

Article

InfoMax Classification-Enhanced Learnable Network for Few-Shot Node Classification

Xin Xu ¹ , Junping Du ^{1,*}, Jie Song ² and Zhe Xue ¹

¹ Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

² FreeWheel, Beijing 100026, China

* Correspondence: junpingd@bupt.edu.cn

Abstract: Graph neural networks have a wide range of applications, such as citation networks, social networks, and knowledge graphs. Among various graph analyses, node classification has garnered much attention. While many of the recent network embedding models achieve promising performance, they usually require sufficient labeled nodes for training, which does not meet the reality that only a few labeled nodes are available in novel classes. While *few-shot learning* is commonly employed in the vision and language domains to address the problem of insufficient training samples, there are still two characteristics of the few-shot node classification problem in the non-Euclidean domain that require investigation: (1) how to extract the most informative knowledge for a class and use it on testing data and (2) how to thoroughly explore the limited number of support sets and maximize the amount of information transferred to the query set. We propose an InfoMax Classification-Enhanced Learnable Network (ICELN) to address these issues, motivated by Deep Graph InfoMax (DGI), which adapts the InfoMax principle to the summary representation of a graph and the patch representation of a node. By increasing the amount of information that is shared between the query nodes and the class representation, an ICELN can transfer the maximum amount of information to unlabeled data and enhance the graph representation potential. The whole model is trained using an episodic method, which simulates the actual testing environment to ensure the meta-knowledge learned from previous experience may be used for entirely new classes that have not been studied before. Extensive experiments are conducted on five real-world datasets to demonstrate the advantages of an ICELN over the existing few-shot node classification methods.



Citation: Xu, X.; Du, J.; Song, J.; Xue, Z. InfoMax Classification-Enhanced Learnable Network for Few-Shot Node Classification. *Electronics* **2023**, *12*, 239. <https://doi.org/10.3390/electronics12010239>

Academic Editor: Javid Taheri

Received: 13 December 2022

Revised: 25 December 2022

Accepted: 28 December 2022

Published: 3 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: graph representation; node classification; few-shot learning; mutual information maximization

1. Introduction

A variety of real-world situations such as social networks, citation networks, and knowledge graphs can be modeled with graphs. Among graph-based processing and analysis, the node classification problem is an essential but necessary topic. Nodes' classification results and predicted labels are widely used for downstream tasks. However, graph structure data are non-Euclid, which increases the difficulties of learning the representation of nodes. Thus, special care should be taken with the topology and the nodes' attributed information when learning node representations.

End-to-end training is utilized by many graph neural network (GNN) models in order to acquire knowledge about the nodes' embedding vector representations in low dimensions, which benefits the node classification problem. Within this supervised or semi-supervised training paradigm, large numbers of node classes and node labels are essential to obtain good performance. Unfortunately, data labeling is a task that requires a lot of time and effort. Usually, only a small part of the data can be labeled. Insufficient labeled data may easily lead to the overfitting problem, which inhibits the generalization

ability of the model. To this end, learning node representation in a low-data regime is highly significant.

Recently, few-shot learning (FSL) has been proposed and researched to solve the problem of learning from very few labeled examples. An FSL model usually adheres to the *meta-learning* paradigm, which involves the extraction of information from a variety of meta-training tasks that are derived from classes including a significant amount of data with labels. With the episodic training method, the model may be fine-tuned in only a few stages. The information can be automatically extended to a new task (also known as *meta-testing*) from previously unseen classes throughout training. When performing a task, the *support set* and *query set* will operate as simulations of the real-world testing conditions, in which a handful of labeled instances (the support set) will be made accessible alongside a massive portion of unlabeled data (the query set) to be categorized.

A significant line of FSL research has been explored on image and text data, while few focus on graph structure data. The high-dimension property and topology information bring difficulties to node representations. Approaches such as the Meta-GNN [1], AMM-GNN [2], and GPN [3] have been designed specifically for few-shot node classification. However, they focus on the meta-learning framework and the attributed feature extraction and ignore taking care of the amount of information transferred between a support instance and a query instance. The challenges for few-shot learning on node classification still lie in two aspects: (1) It is impossible to quantify directly the quantity of information included inside the learned node representation. Therefore, when transferring knowledge from the support set instances to the query set instances, there is no guarantee that the maximum effective information can be delivered. (2) Limited by the number of labeled support set instances, obtaining discriminative feature representation for a particular class is difficult. Under this low-data circumstance, the crucial problem is ensuring that every unlabeled query instance can absorb as much category information as possible from the class representation.

This paper uses the mutual information maximization principle to address the challenges mentioned above and proposes an InfoMax Classification-Enhanced Learning Network (ICELN). The minimal support set nodes are not only used for the task-specific classifier but have also been processed as a class summary to anchor and maximize the amount of information for the query set nodes. The proposed framework is made up of two crucial components that effortlessly cooperate in obtaining node representations: (1) The few-shot graph representation learning module consists of a GNN-based network encoder and samples a series of few-shot node classification tasks. The encoder is responsible for the extraction of expressive node representations, and meta-knowledge is passed between these tasks. (2) The InfoMax classification-enhanced learning module is designed to get the most out of the information that is shared by the class anchor and the query nodes, which restricts and enhances the network representation learning to obtain informative features. With InfoMax classification-enhanced learning, the finite labeled support nodes are being put to good use, and the valid information can be transferred to unlabeled query nodes as much as possible, which ultimately improves the model's capacity to characterize the node class feature. The episodic meta-training paradigm also guarantees that the meta-knowledge about the graph structure and attributed features extracted through various meta-training tasks may be dynamically adapted to specific few-shot classification tasks. The contributions of this paper are the following:

- A new few-shot node classification framework (ICELN) is proposed, where we emphasize learning task-specific classifiers from a limited number of labeled nodes and transfer the discriminative class characteristics to unlabeled nodes. The ICELN is able to explore the limited number of support nodes to achieve a better generalization ability.
- We explore the effectiveness of the mutual information maximization principle in node representation learning. By increasing the amount of mutual information shared between the known class representation and the corresponding node representation,

the most representative features and information can be transferred, enhancing the node representation learning.

- To demonstrate the effectiveness of the ICELN, we test it using a broad range of datasets derived from the actual world and conduct a number of intensive experiments. The experiments show that the ICELN achieves competitive performance on several challenging few-shot node classification datasets.

2. Related Works

The proposed ICELN is connected to the following three works: graph representation learning, few-shot learning, and mutual information maximization.

