**ROG\_PL**: Robust Open-Set Graph Learning via Region-Based Prototype Learning

Qin Zhang\(^1\), Xiaowei Li\(^1\), Jiexin Lu\(^1\), Liping Qiu\(^1\), Shirui Pan\(^2\), Xiaojun Chen\(^1*\), Junyang Chen\(^1\)

\(^1\)College of Computer Science and Software Engineering, Shenzhen University, China \(^2\)School of Information and Communication Technology, Griffith University, Australia.

qinzhang@szu.edu.cn, \{lixiaowei2022, lujiexin2022, qiuiping2021\}@email.szu.edu.cn, s.pan@griffith.edu.au, \{xjchen, junyangchen\}@szu.edu.cn

**Abstract**

Open-set graph learning is a practical task that aims to classify the known class nodes and to identify unknown class samples as unknowns. Conventional node classification methods usually perform unsatisfactorily in open-set scenarios due to the complex data they encounter, such as out-of-distribution (OOD) data and in-distribution (IND) noise. OOD data are samples that do not belong to any known classes. They are outliers if they occur in training (OOD noise), and open-set samples if they occur in testing. IND noise are training samples which are assigned incorrect labels. The existence of IND noise and OOD noise is prevalent, which usually cause the ambiguity problem, including the *intra-class variety problem* and the *inter-class confusion problem*. Thus, to explore robust open-set learning methods is necessary and difficult, and it becomes even more difficult for non-IID graph data. To this end, we propose a unified framework named ROG\_PL to achieve robust open-set learning on complex noisy graph data, by introducing prototype learning. In specific, ROG\_PL consists of two modules, *i.e.*, denoising via label propagation and open-set prototype learning via regions. The first module corrects noisy labels through similarity-based label propagation and removes low-confidence samples, to solve the intra-class variety problem caused by noise. The second module learns open-set prototypes for each known class via non-overlapped regions and remains both interior and border prototypes to remedy the inter-class confusion problem. The two modules are iteratively updated under the constraints of classification loss and prototype diversity loss. To the best of our knowledge, the proposed ROG\_PL is the first robust open-set node classification method for graph data with complex noise. Experimental evaluations of ROG\_PL on several benchmark graph datasets demonstrate that it has good performance.

**Introduction**

Graph neural networks (GNNs) (Gilmer et al. 2017; Guo et al. 2022; Hamilton, Ying, and Leskovec 2017; Tan et al. 2023) have become a prominent technique to analyze graph structured data in many real-world systems, such as traffic state prediction (Zheng et al. 2020), disease classification (Chereda et al. 2019), and user profile completion in social networks (Wong et al. 2021). The recent success of supervised GNNs is built upon two crucial cornerstones: that the training and test data are drawn from an *identical distribution*, and that large-scale reliable *high-quality labeled* data are available for training. However, in real-world applications, large-scale labeled data drawn from the same distribution as test data are usually unavailable.

Real-world applications normally are in open-set scenarios (Zhang et al. 2023a; Nimah et al. 2021; Zhang et al. 2022), the existence of new-emerged out-of-distribution (OOD) samples, *i.e.*, samples that do not belong to any known classes, is prevalent. They are outliers if they occur in training (OOD noise), and open-set samples if they occur in testing. Moreover, the manually generating clean labeled data set would involve domain experts evaluating the quality of collected data and thus is very expensive and time-consuming (Zhang, Luo, and Gu 2023). Alternatively, we can collect data and labels based on web search (Yu et al. 2018), crowdsourcing (Fang, Yin, and Tao 2014; Li et al. 2017a) and user tags (Li et al. 2017b; Xiao et al. 2015). These data and labels are cheap but inevitably noisy (Zhang, Luo, and Gu 2023; Li et al. 2022).

The presence of OOD noise and IND noise (in-distribution samples with incorrect labels) can be detrimental to GNNs (Zhang et al. 2021), as they would cause the ambiguity problem, including the *intra-class variety problem* in which samples with same class labels may contain objects of different semantic categories, and the *inter-class confusion problem* in which samples of different intent classes may contain objects of similar semantic categories (Wang et al. 2023). The open-set unknown class samples occur during testing further complicates this problem.

