high datasets, represented in Figure 3. We first divide the classes into the head, medium, and tail based on the sample quantity, and the corresponding results are recorded as the count of experts $M$ increases. An initial boost in accuracy for head classes is noted with an increase in $M$ from 1 to 2, but additional increases in $M$ yield negligible benefits. For the medium and tail classes, leveraging more experts results in an overall improvement in performance. However, as the $M$ increases, the improvement becomes increasingly small. This observation supports the notion that easy samples (head classes) typically require fewer experts, suggesting an optimization potential by allocating experts more flexibly based on class complexity.

**Table 3: Comparison of computational expenses across various methods, focusing on total training duration (seconds) at model convergence, inference duration (seconds) over 1000 epochs, and GPU memory usage during training (MB).**

| Model | Train | Inference | GPU Memory | Accuracy |
|---|---|---|---|---|
| GraphSAGE | 445.61 | 353.58 | 1176 | 38.80 |
| Up-sampling | 2183.42 | 362.23 | 1178 | 39.20 |
| CB loss | 463.35 | 356.28 | 1176 | 41.72 |
| LACE loss | 466.92 | 354.32 | 1176 | 41.96 |
| Augmentation | 2823.64 | 375.23 | 1178 | 38.18 |
| $G^2GNN_n$ | 5212.39 | 373.57 | 1536 | 38.32 |
| $G^2GNN_e$ | 5432.08 | 394.07 | 1536 | 39.18 |
| GraphCL | 1188.12 | 369.23 | 1176 | 42.02 |
| SupCon | 1142.24 | 368.93 | 1176 | 42.93 |
| GraphEVER (1 expert) | 913.09 | 345.34 | 1178 | 41.64 |
| GraphEVER (2 experts) | 1087.77 | 364.46 | 1180 | 42.96 |
| GraphEVER (3 experts) | 1215.41 | 388.02 | 1182 | 45.52 |
| GraphEVER (4 experts) | 1385.73 | 419.96 | 1186 | 45.93 |
| GraphEVER (4 experts + router) | 1399.32 | 359.43 | 1186 | 45.57 |

## 4.4 Efficiency on Expert Engagement

Here we examine the computational efficiency of the proposed dynamic routing module and provide visualizations of the engagement of experts for the head, medium, and tail classes, to demonstrate that our dynamic routing module effectively makes performance and efficiency trade-offs.

*4.4.1 Computational Cost.* We assess the computational expenses of our proposed GraphEVER in comparison to other baseline methods, focusing specifically on the large-scale COIL-DEL dataset. We record the total training time when the model converges and the inference time of 1000 validation epochs in Table 3. To investigate the computational cost of the different numbers of experts, we create five variants of GraphEVER by varying the number of experts from 1 to 4. Additionally, we implement a dynamic router in the variant of GraphEVER that uses 4 experts. All baselines are trained and evaluated on a single NVIDIA A40 GPU. As is demonstrated in the table, our proposed GraphEVER has a lower computational cost in terms of total training time compared to prior data re-sampling (Up-sampling) and information augmentation (Augmentation, $G^2GNN$) methods, as these methods substantially increase the number of the input graphs through up-sampling. Moreover, $G^2GNN$ introduces extra time complexity in the kernel similarity computation stage and message passing in the $k$NN graphs. As the number of experts increases from 1 to 4, both the training time and inference time increase accordingly, and we also observe a significant improvement in accuracy when increasing the number of experts. Moreover, the dynamic router effectively reduces the inference time by dynamically decreasing the engagement of experts for easy samples, while resulting in only a minor decline in overall performance. This finding validates the superiority of our dynamic routing mechanism. The GPU memory usage across all methods remains relatively consistent, indicating that incorporating multiple classifier heads as experts introduces affordable additional GPU memory overhead.