2.1. Graph Representation Learning

Graphs, which describe the items in the actual world as well as the connections between them using nodes and edges, are everywhere. Many research interests have been developed in graphs to capture the intrinsic property inside node connections and then apply the knowledge to downstream tasks. Driven by the success of deep learning, network embedding [4] extended the traditional convolution operations on grid data to the graph structure domain and has become a popular method for graph representation learning. The GNN [5] was one of the pioneer works that propagated neighboring nodes' and edges' information and iteratively updated the representation of a node via a recurrent neural architecture. A number of graph convolutional networks (GCNs) have arisen in diverse applications as a result of the GNN's outstanding performance in graph spectral theory, such as recommendation systems [6,7], behavior modeling [8], and anomaly detection [9]. The GCN [10] takes advantage of the localized first-order approximation of spectral convolutions and aggregates data from a node's immediate surroundings. GraphSAGE [5] adapts several types of aggregation functions to obtain nodes' embedding representations, such as mean, pooling, and LSTM aggregators. During the aggregation phase, the graph attention networks (GATs) [11] make use of a technique called attention to assign varying weights to the nodes that are neighbors to each individual node. The graph isomorphism network (GIN) [12] also uses an attention mechanism with arbitrary aggregation functions and theoretically proves that it is as powerful as the Weisfeiler–Lehman (WL) graph isomorphism test. Simple graph convolution (SGC) [13] is a modification of the original GCN model that eliminates nonlinearities and collapses unimportant weight matrices across consecutive layers in order to cut down on unnecessary complexity. Nevertheless, these GNN-based methods focus on semi-supervised node representation, and a significant number of positive examples are required in training. In situations where there is an inadequate amount of labeled data, the performance of these models will suffer an inevitable decline and generalize poorly with unseen classes. With such a small sample size, the proposed ICELN utilizes the GNN architecture as the encoder to obtain node representations from the graph structure data and utilizes the meta-learning training paradigm to address the problem of insufficient training data.

2.2. Few-Shot Learning

Few-shot learning (FSL) is an approach to gaining information from previous experiences and applying that meta-knowledge to new challenges using just a small amount of labeled data. In the context of few-shot learning, tasks are typically selected at random from a task distribution and then further subdivided into *meta-train* tasks and *meta-test* tasks. These tasks cover both the base class and new classes. Each task is trained to provide an accurate prediction of query sets' labels after adapting the transferable prior knowledge from a handful of support set samples. Generally, the few-shot learning models are of two kinds: models based on *optimization* and models based on *metrics*. The optimization-based models aim to obtain a meta-learner that can be optimized within a few steps given the gradient of limited positive examples. MAML [14] is one of the most representative optimization-based models that learns a parameter initialization capable of learning how

to perform a new task rapidly by performing just a few gradient modifications. MAML is model-agnostic and is suitable for different FSL tasks. The meta-learner LSTM [15] is a model that is capable of learning parameters' initialization and updating mechanisms by giving only small training steps. The LSTM states are used to represent the classifier's parameters' updates. Meta-SGD [16] can quickly initialize and adjust any differentiable of the learner in a single step by reasoning how to initialize the weights, how to update the gradient, and how to set the learning rate. SNAIL [17] employs a mix of soft attention and temporal convolutions to train a general meta-learner architecture that combines knowledge from previous experiences and identifies certain information. The metric-based models aim to learn task-invariant metrics and generalizable matching functions between the labeled samples from the support set and the unlabeled samples from the query set across different tasks. Similar data are clustered together in the learned embedding space, whereas dissimilar data are spread away, and the classification process then turns into finding the nearest neighbor. Matching networks [18] make predictions by comparing the input instances with a few-shot labeled support set and training a weighted differentiable closest neighbor classifier from end to end. Prototypical networks (PNs) [19] will first calculate a class prototype for each class, and then they will categorize the instances according to the Euclidean distances that separate them from the class prototype. In a relation network (RN) [20], scores indicating the degree of the link between a query sample and the support sample are used during training of the auxiliary network as the metric to categorize data. With the idea of learning class prototypes, graph prototypical networks (GPNs) [3] explore metric learning on attributed graph data. In the low-data regime, a class's prototypical representation is obtained by considering the support nodes' informativeness, and the prediction is made by finding each query's nearest class prototype. With the inspiration of the GPN, we developed the ICELN, which aims to enhance graph representation learning by boosting the amount of mutual information that can be gained between the class prototype and a query instance.

2.3. Mutual Information Maximization

Mutual information (MI) quantifies the interdependence between two random variables or distributions [21]. The higher the mutual information is, the more significant the reduction in uncertainty is, and two independent variables have zero mutual information. However, it is challenging to compute the MI in environments where data are high-dimensional and continuous [22], especially when the probability distribution is uncertain for learning. Fortunately, MINE [21] made significant progress in tackling the problem with the neural estimator. It mathematically derives the lower limit of MI and then trains a statistical network to act as a discriminator in order to separate samples from two variables' joint distribution or their marginals' product. Based on the neural estimator, DIM [23] utilizes a neural network encoder to obtain image representations, and the mutual information between the encoder's input character and output embedding vector is maximized by the estimator. In order to learn informative node representations, DGI [24] first extends MINE and DIM into the graph structure data. The mutual information between every single node (a local patch) and the overall graph representation (a global summary) is maximized. Based on DGI, InfoGraph [25] learns node representations by taking advantage of different scales' substructures. The MI maximization takes place in the substructure representations and the graph-level representations. GIC [26] further exploits MI on the heterogeneous graph and acquires node-embedding vectors by concurrently maximizing the MI between the summaries of different clusters and a summary of the whole graph. In order to simulate a heterogeneous graph, DMGI [27] divides it into homogeneous graphs and then adapts DGI's discriminator and mutual information maximization goal to represent each split. The simple but essential DGI model inspires the ICELN. We also optimize the mutual information that exists between a class prototype and the query representations to enhance the class information transfer process and boost the network encoding efficiency. The most significant difference between the ICELN and DGI is that we adapt the maximization

principle to the few-shot problem and encourage each query to integrate as much class information as possible, where the class prototype is generated with only a handful of support samples.

3. The Proposed Method

Following the overview of our methodology, which is shown in Figure 1, we will proceed to provide in-depth explanations of the primary components of the proposed ICELN, which are as follows.

The **few-shot graph representation learning** component is the backbone of our framework. It is composed of a graph encoder and a number of meta-training tasks. The graph representation learning process is trained in an episodic paradigm and optimized based on the classification performance of the query set nodes.

The **InfoMax classification-enhanced learning** component works within a meta-training task and jointly uses support nodes and query nodes to maximize the mutual information while minimizing the classification cross-entropy loss.

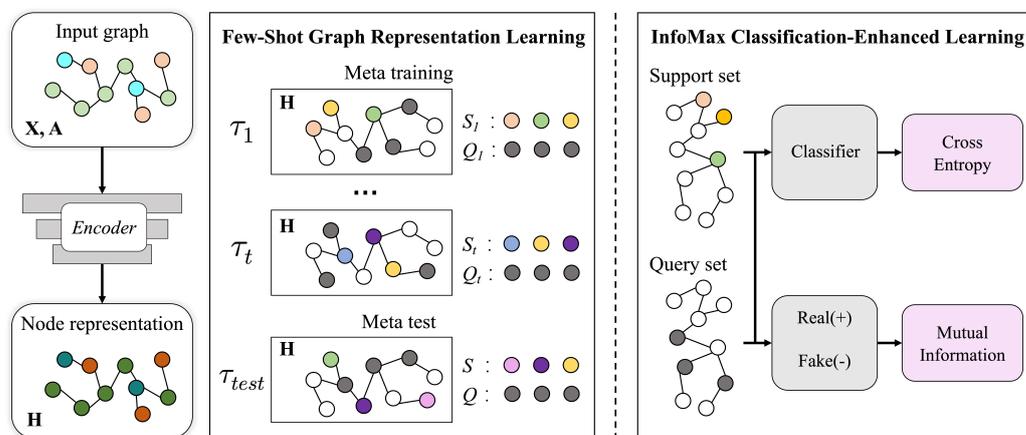


Figure 1. Overall flow and framework of the proposed ICELN.