Thus, it is necessary to build robust open-set learning models (Wu et al. 2021) that can learn from noisy data, achieving classification of known class samples and identification of unknown open-set class samples during testing. Previous works mainly concentrated on robust learning or open-set learning separately, the problem of robust open-set learning with complex noise has not been sufficiently explored so far (Huang, Wang, and Fang 2022; Zhang et al. 2023b), and it becomes even more difficult for non-IID graph data.

In this paper, as shown in Fig. 1, for solving the problem of open-set graph learning with complex IND and OOD...
noise, we propose a new framework ROGPL to do robust open-set node classification. It consists of two main steps: denoising via label propagation and open-set prototype learning via regions. Specifically, in the latent representation space, ROGPL first corrects noisy labels through similarity-based pseudo-label propagation and removes low-confidence samples, to solve the intra-class variety problem caused by noise. Then we learn open-set prototypes for each known class via non-overlapped regions and remains both interior and border prototypes to remedy the inter-class confusion problem. These two modules are iteratively updated under the constraints of classification loss and prototype diversity loss.

Figure 1: Overview of the proposed ROGPL. In the latent representation space, two modules are designed: denoising via label propagation and open-set prototype learning via regions. In specific, we first correct noisy labels through similarity-based label propagation and removes low-confidence samples, to solve the intra-class variety problem caused by noise. Then we learn open-set prototypes for each known class via non-overlapped regions and remains both interior and border prototypes to remedy the inter-class confusion problem. These two modules are iteratively updated under the constraints of classification loss and prototype diversity loss.

Preliminaries

This study focuses on the node classification problem for a graph. A graph is denoted as $G = (V, E, X)$, where $V = \{v_i | i = 1, \ldots, N\}$ is a set of $N$ nodes in the graph, and $E = \{e_{i,j} | i, j = 1, \ldots, N, i \neq j\}$ is a set of edges connecting pairs of nodes $v_i$ and $v_j$. $X \in \mathbb{R}^{N \times s}$ denotes the feature matrix of nodes, where $s$ is the dimension of node features. The feature vector of each node $v_i$ is indicated by $x_i \in X$. The topological structure of $G$ is represented by an adjacency matrix $A \in \mathbb{R}^{N \times N}$, where $A_{i,j} = 1$ if the nodes $v_i$ and $v_j$ are connected, i.e., $e_{i,j} \in E$, and otherwise $A_{i,j} = 0$. The label matrix of $G$ is $Y \in \mathbb{R}^{N \times C}$, where $C$ is the already-known node classes. If a label is assigned to a node $v_i \in V$, then $y_{i,c} = 1$, and otherwise, $y_{i,c} = 0$.

For a typical closed-set node classification problem, a GNN encoder $f_\theta$ takes the node features $X$ and adjacency matrix $A$ as input, aggregates the neighborhood information and outputs representations. Then, a classifier $f_\phi$ is used to classify the nodes into $C$ already-known classes. The GNN encoder and the classifier are optimized to minimize the expected risk (Yu et al. 2017) in Eq. (1), assuming that the testing data $D_{te}$ and the training data $D_{tr}$ have the same feature space and label space, i.e.,

$$f^* = \arg \min_{f \in \mathcal{H}} \mathbb{E}_{(x,y) \sim D_{te}} I(y \neq f(\theta, \phi; x, A))$$

where $\mathcal{H}$ is the hypothesis space, $I(\cdot)$ is the indicator function which outputs 1 if the expression holds and 0 otherwise. This function can be generally optimized using cross-entropy to distinguish between known classes.