*4.4.2 Visualization of Expert Engagement.* In Figure 4, we illustrate the distribution of expert usage among samples from head, medium, tail, and all class categories within the Letter-high dataset. We set

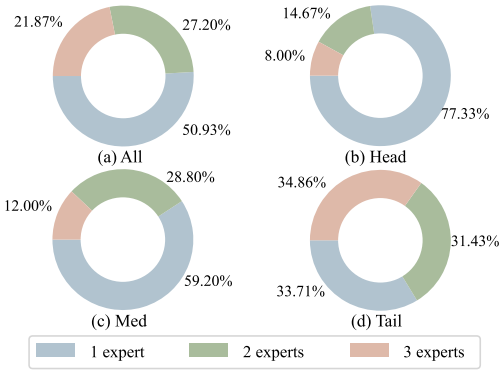
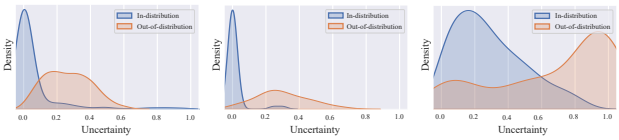



**Figure 4: Visualization of expert utilization on Letter-high dataset through pie charts, which indicate the proportion of samples that involve a particular number of experts.**




**(a) Letter-low** **(b) MSRC_9** **(c) ENZYMES**

**Figure 5: Uncertainty distribution measured by our model of in-/out-of-distribution samples.**

the maximum expert count to 3, and leverage the routing mechanism to assign experts to different samples dynamically. As depicted in the figure, over half of the samples require only one expert to make confident decisions, suggesting the practicality of minimizing expert involvement. Furthermore, instances in tail classes tend to require more experts, while the majority of instances in head classes only require the first expert. The results validate the efficiency of our dynamic routing module in reducing unnecessary expert deployment for simpler cases, thereby improving computational efficiency. Moreover, the dynamic router allocates more experts to unconfident samples, resulting in improved performance.

## 4.5 Uncertainty Visualization

Here we assess the reliability of the estimated uncertainty by visualizing its distribution. Additionally, we present representative examples of the Dirichlet distribution to investigate the difference between the head and tail samples.

*4.5.1 Uncertainty Estimation.* We assess the reliability of the estimated uncertainty by illustrating how in-distribution and out-of-distribution samples are distinguished through their uncertainty scores. In-distribution samples are considered as those from the original dataset, whereas out-of-distribution samples are created by introducing Gaussian noise to the test samples, using a fixed standard deviation ($\sigma$ = 10). This allows us to analyze the ability of the model to distinguish between in-distribution and out-of-distribution data. As depicted in Figure 5, the experimental results reveal the following observations: (i) Datasets with higher classification accuracy, such as Letter-low and MSRC_9, generally exhibit

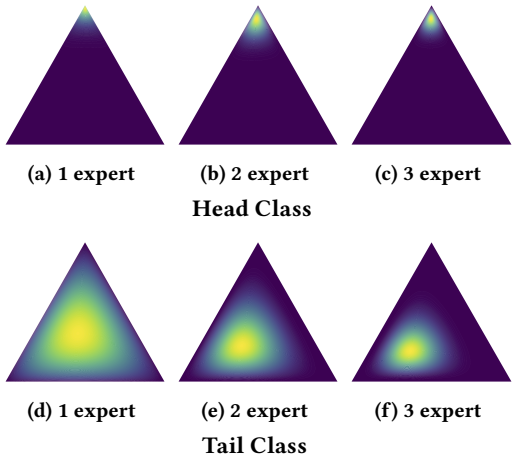



**Figure 6: Case study of Dirichlet distribution.**

less uncertainty for in-distribution samples, while datasets with lower classification performance display greater uncertainty for these samples. (ii) Across all datasets, we observe significantly higher uncertainties for out-of-distribution samples, indicating that our model effectively characterizes uncertainty, thereby facilitating more reliable and robust discrimination between in-distribution and out-of-distribution data.

*4.5.2 Case Study.* To vividly demonstrate the Dirichlet distribution of head and tail samples under a triple classification task, we present representative examples in Figure 6 using the COLLAB dataset. The visualization shows the Dirichlet distribution of head and tail samples with varying numbers of experts. It is noted that the Dirichlet distribution associated with the head sample presents a concentrated distribution, with the density focused at the peak of a standard 2-simplex. This indicates that sufficient evidence has been gathered for precise classification, resulting in low overall uncertainty. Moreover, it can be seen that a single expert is able to make confident decisions. In contrast, for tail classes, a single expert provides weak evidence, resulting in relatively high uncertainty and a flat distribution over the simplex. However, as more experts are engaged, the density of the distribution gradually shifts towards a specific direction, indicating that increasing expert engagement can improve the model's ability to form confident decisions when single-expert evidence is insufficient for uncertain tail samples.