3.1. Notation and Problem Definition

In accordance with the notations that are most often used, we denote sets as calligraphic fonts (e.g., \mathcal{G}) and vectors as bold lowercase letters (e.g., \mathbf{u}). Scalars are represented by lowercase letters (e.g., l). We refer to matrices as bold uppercase letters (e.g., \mathbf{W}), and \mathbf{w}_i indicates the i th row in \mathbf{W} .

An undirected attributed graph is referred to as a quadruple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}\}$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the node set, n denotes the total number of nodes, $\mathcal{E} = \{e_{ij} = (v_i, v_j)\} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, $\mathbf{A} \in \mathbb{R}^{n \times n}$ represents the asymmetric adjacency matrix that represents the graph structure, and $a_{ij} = 1$ means node v_i and node v_j are connected by an edge. $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the attributed feature matrix, a row vector $\mathbf{x}_i \in \mathbb{R}^d$ represents a node's (v_i) characteristic information, and its dimension is d .

Problem Definition: With a direct attributed graph \mathcal{G} , we are looking forward to learning a network encoder and a classifier from a limited number of labeled nodes that can be adapted to a disjoint set of new classes to predict nodes' labels. In most cases, there are just a few nodes for which the labels may be provided. Formally, following the few-shot learning (FSL) settings, the nodes in training set \mathcal{D}_{train} contain C classes, and the nodes in the disjoint testing set \mathcal{D}_{test} contain N unseen classes. In \mathcal{D}_{test} , if each class has K nodes that are labeled and make up the *support set* S , the task is referred to as an N -way K -shot node classification task, and K is usually a relatively small number, such as one, three, or five. Our purpose is finding and training a suitable network encoder f_θ that can obtain good node representations and make accurate predictions about the labels of the remaining unlabeled nodes (also called the *query set* Q). Thus, the meta-learning problem on the graph

is addressed as a few-shot node classification problem, and the key lies in how to extract knowledge that can be transferred from the training data to testing data that have not been seen before.

3.2. Few-Shot Graph Representation Learning

The methodology for addressing the problem of few-shot node classification largely depends on the graph representation learning module as its core component. In designing the framework, we made sure to take into account two difficult research topics:

- How to conduct meta-learning on graph structure data and how to extract information that can be transferred from the training data to testing data when there are only a handful of labeled nodes available.
- How to make the most of the limited number of nodes that are available in the support set and restrict the query set to obtain informative and discriminative representations?

As can be seen in Figure 1, we constructed a meta-learning structure that is based on the concept of episodic categorization. To be more specific, the training process is composed of a large number of meta-training tasks sampled from task distribution, which is denoted as \mathcal{T} , and trained episodically. The training process mimics the actual testing setting to alleviate the distribution gap between training and testing. After studying a significant amount of different episodes, the knowledge in graph topology and feature distribution can be extracted and passed through each episode and then applied to unseen classes in meta-testing tasks.

In order to assure that there will be no differences between the training and the tests, each task used in the meta-training phase and the meta-testing phase came from the same task distribution \mathcal{T} and were formed as an N -way K -shot classification task \mathcal{T}_t . For every task \mathcal{T}_t , we used \mathcal{S}_t to represent the support set and used \mathcal{Q}_t to represent the query set. $\mathcal{S}_t = \{(v_i, y_i)\}_{i=1}^{N \times K}$ contains $N \times K$ nodes and their corresponding labels from N different classes. $\mathcal{Q}_t = \{(v_j, y_j^*)\}_{j=1}^{N \times M}$ contains $N \times M$ nodes taken from the rest of the N classes, and the unknown labels y^* are the labels to be predicted. The true labels of the query set nodes are disguised throughout the training phase so that it may be more similar to the testing procedure, and the actual labels are used to compute and evaluate the training loss. After T meta-training tasks and optimization with a gradient descent, the model is convergent and may be used to make predictions for unknown query nodes in meta-testing tasks with unseen classes. By completing these meta-training tasks, the model is able to acquire information that is not only portable from one episode to the next but also may be generalized to be applied to meta-testing tasks.

With a given N -way K -shot task, we employed an encoder network to capture the graph topology information \mathbf{A} and the nodes' feature matrix \mathbf{X} to obtain a node's representation. Specifically, we stacked several GNN layers to convey a node to a low-dimensional vector \mathbf{h}_v . The GNNs follow the *AGGREGATION – COMBINE* pipeline, where the *AGGREGATION* step aggregates a node's neighboring information, and the *COMBINE* step compresses the node features from the local neighborhoods. After l instances of *AGGREGATION – COMBINE* operations, each node v 's multi-hop messages are passed, and ultimate node representation $\mathbf{h}_v \in \mathbb{R}^D$ is obtained for predicting its class label y_v . A GNN layer may be described in a formal way as follows, and l is the layer number:

$$\begin{aligned} \mathbf{s}_v^{(l-1)} &= \text{AGGREGATION}\left(\left\{\mathbf{h}_i^{(l-1)} : v_i \in N_v\right\}\right), \\ \mathbf{h}_v^{(l)} &= \text{COMBINE}\left(\left\{\mathbf{s}_v^{(l-1)}, \mathbf{h}_v^{(l-1)}\right\}\right) \end{aligned} \quad (1)$$

where $\mathbf{h}_v^{(l)}$ is the node embedding at layer l that is updated with the *COMBINE* function based on this node's representation and its neighborhood representation $\mathbf{s}_v^{(l-1)}$ at the upper layer and $\mathbf{s}_v^{(l)}$ is the neighborhood representation based on v 's neighborhood set N_v

and the *AGGREGATION* function. The *AGGREGATION* operation can be any form of information aggregator, such as *mean*, *max*, *sum*, and the *COMBINE* function can be *sum*, *concatenation* or others. With the different choices of *AGGREGATION* and *COMBINE* functions, there is a series of implementations [5,10,28]. We stacked L instances of GNN layers above these two operations to build the graph encoder and obtain the node representations that may be used in the output layer for the purpose of node categorization. For simplicity, we denote the graph encoder as f_θ and the graph encoder's output matrix as \mathbf{Z} . \mathbf{Z} 's i th row element, denoted as \mathbf{z}_i , represents the ultimate node representation of node v_i .

3.3. InfoMax Classification-Enhanced Learning

After obtaining the node representations, we could perform classification and evaluate the model based on the misclassification rate. However, when it comes to learning in a few-shot task, the labeled nodes of the support set are limited to a small number, and the unlabeled query set nodes' number is not fixed. We needed to make the most use of the scanty labels to train a task-specific classifier for unlabeled nodes and also make sure the query set nodes kept a similar feature distribution to the support set nodes. Thus, we propose InfoMax classification-enhanced learning. In a few-shot task, we first train a task-specific classifier with support set nodes and then restrict the query set nodes to have similar feature distribution to the support set nodes with mutual information maximization. By doing so, we enhance the accuracy of the node classification while simultaneously training the network encoder to learn the maximum comparable feature representation for the same class of nodes as before.