In the open-set node classification problem, given a graph $G = (V, A, X)$, $D_{tr} = (X, Y)$ denotes the training nodes. The test nodes are denoted by $D_{te} = (X_{te}, Y_{te})$, where $X_{te} = S \cup U$, $Y_{te} = \{1, \ldots, C, C + 1, \ldots\}$. The set $S$ is the nodes that belong to seen classes that already appeared in $D_{tr}$ and $U$ is the set of nodes that do not belong to any seen class (i.e., unknown class nodes). The goal of open-set node classification is to learn a $(C + 1)$-class classifier $f_\theta$, such that $f(\theta, \bar{y}; X_{te}, \bar{A}) : \{X_{te}, \bar{A}\} \rightarrow \overline{\mathcal{Y}}$, $\overline{\mathcal{Y}} = \{1, \ldots, C, \text{unknown}\}$, by minimizing the expected risk (Yu et al. 2017):

$$\bar{f}^* = \arg \min_{f \in \mathcal{H}} \mathbb{E}_{(x,y) \sim D_{te}} I(y \neq f(\theta, \bar{y}; x, \bar{A}))$$

where $\bar{A}$ is the adjacency matrix for $X_{te}$. The predicted class $\text{unknown} \in \overline{\mathcal{Y}}$ consists of a group of novel categories, which may contain multiple classes. An intuitive way of transforming a close-set classifier into an open-set classifier is thresholding (Hendrycks and Gimpel 2017).

In the problem of Open-set node classification with IND noise and OOD noise, given a graph $G = (V, A, X)$, $D_{tr} = (X, Y = S) = \{(x_i, y_i)\}_{i=1}^N$ is the training set and $D_{te} = (X_{te}, Y_{te} = S \cup U)$ is the test set. In $D_{tr}$, we assume that the instance-label pair $(x_i, y_i), 1 \leq i \leq N$, consists of three types. Let $y_i^g$ denote the ground-truth label of $x_i$. A clean sample is a node whose assigned label matches the ground-truth label, i.e., $y_i = y_i^g$. An IND noise sample is a node whose assigned label does not match the ground-truth label, but the node matches one of the classes in $S$, i.e., $y_i \neq y_i^g, y_i^g \notin S$. An OOD noise sample is a node whose assigned label does not match the ground-truth label and any known class label neither, i.e., $y_i \neq y_i^g, y_i^g \notin S$. Moreover, under the strict setting of open-set classification, it is assumed that the ground-truth label of OOD noise $y_i^g \notin U$, i.e., there is no overlap between the classes of OOD noise.
and the unknown classes in the test set. Due to the memorization effect (Arpit et al. 2017), noisy data can severely impair the performance of network training. Therefore, it is desirable to develop noise-robust methods for open-set node classification, which can handle complex and diverse noises. The ultimate goal is to learn a noise-robust open-set node classifier that can minimize the expected risk in Eq. (2).

**Methodology**

For solving the problem of open-set graph learning with complex IND and OOD noise, we propose a new framework ROG∗PL, which consists of two main steps: denoising via label propagation and open-set prototype learning via regions. As shown in Fig. 1, in the latent representation space, we first corrects noisy labels through soft pseudo-label propagation and removes low-confidence samples, to solve the intra-class variety problem caused by noise. Then we learn open-set prototypes for each known class via non-overlapped regions and remains both interior and border prototypes to remedy the inter-class confusion problem. The two modules are iteratively updated under the constraints of both classification loss and prototype diversity loss.

**Denoising via Label Propagation**

Inspired by label propagation algorithms for semi-supervised learning (Grandvalet and Bengio 2004; Iscen et al. 2018; Chandra and Kokkinos 2016), which seek to transfer labels from supervised examples to neighboring un-supervised examples according to their similarity in feature space, we leverage neighbour consistency to modify the supervision of each sample and to correct noise. First, we build a k-nearest neighbor graph GN upon Z, where Z ∈ R|N|×|K| is the latent representation given by an encoder network f0 by taking the nodes feature X, i.e. zi = f0(x), K indicates the dimension of the representation vectors and is determined by the network f0.