## 5 Conclusion

In this work, we explore an under-explored setting which so-called long-tailed graph classification. We propose a principled framework GraphEVER, which is built upon the decision support of the multiple experts to incorporate diverse beliefs from an evidence theory, and the subjective logic offers a methodology for integrating evidence and uncertainty assessments for each expert within the framework of Dempster-Shafer Evidence Theory. Further, to draw more attention to the tail classes and reduce redundant experts for the head classes, we develop a routing mechanism to dynamically assign experts. Extensive experiments and visualizations on seven datasets demonstrate the effectiveness of our GraphEVER.

## Acknowledgments

This work is supported in part by the National Key Research and Development Program of China with Grant No. 2023YFC3341203, the National Natural Science Foundation of China under Grant 62276002, 62306014 and 12501344, Postdoctoral Fellowship Program (Grade A) of CPSF under Grant BX20250376 and BX20240239, China Postdoctoral Science Foundation under Grant 2024M762201, Sichuan Science and Technology Program under Grant 2025ZNSFSC1506 and 2025ZNSFSC0808, the Fundamental Research Funds for the Central Universities under Grant 1082204112K97, and Sichuan University Interdisciplinary Innovation Fund.

## References

[1] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems* 32 (2019).

[3] Nitesh V Chawla. 2003. C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML*, Vol. 3. CIBC Toronto, ON, Canada, 66.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[6] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9268–9277.

[7] Arthur P Dempster. 1967. Upper and lower probabilities induced by a multivalued mapping. In *The Annals of Mathematical Statistics,*. 325–339.

[8] Arthur P Dempster. 1968. A generalization of Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)* 30, 2 (1968), 205–232.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[10] Hao Guo and Song Wang. 2021. Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15089–15098.

[11] Zongbo Han, Changqing Zhang, Huazhu Fu, and Joey Tianyi Zhou. 2021. Trusted multi-view classification. *arXiv preprint arXiv:2102.02051* (2021).

[12] Zongbo Han, Changqing Zhang, Huazhu Fu, and Joey Tianyi Zhou. 2022. Trusted multi-view classification with dynamic evidential fusion. *IEEE transactions on pattern analysis and machine intelligence* 45, 2 (2022), 2551–2566.

[13] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5375–5384.

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

[15] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. 2024. A comprehensive survey on deep graph representation learning. *Neural Networks* (2024), 106207.

[16] Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Jianhao Shen, Ziyue Qiao, and Ming Zhang. 2025. Cluster-guided contrastive class-imbalanced graph Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 11924–11932.

[17] Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Yifan Wang, Xiao Luo, and Ming Zhang. 2024. Hypergraph-enhanced Dual Semi-supervised Graph Classification. In *Forty-first International Conference on Machine Learning*. 22594–22604.

[18] Wei Ju, Siyu Yi, Yifan Wang, Qingqing Long, Junyu Luo, Zhiping Xiao, and Ming Zhang. 2024. A survey of data-efficient graph learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. 8104–8113.

[19] Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. 2025. A Survey of Graph Neural Networks in Real world: Imbalance, Noise, Privacy and OOD Challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[20] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. 2003. Marginalized kernels between labeled graphs. In *Proceedings of international conference on machine learning*. 321–328.

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020), 18661–18673.

[22] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[23] Ryosuke Kojima, Shoichi Ishida, Masateru Ohta, Hiroaki Iwata, Teruki Honma, and Yasushi Okuno. 2020. kGCN: a graph-based deep learning framework for chemical structures. *Journal of Cheminformatics* 12, 1 (2020), 1–10.

[24] Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang. 2022. Trustworthy Long-Tailed Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6970–6979.

[25] Ke Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiangjun Dong, and Xinwang Liu. 2023. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering* 36, 1 (2023), 226–238.