Class-Specific Classifier: We used support set nodes and their labels to train a task-specific classifier. The following is the definition of the classifier:

$$L_{\text{classifier}} = -\frac{1}{|N \times K|} \sum_{i \in N \times K} \sum_{k=1}^K y_{ik} \log(\mathbf{z}_{ik}) \quad (2)$$

where \mathbf{z}_{ik} denotes the i th node representation of K nodes that belong to one of the N different classes that make up the support set, y_{ik} is its corresponding label prediction, and $N \times K$ is the total number of nodes in the support set.

Mutual Information Constraint: We worked under the assumption that nodes that belong to the same class ought to have feature representations that are comparable to one another. Therefore, we used the InfoMax principle [29] to make the query set nodes representation near the support set nodes representation. To calculate the mutual information (MI) between random variables X and Y , we used the following formula:

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) \\ &= \int_{XY} \log \frac{dP_{XY}}{dP_X \times dP_Y} dP_{XY} \end{aligned} \quad (3)$$

where P_{XY} is the joint distribution of X and Y while P_X and P_Y are the marginal distributions of X and Y , respectively. P_{XY} means considering X and Y comprehensively, and $P_X \times P_Y$ means considering them separately. The MI between X and Y is maximized to increase the distance between P_{XY} and $P_X \times P_Y$, which ultimately means X and Y are highly correlated, and the amount of information these two variables carry is maximized.

However, when X and Y are high-dimensional and with unknown probability distributions, it is not easy to calculate the MI directly. Based on the discriminator, MINE [21] proposed a simple and scalable neural network estimator to estimate mutual information, and DGI [24] applied the method to graph structure data. DGI is designed to maximize the amount of information that the local patches can gain from the graph's global summary. $\mathbf{h}_i \in \mathbb{R}^d$, denoted as the representation of node v_i , is the local patch, and the global sum-

mary is generated by a readout function that is calculating the average of all the attributes of the nodes as its input:

$$\mathbf{s} = \mathcal{R}(\mathbf{Z}) = \sigma \left(\frac{1}{N} \sum_{i=1}^N \mathbf{h}_i \right) \quad (4)$$

where σ represents a logistic sigmoid nonlinearity and N represents the total number of nodes. The discriminator in DGI assigns points based on a bilinear scoring function to the pairings of summary-patch representations:

$$\mathcal{D}(\mathbf{h}_i, \mathbf{s}) = \sigma \left(\mathbf{h}_i^T \mathbf{W} \mathbf{s} \right) \quad (5)$$

where \mathbf{W} represents a score-calculating matrix that can be learned and σ stands for the logistic sigmoid nonlinearity. The higher probability of the summary-patch representation pair is a positive pair, and the more information it conveys, the higher the score given by the discriminator. As MI has been maximized, all the local patch representations are trained and learned to preserve MI with the graph-level representation in order to enable the capture of the global properties shared by the whole graph.

Following the settings in DGI, we aim to learn the node representations for each node to ensure the MI between the query set's nodes' representations and the support set's global summary representation is maximized. To this end, with the help of the readout function, we created a global representation that was a summary of the representations of the nodes in the support set that belonged to the same class:

$$\mathbf{p}_c = \sigma \left(\frac{1}{K} \sum_{i=1}^K \mathbf{h}_i \right) \quad (6)$$

Here, $c = \{1, 2, \dots, N\}$ represents the total number of classes in the few-shot task, and \mathbf{p}_c is the global summary representation of a class in a few-shot task. Thus, the nodes' representations in the query set are the patch representations, and with each class's summary representation, they form the summary-patch pairs to be discriminated. If the pair is true, and the score given by the discriminator is high, then the node representation and class summary are highly correlated. Otherwise, the node representation is irrelevant to this class summary. Therefore, we now have positive samples and negative samples that lead us to the restraint loss:

$$L_{restraint} = \sum_{v_i \in \mathcal{Q}} \log \mathcal{D}(\mathbf{h}_i, \mathbf{p}_c) + \sum_{v_i \in \mathcal{Q}} \log(1 - \mathcal{D}(\mathbf{h}_i, \tilde{\mathbf{p}}_c)) \quad (7)$$

where $\tilde{\mathbf{p}}_c$ is the class summary representation that is irrelevant to this node. Finally, we joined the restraint loss to the classification loss to obtain the final objective \mathcal{L} for a few-shot task as follows:

$$\mathcal{L} = L_{classifier} + \lambda L_{restraint} \quad (8)$$

λ governs how significant the restraint loss is to the overall balance. After completing a number of meta-training tasks using Equation (8), our model can be trained and used to classify unlabeled nodes on the meta-test tasks. Algorithm 1 presents the specific steps that ICELN takes throughout the learning process.

Time Complexity Analysis: The time complexity of Algorithm 1 is mainly spent on acquiring the nodes' representations, which is $O(l(nd)^2)$, where l represents the number of GNN layers, n represents the number of nodes in the graph, and d represents the dimension of the node feature.

Algorithm 1: The detailed ICELN learning framework for few-shot node classification.

Input: Attributed Network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{X}\}$, few-shot node classification tasks $T_{meta-training} = \{\mathcal{S}_t, \mathcal{Q}_t\}$ and $T_{meta-testing} = \{\mathcal{S}, \mathcal{Q}\}$, training episodes E, λ

Output: Node classification results for nodes in \mathcal{Q}

```

1 //Meta-training
2 for  $i < E$  do
3   Sample a meta-training task  $T_i = \{\mathcal{S}_i, \mathcal{Q}_i\}$ ;
4   Obtain node representations in  $\mathcal{S}_i$  and  $\mathcal{Q}_i$ ;
5   Perform task-specific classifier learning according to Equation (2);
6   Perform InfoMax classification-enhanced learning according to
   Equations (6) and (7);
7   Minimize the overall training loss  $\mathcal{L}$  according to Equation (8).
8 end
9 //Meta-testing
10 Obtain node representations in  $\mathcal{S}$  and  $\mathcal{Q}$ ;
11 Perform task-specific classifier learning according to Equation (2);
12 Predict labels for nodes in  $\mathcal{Q}$ .
```

4. Experiments and Analysis

We carried out different kinds of experiments, and we provide the comprehensive results of those experiments in order to provide evidence for the effectiveness of the proposed ICELN.

4.1. Datasets

We carried out the experiments using the set-ups described in [19] and evaluated the proposed ICELN on the following five datasets, which are accessible to the public:

- **Cora** [30] and **CiteSeer** [30] are two similar datasets related to machine learning topics. Their networks are built with the paper citation relation and attached with different words describing the papers' information. There are seven subcategories and six subcategories in Cora and CiteSeer, respectively.
- **DBLP** [31] is also a citation network which was taken from "DBLP dataset (version v11)". The network is linked with the papers' citation relation, and the nodes' attributed information comes from the papers' abstract information. The node class labels represent the locations of the paper's presentations.
- **Amazon-Clothing** [32] is known as a product network built on Amazon (<https://www.amazon.com/>, accessed on 12 December 2022, Amazon review dataset released in 2014). This dataset was first generated in [32], and in [3], the authors preprocessed this dataset so that it could be used in the few-shot problem. The network's nodes are individual goods that are categorized under the heading of "Clothing, Shoes and Jewelry" on Amazon, and the edges are the "also reviewed" relationship between two products. The labels of the nodes represent the lower-level classifications of the products. The attributed information for a node is the description of a product.
- **Amazon-Electronics** [32] is another Amazon product network analogy to Amazon-Clothing. Nodes in this dataset are products belonging to "Electronics". The nodes' attributed information and labels are created in the same way as in Amazon-Clothing. However, the edges are created with a complementary relationship "bought together" between two products.