In graph GN, the affinity matrix W ∈ R|N|×|N| is encoded by the similarities between vertices, which is obtained by:

\[ W_{ij} = \begin{cases} \left[ z_i^T z_j \right]^{\beta}, & \text{if } i \neq j \land x_j \in N_k(x_i), \\ 0, & \text{otherwise}. \end{cases} \tag{3} \]

where N_k(x_i) is a similarity-based neighborhood set, i.e. the set of k nearest neighbors of x_i in X, and β is a parameter for diffusion on region manifolds (Chandra and Kokkinos 2016; Iscen et al. 2017) which we simply set as β in our experiments.

Suppose we have clean nodes along with some noisy nodes, label propagation spreads the label information of each node to the other nodes based on the connectivity in the graph GN. To weaken the influence of noisy labels, we set \( \hat{y}_i = y_i \) to the one-hot label vector of x_i if x_i is selected as a clean sample by Eq.(5), otherwise we use category prediction which is a C-dimensional vector representing the belongingness of x_i to the C known classes, i.e. \( \hat{y}_i = \tilde{p}_i \), as shown in Eq. (8). The propagation process is repeated until a global equilibrium state is achieved, and each example is assigned to the class from which it has received the most information.

Formally, for graph GN = (W, X), D is the degree matrix (a diagonal matrix with entries \( D_{ii} = \sum_j W_{ij} \)), label propagation (Iscen et al. 2019) can be computed by minimizing

\[ J(\bar{Y}) = \sum_{i=1}^{N} \| \bar{y}_i - \hat{y}_i \| + \alpha \sum_{i,j=1}^{N} W_{ij} \frac{1}{\sqrt{D_{ii}}} \bar{y}_i - \frac{1}{\sqrt{D_{jj}}} \bar{y}_j \| \|^2 \tag{4} \]

where α is a regularization parameter to balance the fitting constraint (the first term) and the smoothing term (the second term). The fitting constraint encourages the classification of each node to their assigned label, and the smoothing term encourages the outputs of nearby points in the graph to be similar (Iscen et al. 2019). The obtained \( \bar{Y} = [\hat{y}_1, \ldots, \hat{y}_N] \in \mathbb{R}^{N \times C} \) is the refined soft pseudo-labels for X after label propagation, further we transform \( \bar{Y} \) into hard pseudo-labels by taking the largest prediction score to guide the training.

Finally, we use a sufficiently high threshold \( \eta \in [0, 1] \) to select a reliable subset of nodes as the clean dataset:

\[ g_i = \begin{cases} 1, & \text{if } \bar{Y}_{iy_i} > \frac{1}{C}, \\ \|\max_c \bar{Y}_{ic} > \eta\|, & \text{otherwise}. \end{cases} \tag{5} \]

where \( \|\cdot\| \) is the indicator function which outputs 1 if the expression holds and 0 otherwise. t denotes the number of iteration rounds. \( g_i \) is a binary indicator representing the conservation of node \( v_i \in V \) when \( g_i = 1 \) and the removal of node \( v_i \) when \( g_i = 0 \). Thus, the clean node set \( V_{cln} = \{ v_i \mid v_i \in V \land g_i = 1 \} = V \setminus \{ v_i \mid v_i \in V \land g_i = 0 \} \).  

**Open-Set Prototype Learning via Regions**

Overview. With the filtered training data (clean node set \( V_{cln} \), different from the conventional classification paradigm which directly feeds node features into a GNN to predict the class label, we aim at learning multiple representative prototypes for each category, and predicting class label by calculate the similarity between the node and prototypes. Upon the latent representation Z, in the latent representation space, we learn the prototypes. The prototype pool can be represented as \( P = \{ P_1, P_2, \ldots, P_c \} \) where C is the number of known classes. \( P_c = \{ p_{c,1}, p_{c,2}, \cdots, p_{c,K_c} \} \in \mathbb{R}^{K_c \times D} \) denotes the prototypes of category c and K_c indicates the number of prototypes of category c. Given a node x_i, after obtaining the high-level feature representation z_i, we compare it with all prototypes via calculating cosine similarity:

\[ s_{c,k}^{(i)} = \frac{z_i \cdot p_{c,k}}{\|z_i\| \|p_{c,k}\|} \tag{6} \]