[26] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. 2023. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1559–1568.

[27] Ke Liang, Lingyuan Meng, Yue Liu, Meng Liu, Wei Wei, Suyuan Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. 2024. Simple yet effective: Structure guided pre-trained transformer for multi-modal knowledge graph reasoning. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 1554–1563.

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

[29] Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. 2021. Tail-gnn: Tail-node graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1109–1119.

[30] Qingqing Long, Yuchen Yan, Wentao Cui, Wei Ju, Zhihong Zhu, Yuanchun Zhou, Xuezhi Wang, and Meng Xiao. 2024. MOAT: Graph Prompting for 3D Molecular Graphs. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1586–1596.

[31] Zhengyang Mao, Wei Ju, Yifang Qin, Xiao Luo, and Ming Zhang. 2023. RAHNet: Retrieval Augmented Hybrid Network for Long-tailed Graph Classification. In *Proceedings of the 31st ACM International Conference on Multimedia*. 3817–3826.

[32] Zhengyang Mao, Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Qingqing Long, Nan Yin, Xinwang Liu, and Ming Zhang. 2025. Learning Knowledge-diverse Experts for Long-tailed Graph Classification. *ACM Transactions on Knowledge Discovery from Data* 19, 2 (2025), 1–24.

[33] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2020. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314* (2020).

[34] Christopher Morris, Nils M Kriege, Kristian Kersting, and Petra Mutzel. 2016. Faster kernels for graphs with continuous attributes via hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1095–1100.

[35] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. 2016. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning* 102, 2 (2016), 209–245.

[36] Mark EJ Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary physics* 46, 5 (2005), 323–351.

[37] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32 (2019).

[38] Joonhyung Park, Jaeyun Song, and Eunho Yang. 2021. GraphENS: Neighbor-Aware Ego Network Synthesis for Class-Imbalanced Node Classification. In *International Conference on Learning Representations*.

[39] Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. 2021. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1390–1398.

[40] Kaspar Riesen and Horst Bunke. 2008. IAM graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 287–297.

[41] Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. 2004. BRENDA, the enzyme database: updates and major new developments. *Nucleic acids research* 32, suppl_1 (2004), D431–D433.

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

[43] Kari Sentz and Scott Ferson. 2002. Combination of evidence in Dempster-Shafer theory. (2002).

[44] Zhixiang Shen and Zhao Kang. 2025. When heterophily meets heterogeneous graphs: Latent graphs guided unsupervised representation learning. *IEEE Transactions on Neural Networks and Learning Systems* (2025).

[45] Zhixiang Shen, Shuo Wang, and Zhao Kang. 2024. Beyond redundancy: Information-aware unsupervised multiplex graph structure learning. *Advances in neural information processing systems* 37 (2024), 31629–31658.

[46] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* 12, 9 (2011), 2539–2561.

[47] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *Proceedings of International Conference on Artificial Intelligence and Statistics*. 488–495.

[48] Min Shi, Yufei Tang, Xingquan Zhu, David Wilson, and Jianxun Liu. 2020. Multi-class imbalanced graph convolutional network learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*.

[49] Jaeyun Song, Joonhyung Park, and Eunho Yang. 2022. TAM: Topology-Aware Margin Loss for Class-Imbalanced Node Classification. In *International Conference on Machine Learning*. PMLR, 20369–20383.

[50] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

[51] Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. 2025. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. *arXiv preprint arXiv:2502.02017* (2025).

[52] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. 2020. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809* (2020).

[53] Yu Wang, Yuying Zhao, Neil Shah, and Tyler Derr. 2022. Imbalanced graph classification via graph-of-graph neural networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2067–2076.

[54] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. *Advances in neural information processing systems* 30 (2017).

[55] Liuyu Xiang, Guiguang Ding, and Jungong Han. 2020. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision*. Springer, 247–263.

[56] Pinar Yanardag and SVN Vishwanathan. 2015. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1365–1374.

[57] Siyu Yi, Zhengyang Mao, Kangjie Zheng, Zhiping Xiao, Ziyue Qiao, Chong Chen, Xian-Sheng Hua, Yongdao Zhou, Ming Zhang, and Wei Ju. 2025. Learning Generalizable Contrastive Representations for Graph Zero-Shot Learning. *IEEE Transactions on Multimedia* (2025).