The statistical information that can be found in the aforementioned datasets is summarized and presented in Table 1.

Table 1. Statistics of datasets.

Datasets	No. of Nodes	No. of Edges	No. of Attributes	No. of Labels
Cora	2708	5429	1433	7
CiteSeer	3327	4732	3703	6
DBLP	40,672	288,270	7202	41
Amazon-Clothing	24,919	91,680	9034	77
Amazon-Electronics	42,318	43,556	8669	167

4.2. Evaluation Metrics

For the node classification problem, we employed the measurements that are most often used: accuracy (ACC) and $F1$. Based on the classification results, we had T_P to represent the number of true positives, T_N to represent the number of true negatives, F_P to represent the number of false positives, and F_N to represent the number of false negatives. Thus, the formal definition of the ACC is

$$Acc = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (9)$$

To obtain $F1$, we needed the definitions of *Precision* and *Recall*:

$$Precision = \frac{T_P}{T_P + F_P} \quad (10)$$

$$Recall = \frac{T_P}{T_P + F_N} \quad (11)$$

Thus, the definition of the $F1$ score is

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

4.3. Compared Methods

The proposed ICELN was evaluated by comparing it with several different baseline models from different encoders:

- **DeepWalk** [33] acquires the node embedding by making use of the local data obtained through a truncated random walk around the graph.
- **Node2vec** [34] extends DeepWalk. With a given node, its diverse neighborhoods are explored, and the random walk then becomes biased.
- **GCN** [10] uses the spectral graph convolution as the basis for obtaining node representations. The graph's topology information and nodes' features are aggregated and passed through the layer-wise propagation pipeline.
- **SGC** [13] follows the learning paradigm of the GCN while simplifying the extra complexity of the convolution function. There are no nonlinearities and collapsing weight matrices between the layers that boost the training process.
- **PN** [19] presents a metric-based space for meta-learning. The prototypes learned from the encoder are used for distance computing when performing classification in a limited-data regime.
- **MAML** [14] is known as a general optimization-based meta-learning framework. It is model-agnostic and can be expanded with a number of similar meta-training tasks beyond classification problems. The universal model may easily be fine-tuned to accommodate new unseen tasks within only a few simple stages of optimization.
- **Meta-GNN** [1] proposes a meta-learning approach as a solution for the few-shot node classification problem. It is trained and optimized based on the training paradigm of MAML, and the classification is learned based on the parameter initialization of GNNs.

- **GPN [3]** combines the training strategy from Meta-GNN and the idea of prototypes from PN to generate class prototypes as anchors to perform node classification tasks on attributed graphs. This semi-supervised model generalized well in testing scenarios.

The aforementioned baseline models may be classified into one of three groups: (1) conventional random walk-based methods that do not use GNN techniques (DeepWalk and Node2vec); (2) GNN-based methods not designed for few-shot scenarios (GCN and SGC); and (3) few-shot methods that especially consider the small sample problem (PN, MAML, Meta-GNN, and GPN).

4.4. Experimental Settings

We use two layers of GNNs to construct the network encoder f_θ with dimension sizes of 32 and 16. We used *ReLU* in each GNN layer as the activation function. For the task-specific classifier in each few-shot task, we used a trainable linear layer with an Adam optimizer. For the discriminator, we made use of a trainable bilinear layer that had a logistic sigmoid nonlinearity as the scoring function. The readout function is an average operation for node representations and is activated with logistic sigmoid nonlinearity. The proposed ICELN model utilizes an Adam optimizer for training whose learning rate is set to $\alpha = 0.005$ in the beginning and whose weight decay factor is set to 0.0005. The dropout technique was used to fine-tune the model in order to prevent the overfitting problem. It would be terminated to accelerate the training process when the validation accuracy had not been increased in 20 consecutive steps. The initialized training episodes E for each training set were set to be 1500 but may have stopped early with the early-stopping strategy. The value of the balance parameter λ was initialized to be 0.5. All the training and evaluation was carried out with PyTorch and executed on a system running Ubuntu 18.04 equipped with four Nvidia GeForce RTX 2080 Ti GPUs.

4.5. Performance Comparison

Due to the different magnitudes of numbers of the node classes in these datasets, we conducted experiments with different settings for each data set. For DBLP, Amazon-Clothing, and Amazon-Electronics, there were sufficient node classes to sample different tasks, so we devised four distinct types of tasks: 5-way 3-shot, 5-way 5-shot, 10-way 3-shot, and 10-way 5-shot. Following the settings in [3], for all three datasets, we randomly divided them into meta-train/meta-validation/meta-test sets, and there were 80/27/30, 40/17/20, and 90/37/40 classes of nodes for each dataset, respectively.

For Cora and CiteSeer, we designed two distinct forms of few-shot node classification tasks, known as the two-way one-shot task and two-way three-shot task, using random selection to pick two classes for use as meta-testing tasks and the other classes for use as meta-training tasks. In order to make the comparison more impartial and reduce the unstable interference caused by the limitations of the number of training classes, five cross-validations were conducted on Cora and CiteSeer.

All kinds of few-shot node classification tasks were evaluated with the metrics introduced in Section 4.2. When evaluating, the ACC and F1 score were reported with 50 meta-testing tasks on each method with DBLP, Amazon-Clothing, and Amazon-Electronics, and the ACC was reported on 50 nodes' classifications with Cora and CiteSeer. We reused the results reported in [1–3] since our experimental settings were consistent with theirs. The node classification results on Cora and CiteSeer are presented in Table 2, and the classification results on DBLP, Amazon-Clothing, and Amazon-Electronics are presented in Tables 3–5, respectively. The larger number, the better performance in this task setting. The bold font indicates the best performance. Based on these tables, the following are some observations that we made:

- On each of the five datasets, our proposed ICELN delivered performance that was competitive for the few-shot node classification tasks. The high accuracies and F1 scores demonstrate that the ICELN was able to extract the meta-knowledge across diverse tasks and could achieve a better generalization ability in unseen target tasks. In

general, the classification accuracies of the ICELN outperformed the most comparative method (GPN) by 12.7% under the 5-way 3-shot task on Amazon-Electronics.