After obtain similarity with all prototypes, we regard the class-wise largest similarity score to obtain the scores of x_i belonging to class c, i.e.,

\[ s_c^{(i)} = \max_k (s_{c,k}^{(i)}) = \max_k (s_{c,1}^{(i)}, s_{c,2}^{(i)}, \ldots, s_{c,K_c}) \tag{7} \]

And the vector of score of belonging between x_i and each class is
\[ \hat{p}_i = [s_1^{(i)}, s_2^{(i)}, \ldots, s_C^{(i)}] \]  
\[ \tilde{y}_i = \arg \max_c \hat{p}_{i,c} = \arg \max_c s_c^{(i)} \]  
\[ P_t = [P_{c,1}^t, \ldots, P_{c,C}^t] \in \mathbb{R}^{C \times D} \]  
\[ P_{c,t}^t = P_{c,t-1}^t - \varphi \frac{\partial \mathcal{L}}{\partial P_{c,t}^t} \]

The final classification prediction of \( x_i \) is

\[ \hat{y}_i = \arg \max_c \hat{p}_{i,c} = \arg \max_c s_c^{(i)} \]  

**Open-Set Prototype Learning.** The open-set node classification is different with the closed set classification problem, in which the class boundaries of known classes require to be tight and clear. Traditional prototype (Yang et al. 2018) learning normally use typical interior prototypes (such as the mean vectors of node representation in the same class) to represent the class. However, we argue that border prototypes are also crucial for classification, especially for open-set scenarios, since they provide more detailed information for preserving discrimination between classes and the information of class boundaries. We believe the combination of interior prototypes and border prototypes can well relieve the intra-class variety problem caused by the noise and inter-class confusion problem, and obtain tight and clear boundaries for known class, reserve more space for unknown classes.

Since the original training data containing IND and OOD noises, to avoid the ambiguity problem caused by noise data, we optimize the prototype learning based on the clean nodes \( V_{cls} \) and the refined label annotations \( Y \). We first divide the latent space into regions by clustering the clean nodes where any clustering algorithm could be used, and we use K-means for its simplicity. With the obtained clusters, under the guidance of label matrix \( Y \), we use homogeneous clusters to update the interior prototypes and use non-homogeneous clusters to obtain border prototypes.

To obtain the most representative interior prototype for each class, we make interior prototypes as trainable weights for each class, \( \hat{p}_c \) consists of samples of two classes, i.e., \( M_k = V_k^1 \cup V_k^2 \cup \ldots \cup V_k^C \), where \( V_k^c \) consists of a couple of nodes from class \( c \), and there exists \( m, n \in \{1, \ldots, C\} \) such that \( V_m^k \neq \emptyset \land V_n^k \neq \emptyset \), then we can obtain border prototypes of class \( m \) and \( n \) by:

\[ P_m^k = \frac{1}{||V_m^k||} \sum_{i \in V_m^k} z_i, \quad P_n^k = \frac{1}{||V_n^k||} \sum_{i \in V_n^k} z_i \]  

It is the same for clusters that contain nodes from several different classes.

We utilize cross-entropy loss to train the encoder network \( f_\theta \) and prototype network \( f_\phi \) on clean nodes \( V_{cls} \), under the supervision of refined class label \( \hat{Y} \):

\[ \mathcal{L}_{cls} = -\frac{1}{N} \sum_{i=1}^{N} \hat{y}_i \log \frac{\exp(s_{\hat{y}_i}^{(i)}/T)}{\sum_{c=1}^{C} \exp(s_c^{(i)}/T)} \]  

where \( T \) is a temperature hyperparameter that we introduced to make the results more differentiated (Agarwala et al. 2020).