[58] Si-Yu Yi, Zhengyang Mao, Wei Ju, Yong-Dao Zhou, Luchen Liu, Xiao Luo, and Ming Zhang. 2023. Towards Long-Tailed Recognition for Graph Classification via Collaborative Experts. *IEEE Transactions on Big Data* (2023).

[59] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems* 31 (2018).

[60] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems* 33 (2020), 5812–5823.

[61] Shuo Yu, Huafei Huang, Minh N Dao, and Feng Xia. 2022. Graph augmentation learning. In *Companion Proceedings of the Web Conference 2022*. 1063–1072.

[62] Wang Yu-Hang, Junkang Guo, Aolei Liu, Kaihao Wang, Zaitong Wu, Zhenyu Liu, Wenfei Yin, and Jian Liu. 2025. TAET: Two-Stage Adversarial Equalization Training on Long-Tailed Distributions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 15476–15485.

[63] Sukwon Yun, Kibum Kim, Kanghoon Yoon, and Chanyoung Park. 2022. LTE4G: Long-Tail Experts for Graph Neural Networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2434–2443.

[64] Chongsheng Zhang, George Almpanidis, Gaojuan Fan, Binquan Deng, Yanbo Zhang, Ji Liu, Aouaidjia Kamel, Paolo Soda, and João Gama. 2025. A systematic review on long-tailed learning. *IEEE Transactions on Neural Networks and Learning Systems* (2025).

[65] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[66] Wenyi Zhao, Wei Li, Yuhan Li, Lu Yang, Zhenhao Liang, Enwen Hu, Weidong Zhang, and Huihua Yang. 2025. Constructing balanced training samples: a new perspective on long-tailed classification. *IEEE Transactions on Multimedia* (2025).

[67] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. 2020. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9719–9728.

# A Algorithm for our GraphEVER

**Algorithm 1:** Optimization Algorithm of GraphEVER

**Input:** Long-tailed graph dataset $\mathcal{G} = \{G_i, y_i\}_{i=1}^{N}$, number of experts $M$, number of classes $K$.

Initialize GNN classifier parameter $\theta$ and trainable weight matrices $\{\mathbf{W}^{(m)}\}_{m=1}^{M}$.

//***Stage 1***

**while** *not converged* **do**

  **for** $m = 1 : M$ **do**

    $\boldsymbol{e}^m \leftarrow$ the output of the $m$-th expert;

    $\boldsymbol{\alpha}^m \leftarrow \boldsymbol{e}^m + 1$;

    $S^m \leftarrow \sum_{k=1}^{K} \alpha_k^m$;

    $u^m \leftarrow K/S^m$;

    Compute the loss $\mathcal{L}^m$ of the $m$-th expert with Eq. (14);

  **end**

  Compute the overall loss with Eq. (17) and minimize it to update parameters $\theta$ by gradient descent;

**end**

//***Stage 2***

**while** *not converged* **do**

  Compute the routing loss with Eq. (19) and minimize it to update parameters $\{\mathbf{W}^{(m)}\}_{m=1}^{M}$ by gradient descent;

**end**

**Output:** The joint evidence $\boldsymbol{e}$ with Eq. (12) for inference.

We show the pseudo-code of our proposed GraphEVER in the Algorithm 1.

# B Related Work

## B.1 Graph-level Classification

The development of graph representation learning algorithms [19, 26, 51] opens great opportunities for graph analysis. One core problem in the graph domain is graph-level classification, which involves predicting the category of the whole graph. Compared to node-level classification [27, 44, 45, 57] which targets the properties of individual nodes in a graph. Graph-level classification focuses on extracting more comprehensive representations of the graphs to make a better prediction. Early studies develop graph kernel methods [20, 46, 47] to solve the problem. Recently, graph neural networks [16, 31, 32, 58] emerge as the promising methods for this task, which have achieved unprecedented success in identifying categorical labels of graphs. To step further, we study a challenging scenario where the data are long-tailed, and we are the first to incorporate uncertainty estimation into this task.