- We found that the accuracies and F1 scores of DeepWalk and Node2vec were not at the same level as other methods. These two random walk-based methods are supervised and rely largely on sufficient training samples to obtain good node representations. Meanwhile, the GCN and SGC are GNN-based models and use an end-to-end training paradigm, which fluctuates in different task settings. The fact that standard GNN models are susceptible to overfitting when the training instances are restricted to a small number and tasks are distinct from one another demonstrates the requirement for a meta-learning architecture for the low-data challenge.
- Classic meta-learning approaches such as PN and MAML provided results that performed pretty well when used for the classification of few-shot images, while they were only passable for few-shot node classification. Images and graphs are inherently different, and the nodes' correlation and attributed features are the most crucial information during graph learning. Thus, these two models failed to characterize the topological and semantic information, resulting in unsatisfactory performance.
- Both the Meta-GNN and the GPN are two correlative approaches that were developed for the few-shot node classification problem. They are capable of achieving significant gains in comparison with other baselines. Meta-GNN is an optimization-based model that needs to be fine-tuned on target tasks. The GPN learns the informative prototypes for each class while neglecting to pass the support set's class-specific information to the query set. However, by taking full advantage of the handful support set and performing InfoMax classification-enhanced learning, the proposed ICELN was capable of providing better results than these baselines in the majority of situations.
- The following are some of the reasons why the proposed ICELN was successful in achieving outstanding results in few-shot node classification: (1) The ICELN trains a task-specific classifier with the finite support set nodes and successfully passes this valid information to the query set nodes. (2) The ICELN maximizes the information transferred from the support class representation to the query nodes representation, which in turn improves the learning ability of the node representation.

Table 2. The few-shot node classification ACC results on Cora and CiteSeer.

Methods	Cora		CiteSeer	
	2-Way 1-Shot	2-Way 3-Shot	2-Way 1-Shot	2-Way 3-Shot
	ACC	ACC	ACC	ACC
DeepWalk	16.1	25.7	14.5	21.2
Node2vec	15.2	25.7	13.0	20.0
GCN	60.3	75.2	58.4	68.0
SGC	61.6	75.7	56.9	65.7
PN	56.2	63.5	54.3	58.4
MAML	58.3	68.2	56.9	62.8
Meta-GNN	65.3	77.2	61.9	69.4
GPN	64.3	67.5	62.2	64.2
ICELN	71.4	81.9	67.6	72.9

Table 3. The few-shot node classification ACC and F1 results on DBLP.

Methods	DBLP							
	5-Way 3-Shot		5-Way 5-Shot		10-Way 3-Shot		10-Way 5-Shot	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
DeepWalk	44.7	43.1	62.4	60.4	33.8	30.8	45.1	43.0
Node2vec	40.7	38.5	58.6	57.2	31.5	27.8	41.2	39.6
GCN	59.6	54.9	68.3	66.0	43.9	39.0	51.2	47.6
SGC	57.3	55.2	65.0	62.1	40.2	36.8	50.3	46.4

Table 3. *Cont.*

Methods	DBLP							
	5-Way 3-Shot		5-Way 5-Shot		10-Way 3-Shot		10-Way 5-Shot	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
PN	37.2	36.7	43.4	44.3	26.2	26.0	32.6	32.8
MAML	39.7	39.7	45.5	43.7	30.8	25.3	34.7	31.2
Meta-GNN	70.9	70.3	78.2	78.2	60.7	60.4	68.1	67.2
GPN	74.5	73.9	80.1	79.8	62.6	62.6	69.0	69.4
ICELN	76.8	75.7	82.9	82.5	63.4	62.6	70.8	69.9

Table 4. The few-shot node classification ACC and F1 score results on Amazon-Clothing.

Methods	Amazon-Clothing							
	5-Way 3-Shot		5-Way 5-Shot		10-Way 3-Shot		10-Way 5-Shot	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
DeepWalk	36.7	36.3	46.5	46.6	21.3	19.1	35.3	32.9
Node2vec	36.2	35.8	31.9	40.7	17.5	15.1	32.6	30.2
GCN	54.3	51.4	59.3	56.6	41.3	37.5	44.8	40.3
SGC	56.8	55.2	62.2	61.5	43.1	41.6	46.3	44.7
PN	53.7	53.6	63.5	63.7	41.5	41.9	44.8	46.2
MAML	55.2	54.5	66.1	67.8	43.3	46.8	45.6	53.3
Meta-GNN	74.1	73.6	77.3	77.5	61.4	59.7	64.2	62.9
GPN	75.4	74.7	78.6	79.0	65.0	66.1	67.7	68.9
ICELN	77.0	75.8	82.1	81.4	67.2	66.2	71.0	70.0

Table 5. The few-shot node classification ACC and F1 score results on Amazon-Electronics.

Methods	Amazon-Electronics							
	5-Way 3-Shot		5-Way 5-Shot		10-Way 3-Shot		10-Way 5-Shot	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
DeepWalk	23.5	22.2	26.1	25.7	14.7	12.9	16.0	14.7
Node2vec	25.5	23.7	27.1	24.3	15.1	13.1	17.7	15.5
GCN	53.8	49.8	59.6	55.3	42.4	38.4	47.4	48.3
SGC	54.6	53.4	60.8	59.4	43.2	41.5	50.0	47.6
PN	53.5	55.6	59.7	61.5	39.9	40.0	45.0	44.8
MAML	52.1	59.0	58.3	37.4	36.1	43.4	43.4	41.4
Meta-GNN	63.2	61.5	67.9	66.8	58.2	55.8	60.8	60.1
GPN	64.6	62.9	70.9	70.6	60.3	60.7	62.4	63.7
ICELN	77.3	76.8	82.9	82.5	63.4	62.6	70.8	69.9

4.6. Parameter Analysis

InfoMax classification-enhanced learning has two critical parts: the task-specific classifier trained with the support set nodes and the mutual information constraint trained with the query set nodes. The final objective \mathcal{L} is obtained from these two parts, and λ is the balance parameter that controls how important the mutual information constraint loss is in the final objective. In order to determine how selecting λ impacts the outcome, we performed sensitivity analysis on three different datasets using four distinct sorts of settings, and the accuracy (ACC) results are reported in Figures 2–4. From Figure 2 through Figure 4, we were able to conclude that the optimal performance on DBLP was accomplished by setting λ to 0.3 while using the 5-way 5-shot set-up, while the second-highest accuracy was obtained when $\lambda = 0.7$ on Amazon-Electronics using the same set-up. On the Amazon-Clothing dataset, the accuracy reached its peak when $\lambda = 0.7$. Under other few-shot settings, the highest accuracies were obtained when $\lambda = 0.3$, $\lambda = 0.6$, and $\lambda = 0.8$. Based on the above observations, we set $\lambda = 0.7$ in the other experiments.

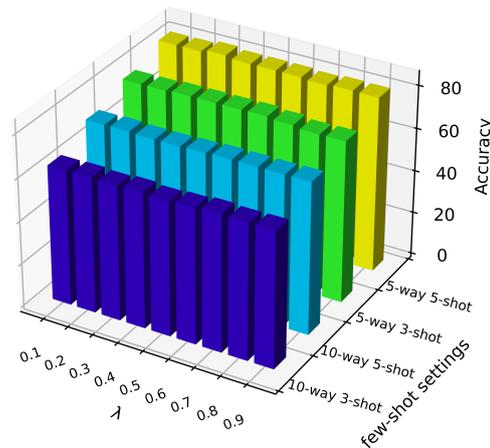


Figure 2. Node classification results based on choosing the balancing parameter λ on DBLP.