With the adoption of both interior and border prototypes, the diversity of prototypes within a class can be well mined. To further relieve the inter-class confusion problem, we hope the prototypes of different categories also away from each other. Thus, we enhance the diversity of the prototypes of known classes by adopting the orthogonal constraint to keep the orthogonality of interior prototypes by using the diversity loss:

\[ \mathcal{L}_{div} = \| P_t P_t^\top - I \|_F^2 \]  

where \( \| \cdot \|_F \) is the Frobenius-norm and \( I \) is the identity matrix of any desired dimension. The overall loss function of ROG\(_{PL} \) is:

\[ \mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{div} \]  

where \( \lambda \) is the loss hyper-parameter.

So far, we obtain interior prototype and border prototypes for each known class, and we predict the class label of clean nodes through Eq. (7) and (9). To make a hard prediction, we adopt a probability threshold \( \tau \), such that a testing point \( x_i \) is deemed as an unknown class sample if \( \max_c \hat{p}_{i,c} < \tau \).

**Experiments**

We design our experiments to evaluate ROG\(_{PL} \), focusing on the following aspects: open-set classification comparison, robustness analysis, and ablative study. Codes will be available online.

**Experimental Setup**

**Dataset and Metrics.** To evaluate the performance of the proposed framework for robust open-set node classification, We conducted experiments on three main benchmark graph datasets (Wu, Pan, and Zhu 2020; Zhu et al. 2022), namely Cora\(^1 \), Citeseer\(^2 \) (Yang, Cohen, and Salakhudinov 2016),

\(^1\)https://graphsandnetworks.com/the-cora-dataset/

\(^2\)https://networkrepository.com/citeseer.php
and Coauthor-CS\textsuperscript{3} (Zhou et al. 2023), which are widely used citation network datasets. The statistics of the datasets are presented in the Appendix. In terms of the metrics, following the study of Wu et al. (Wu et al. 2021), we adopt macro-F1 and AUROC to measure the performance.

**Implementation Details.** In the experiments, we adopt GCN (Kipf and Welling 2016) as backbone neural network for the encoder $f_\theta$, configured with two hidden layers with a dimension of 128. The prototype network $f_\phi$ is configured with a linear layer with a dimension of $C + 1$. ROG_{PL} is implemented with PyTorch and the networks are optimized using adaptive moment estimation with a learning rate of $10^{-3}$. The balance parameters $\lambda$ is set to $10^{-2}$. The threshold $\eta$ were selected by a grid search in the range from 0 to 1 with a step of $10^{-1}$. The number of nearest neighbors $k$ were selected by a grid search in the range from 30 to 35 with a step of 1.

For each experiment, the baselines and the proposed method were applied on the same training, validation, and testing datasets. All the experiments were conducted on a workstation equipped with an Intel(R) Xeon(R) Gold 6226R CPU and an Nvidia A100 GPU.

**Test Settings.** To evaluate the performance of open-set node classification, for each dataset, the data of several classes were held out as the unknown classes for testing and the remaining classes were considered as the known classes. 70% of the known class nodes were sampled for training, 10% for validation, and 20% for testing.

To assess the performance of the proposed ROG\textsubscript{PL} framework on graph data with different noises, we tested it with mixed IND and OOD noise. In the training set, we randomly selected 5%/25%/50% of the known class samples to be IND noise, and randomly replaced their ground-truth labels with wrong known class labels. Then, we used the nodes from neither the known classes of the training set nor the unknown classes of the test set as OOD noise, and randomly assigned known class labels to them. The samples with right known class labels (clean data), the known class samples with incorrect labels (IND noise) and the far-away unknown class samples with wrong known class labels (OOD noise) constitute the final training set. The test set includes the known class samples and the unknown class samples, where the unknown classes are different from the classes of OOD noise. The setting of inductive learning was adopted for all the experiments, where no information about the unknown class in the test set (such as the feature $x_i$ or other side information of unknown classes) is utilized during training or evaluation.

**Baselines.** We compare ROG\textsubscript{PL} with 10 baselines, which are from three categories.

- **1) Closed-set classification methods:** GCN\textsubscript{soft} and GCN\textsubscript{sig}. They are GCNs (Kipf and Welling 2016) with a softmax layer or a multiple 1-vs-rest of sigmoids layer as output layer.