## B.2 Long-Tailed Recognition

Class-imbalanced learning (also known as long-tailed recognition) is studied actively in the vision domain. Existing approaches can be broadly categorized into re-sampling [4, 10, 66], re-weighting [2, 6, 54, 62], and ensembling learning [24, 52, 55]. Re-sampling methods seek to achieve a balanced distribution where tail classes are oversampled, whereas head classes are undersampled. Re-weighting methods allocate different importance weights on different classes to adjust the portion of training loss. The first two strategies typically put more emphasis on the tail classes and improve overall performance at the expense of the head classes. Recently, ensembling learning methods have empirically shown stronger generalization, which combine multiple classifiers in a multi-expert framework. However, this strategy is typically prone to induce redundant computational resources.

Recently, diverse research endeavors has been dedicated to mitigating long-tail recognition challenges in graph-based tasks [28, 29, 38, 39, 48, 49, 63], which have achieved promising achievements. The main idea of these approaches is to either design degree-specific transformations on nodes or utilize structural features. Distinct from these methods that study long-tailed node-level classification, our proposed framework GraphEVER goes further and concentrates on under-explored and promising long-tailed graph-level classification, and innovatively explores this task from the perspective of predictive uncertainty.

## B.3 Evidence Theory

The mathematical foundation known as the Dempster-Shafer Evidence Theory (DST) is introduced by Dempster to address uncertainty reasoning [7]. Acting as an extension of the Bayesian theory to subjective probabilities [8], DST provides the opinion of both imprecision and uncertainty. It has been proposed as a more flexible and general approach than the Bayesian one. It stands out for combining the measures of evidence of multiple sources [11, 12, 43]. Benefits from its theoretical guarantee and flexible capability, our proposed work introduces this technique to accurately provide uncertainty estimation and predictive confidence directly.

# C Model Settings

In our assessment of the GraphEVER and the various baselines, we employ GraphSAGE as the foundational GNN encoder and adjust the embedding size to 64. Adam optimizer is adopted for optimizing all models, setting the batch size to 32 and the learning rate to 0.0001. For our GraphEVER, the expert count $M$ is set to 3 for performance and efficiency trade-off. Moreover, we adjust the hyper-parameters of annealing epoch $T$ and top choices number $s$ for each dataset. The re-weighting hyper-parameter $\beta_{CB}$ for the CB loss is set to 0.99 and the scaling temperature $\tau_{LACE}$ for the LACE loss is set to 1.0. Additionally, we fix the number of training epochs at 1000, and we implement the early stopping technique in the encoder training process with a patience of 500 epochs. In our experimental study, we employ accuracy as the standard metric to assess performance.

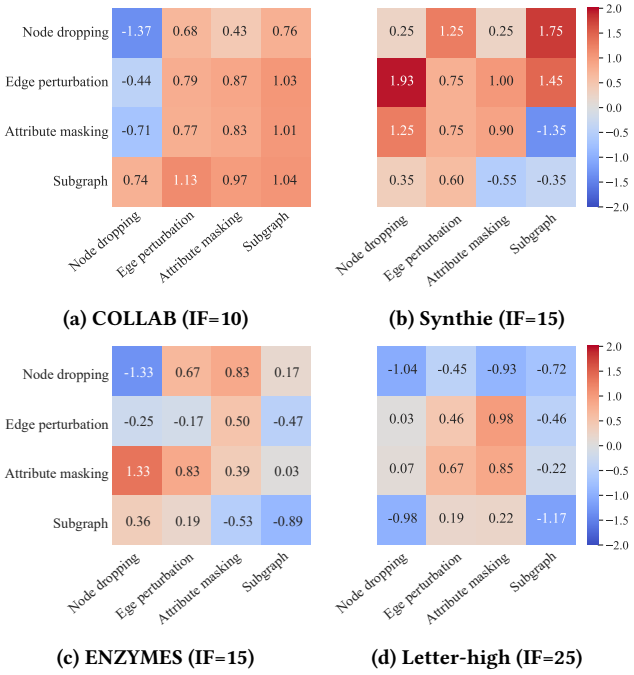
# D Generalization beyond graph domain

To verify whether our proposed framework maintains its effectiveness beyond the graph domain, we evaluate our method on the well-known MNIST dataset from the computer vision field, with results presented in Table 4. As can be seen, our proposed method, GraphEVER and its variants GraphEVER$_{rout}$, consistently outperform the baseline methods by a significant margin. This not only demonstrates the effectiveness of our approach in long-tail graph classification tasks but also shows that even when transferred to other domains, its performance remains competitive, highlighting the generalizability of our technical framework.