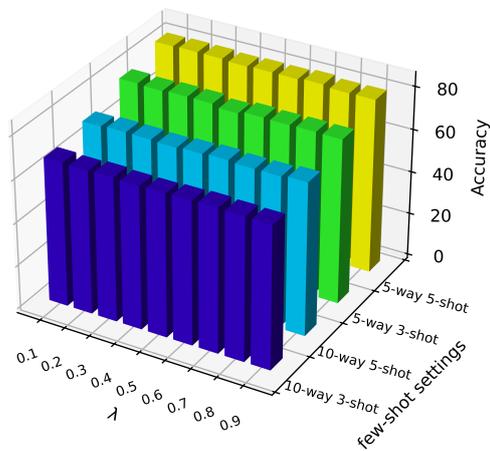


Figure 3. Node classification results based on choosing the balancing parameter λ on Amazon-Clothing.

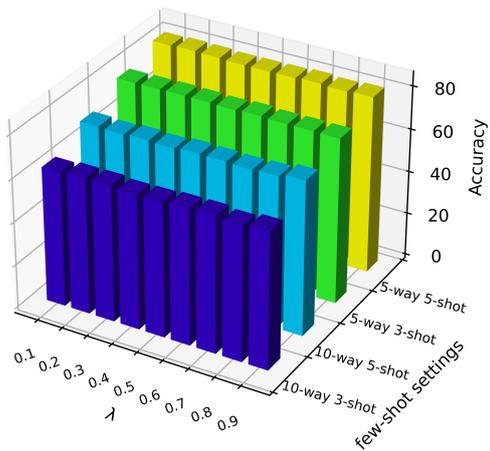


Figure 4. Node classification results based on choosing the balancing parameter λ on Amazon-Electronics.

4.7. Ablation Study

The task-specific classifier and the mutual information constraint in InfoMax classification-enhanced learning jointly promoted the network encoding performance. To validate each component's effectiveness, we introduced two abbreviated versions of the ICELN for an ablation study: ICELN-classify and ICELN-constraint. ICELN-classify contained the task-specific classifier only, and ICELN-constraint contained

the mutual information maximization only. We conducted four kinds of few-shot setting node classification on three relatively larger datasets, and the performances are reported in Figures 5–7. As we can see from Figures 5–7, the integrated ICELN performed consistently better than the two variant models, which validates the necessity of the task-specific classifier module and the mutual information maximization constraint module in the whole ICELN model. Additionally, we were able to deduce from Figures 5 and 6 that ICELN-classify largely fell behind the full ICELN model, which proves the importance of the mutual information maximization component.

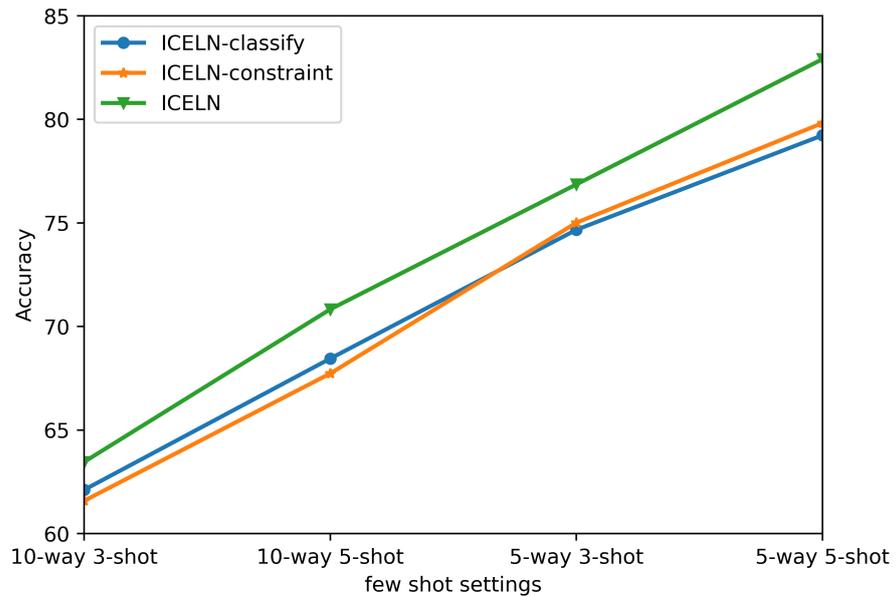


Figure 5. Ablation study on DBLP.

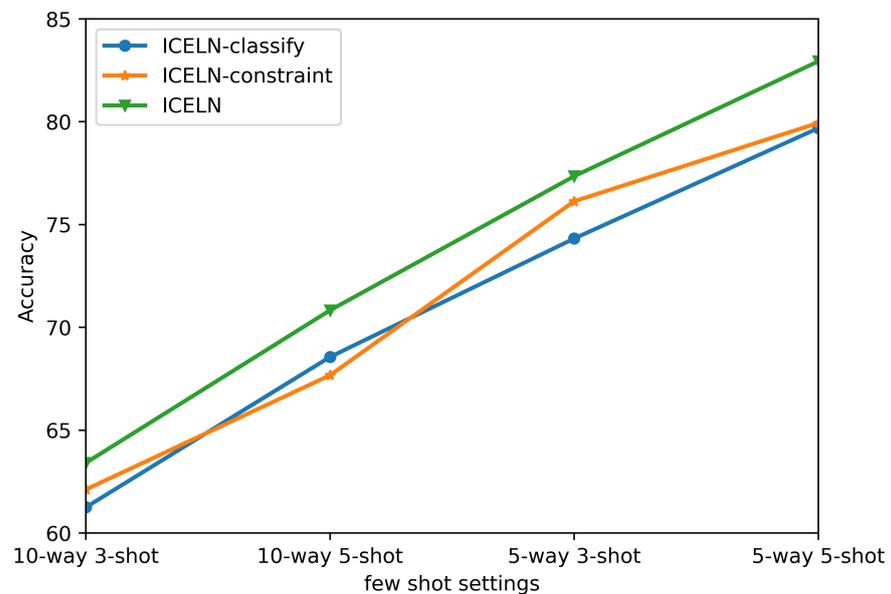


Figure 6. Ablation study on Amazon-Clothing.

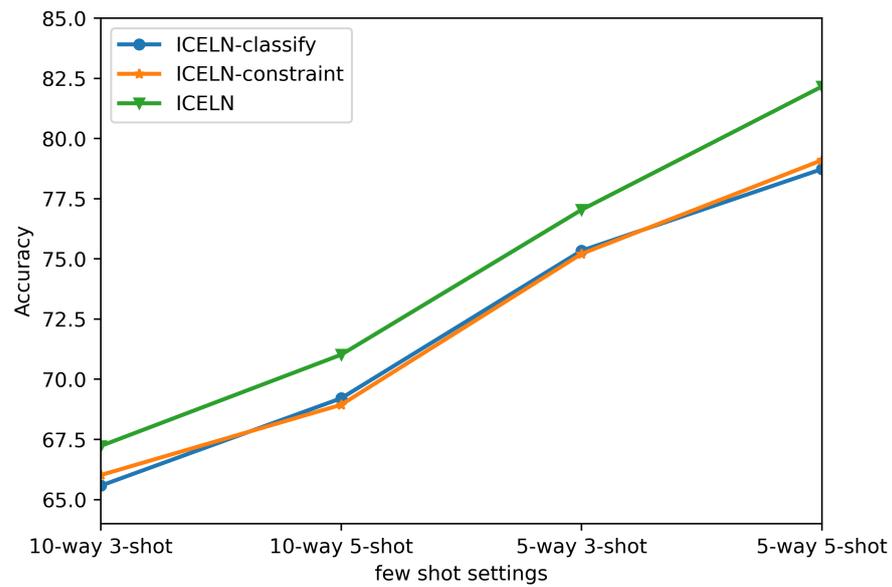


Figure 7. Ablation study on Amazon-Electronics.