- **2) Open-set classification methods:** GCN\textsubscript{soft}, $\tau$, GCN\textsubscript{sig}, $\tau$, NWR, $\tau$ (Tang, Yang, and Li 2022).

- **3) Robust open-set classification methods:** NGC (Wu et al. 2021) and PNP (Sun et al. 2022). NGC (Wu et al. 2021) is an open-world noisy data learning method for image classification which employs geometric structure and model predictive confidence to collect clean sample. PNP, $\tau$ (Sun et al. 2022) is a robust classifier learning method for image data with IND and OOD noise, where data augmentation is used to help the identification of noisy samples. To perform open-set recognition, we adopt a threshold chosen from $\{0.1, 0.2, \ldots, 0.9\}$ to PNP.

Note that the graph data are first embedded by a GCN before being feed into the models that cannot handle graph data. A detailed introduction can be found in the Appendix.

**Open-Set Node Classification with Complex Noisy Graph Data**

Considering that real-world scenarios are complex and noisy data vary across different tasks, we assessed the proposed model for open-set classification with IND noise and two types of OOD data: near OOD data and far OOD data. Here, OOD data include OOD noise in the training set and out-of-distribution samples of unknown classes that occur during testing.

**Open-Set Classification with IND Noise and Near OOD Data.** In this experiment, for each dataset, following the setting of OpenWGL (Wu, Pan, and Zhu 2020), the data of the last class were held out as the unknown class for testing, and the data of the second last class which were re-assigned with random known class labels were set as the OOD noise data and injected into the training set. The remaining classes were considered as known classes, while the known class samples in training set were re-assigned with wrong known class labels with a rate of 5%, 25% and 50% (i.e. IND noise rate), respectively.

Table 1 lists the macro-F1 and AUROC scores for open-set node classification with near OOD noise and different proportions of IND noise. It is observed that ROG\textsubscript{PL} generally obtain the best performance on the benchmarks. This shows that ROG\textsubscript{PL} can better distinguish between a known class and an unknown class, even though there is a large amount of complex and diverse noises during training. Specifically, ROG\textsubscript{PL} achieves an average of 6.23% improvement over the second-best method (PNP) in terms of F1 score and an average of 6.62% improvement in terms of AUROC on the three datasets.

---

\textsuperscript{3}https://docs.dgl.ai/en/0.8.x/generated/dgl.data.CoauthorCSDataset.html
Furthermore, we examined the detailed classification accuracy in terms of known classes and unknown classes. We found that to gain the ability of unknown class detection, compared to the closed-set classifier, there is a slight decrease in the performance of known class classification, i.e., from 86.33% (GCN\textsubscript{soft}) to 75.48% (ROG\textsubscript{PL}) on average, while the unknown class detection accuracy is increased from 0% to a remarkable 83.78% on average. Moreover, compared to other open-set node classification methods, such as $G^2P_{xy}$, ROG\textsubscript{PL} achieves an average improvement of 11.9%, 13.29% and 9.94% in accuracy of known classification, unknown detection and overall classification, respectively. The average improvement in F1 score is 9.94%. Details are provided in the Appendix.

### Open-Set Node Classification with IND Noise and Far OOD Data

To investigate the effect of rough OOD noise, we used OOD samples that are from different dataset as the source of OOD noise. Specifically, we randomly selected samples of the first two classes from Pubmed (Hu et al. 2020) and mixed them with data from Cora, Citeseer and Coauthor-CS to create the training set with far OOD noise, where the OOD noise rate is set to 5% or 25%. Each node from Pubmed was randomly assigned to a known class label, and edges between the node and its $k$-nearest neighbors were added, where $k$ is a random integer in the range from 1 to 5. We also added some samples from the remaining categories in the Pubmed dataset to the test set in corresponding proportion, to evaluate the performance of ROG\textsubscript{PL} on the samples from far open-set. The other settings are the same as the experiment of near OOD data, and the IND noise rate is set as 5%.