**Table 4: Accuracy (%) on the MNIST datasets, varying the degree of imbalance by two IFs. The top classification results are highlighted in boldface and the second-best results are indicated with an underline.**

| | GraphSAGE | Up-sampling | CB loss | LACE loss | Augmentation | $G^2GNN_n$ | $G^2GNN_e$ | GraphCL | SupCon | GraphEVER | GraphEVER$_{rout}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| IF=50 | 68.67 | 64.69 | 68.85 | 69.72 | 69.37 | 69.76 | 72.18 | 70.91 | <u>73.69</u> | 75.38 | **75.62** |
| IF=100 | 63.46 | 59.78 | 63.40 | 64.59 | 65.12 | 64.88 | 68.17 | 66.73 | <u>70.31</u> | 72.71 | **72.93** |

## E Analysis of Graph Augmentations




**(a) COLLAB (IF=10)** **(b) Synthie (IF=15)**

**(c) ENZYMES (IF=15)** **(d) Letter-high (IF=25)**

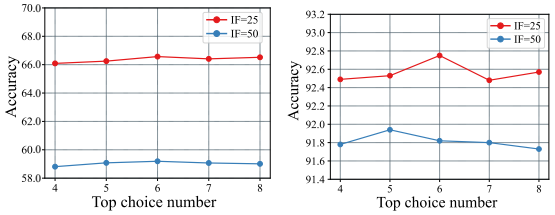
**Figure 7: Long-tailed graph classification accuracy gain (%) when using different graph augmentations. The accuracies of baselines training with no augmentations are 76.00%, 39.70%, 36.84%, 62.42% for the four datasets respectively.**

In this section, we analyze the effectiveness of different types of graph augmentations on various datasets.

- **Node dropping**. It involves randomly removing a certain proportion of nodes and all connected edges, preserving semantics. The dropping probability follows an i.i.d. uniform distribution.
- **Edge perturbation**. It involves randomly adding or deleting a certain ratio of edges to alter the connectivity pattern.
- **Attribute masking**. It involves randomly selecting certain nodes and masking some of their attributes.
- **Subgraph**: It involves utilizing the random walk algorithm to extract a representative subgraph from the original graph, assuming that the semantic meaning of the graph is preserved.

We first set the number of experts to 3, and then adopt different types of augmentations for the input of the second and the third experts. The augmentation ratios for the second and the third expert are set to 0.05 and 0.1, respectively. In Figure 7, we demonstrate the accuracy of long-tailed learning gains when using different pairs of graph augmentations, where the warmer colors indicate better performance gains. As can be seen from the figure, using appropriate graph augmentations can benefit the classification performance significantly, indicating that graph augmentations improve model generalization by further diversifying and enriching the training data. It should be noted that the appropriate augmentation pairs vary among different datasets.

## F Sensitivity Analysis




**(a) Letter-high** **(b) Letter-low**

**Figure 8: Variations of the long-tailed graph classification accuracy of the GraphEVER under different settings of $s$.**

The hyper-parameter of the top choice number $s$ need to be determined for the router training in GraphEVER. This section studies the parameter sensitivity of the method GraphEVER to parameter variations of $s$ in terms of the long-tailed graph classification accuracy. The experiments are conducted on the Letter-high and Letter-low datasets under different imbalance settings. Figure 8 shows that the long-tailed graph classification accuracy of GraphEVER is not sensitive to parameter variations on the Letter-high dataset. For the Letter-low dataset, the accuracy under different values of $s$ differs by no more than 0.4%. This indicates that the competitive performance of GraphEVER can be easily obtained by fine-tuning the hyper-parameters over a limited search space.