The results of open-set node classification with far OOD noise are presented in Table 2. The results show that ROG\textsubscript{PL} generally outperforms the baselines, achieving an average improved of 2.84% in F1 and an average improvement of 5.33% in AUROC, compared to the global second-best method PNP. The detailed classification accuracy in terms of known classes and unknown classes with far OOD noise are given in the Appendix.

### Robustness Analysis under Different Noise Rate

In this section, we evaluate the robustness of ROG\textsubscript{PL} for different levels of IND and OOD noise for the open-set node classification task. We first examine how ROG\textsubscript{PL} reacts to different IND noise rate. We kept the OOD noise rate constant and used the same setting as the experiment of Table 1, while we varied the IND noise rate from 0%, 5%, 25%, 50% to 75%. Fig. 2(a) and Fig. 2(b) show the results on the Cora and Citeseer datasets, respectively. We observe that ROG\textsubscript{PL} maintains a relatively stable performance when the noise rate is within a certain range, for example no more than 50%. However, once the IND noise exceeds a certain threshold, ROG\textsubscript{PL}’s performance drops sharply.

Additionally, we investigate the performance of ROG\textsubscript{PL} in terms of different OOD noise rate, by keeping the IND noise rate constant at 5%, while varying the far OOD noise rate from 0%, 5%, 25%, 50% to 75%. The results on Cora and Citeseer are shown in Fig. 2(c) and Fig. 2(d) respectively. It can be observed that ROG\textsubscript{PL} maintains a surprisingly stable performance even with large amounts of far OOD noise in the training data, which demonstrates the strong robustness of ROG\textsubscript{PL} against OOD noise.

### Ablation Study

We compare variants of ROG\textsubscript{PL} in an ablation study to evaluate the effect of its main modules and settings:

- ROG\textsubscript{PL}--$G_N$: a variant of ROG\textsubscript{PL} without building $k$-nearest neighbor graph $G_N$, and original graph $G$ is used
in the label propagation stage.

- **ROG\(_{PL}-\text{denoise}\)**: a variant of ROG\(_{PL}\) without the label propagation based denoising module.
- **ROG\(_{PL}-\text{region}\)**: a variant of ROG\(_{PL}\) without clustering, i.e., the interior prototype of each class is updated by the clean nodes from all classes, and there is no border prototypes.
- **ROG\(_{PL}-\text{div}\)**: a variant of ROG\(_{PL}\) with loss \(L_{\text{div}}\) removed. We only utilize classification loss \(L_{\text{cls}}\) to train the encoder network \(f_0\) and prototype network \(f_0\).

The performance of the proposed method and its four variants are presented in Table 3. The results demonstrate that both label propagation based denoising module and region-based prototype learning module are important and building \(k\)nn graph \(G_N\) is necessary. The large gap of performance between ROG\(_{PL}\) and ROG\(_{PL}-\text{region}\) verifies the contribution of open-set prototypes on open-set node classification.

**Conclusion**

This paper introduced a novel prototype learning based robust open-set node classification method to learn an open-set classifier from graphs with mixed IND and OOD noisy nodes. By correcting noisy labels through similarity-based label propagation and removing low-confidence samples, the proposed method relieve the intra-class variety caused by noise. Further, by learning open-set prototypes via non-overlapped regions and remaining both interior and border prototypes, the method can remedy the inter-class confusion problem, and save more space for open-set classes. To the best of our knowledge, the proposed ROG\(_{PL}\) is the first robust open-set node classification method for graph data with complex noise. Experimental evaluations on several benchmark graph datasets demonstrates its good performance.
Acknowledgments

This research was supported by National Natural Science Foundation of China (62206179, 92270122), Guangdong Provincial Natural Science Foundation (2022A1515010129, 2023A1515012584), University stability support program of Shenzhen (20220811121315001), Shenzhen Research Foundation for Basic Research, China (JCYJ20210324093000002).

References