4.8. Visualization

Figures 8 and 9 show the similarity matrices of the raw node feature and the node representation learned by our approach on DBLP with the five-way five-shot task, respectively. The horizontal coordinate is the nodes' index from the support set in five classes, and the vertical coordinate is the random five query nodes' index in each query set. Each grid indicates the negative Euclidean distance between two nodes, and the color of the grid demonstrates the divergence between their classes. Figure 8 shows that the nodes were mixed together and had no distinct class patterns. The learned node similarity metric obtained by the ICELN network encoder in Figure 9 demonstrates that the similarities within classes increased while the similarities across classes were decreasing. These two figures display that the ICELN has the capacity to capture class-specific characteristics even when there is only a handful of positive samples in each class.

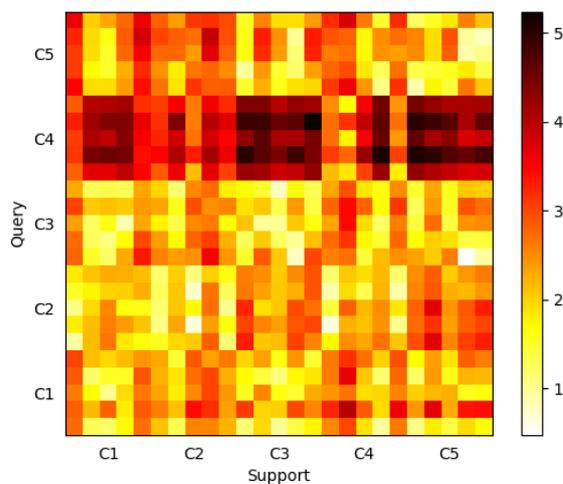


Figure 8. Visualization on DBLP (raw features).

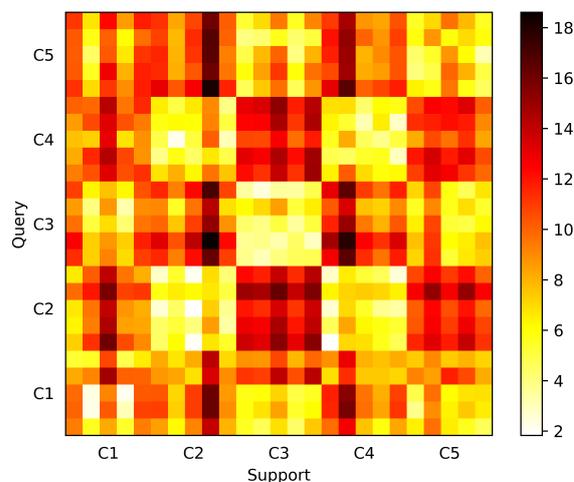


Figure 9. Visualization on DBLP (learned node embeddings).

5. Conclusions

The non-Euclidean domain network structure embedding and the restricted training samples for training are the two primary challenges in the process of node classification, which is an important but not yet fully studied subject. This paper proposes an InfoMax classification-Enhanced Learnable Network (ICELN) to address the few-shot node classification problem. ICELN starts with the process of obtaining node representation by utilizing a GNN-based network encoder and then trains with an episodic strategy to extract knowledge across diverse tasks. Then, the InfoMax Classification-Enhanced Learnable Network trained a task-specific classifier with support set nodes and maximized the mutual information between the class prototype representation and a query node's representation. With these two modules, the ICELN can learn from previous experience, adapt the meta-knowledge to unseen target tasks, and maximize the amount of information transferred across different tasks. The performance over five datasets derived from the real-world demonstrates that the ICELN outperformed other baseline models in the few-shot node classification problem.

Author Contributions: Conceptualization, X.X. and J.S.; methodology, X.X., J.S., and Z.X.; software, X.X. and J.S.; validation, X.X. and J.S.; formal analysis, X.X. and Z.X.; investigation, X.X.; resources, X.X. and J.S.; data curation, X.X. and J.S.; writing—original draft, X.X.; writing—review and editing, X.X., J.D., and Z.X.; visualization, X.X. and J.S.; supervision, J.D. and Z.X.; project administration, J.D. and Z.X.; funding acquisition, J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62192784 and No. 62172056).

Data Availability Statement: The data presented in this study are openly available in [3,30,32].

Acknowledgments: The authors would like to thank all anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions, which helped us improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; Geng, J. Meta-gnn: On few-shot node classification in graph meta-learning. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2357–2360.

2. Wang, N.; Luo, M.; Ding, K.; Zhang, L.; Li, J.; Zheng, Q. Graph Few-shot Learning with Attribute Matching. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 1545–1554.
3. Ding, K.; Wang, J.; Li, J.; Shu, K.; Liu, C.; Liu, H. Graph prototypical networks for few-shot learning on attributed networks. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 295–304.
4. Cai, H.; Zheng, V.W.; Chang, K.C.C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1616–1637. [[CrossRef](#)]
5. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. *arXiv* **2017**, arXiv:1706.02216.
6. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph neural networks for social recommendation. In Proceedings of the the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 417–426.
7. Song, W.; Xiao, Z.; Wang, Y.; Charlin, L.; Zhang, M.; Tang, J. Session-based social recommendation via dynamic graph attention networks. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, 11–15 February 2019; pp. 555–563.
8. Yu, W.; Yu, M.; Zhao, T.; Jiang, M. Identifying referential intention with heterogeneous contexts. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 962–972.
9. Zhao, T.; Ni, B.; Yu, W.; Jiang, M. Early anomaly detection by learning and forecasting behavior. *arXiv* **2020**, arXiv:2010.10016.
10. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the 6th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
11. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2017**, arXiv:1710.10903.
12. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? *arXiv* **2018**, arXiv:1810.00826.
13. Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K. Simplifying graph convolutional networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6861–6871.
14. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
15. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the 5th International Conference on Learning Representations, ICLR, Toulon, France, 24–26 April 2017; (*Oral*).
16. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv* **2017**, arXiv:1707.09835.
17. Mishra, N.; Rohaninejad, M.; Chen, X.; Abbeel, P. A Simple Neural Attentive Meta-Learner. In Proceedings of the 6th International Conference on Learning Representation, Vancouver, BC, Canada, 30 April–3 May 2018.
18. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3630–3638.
19. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical networks for few-shot learning. *arXiv* **2017**, arXiv:1703.05175.
20. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
21. Belghazi, M.I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; Hjelm, R.D. Mine: Mutual information neural estimation. *arXiv* **2018**, arXiv:1801.04062.
22. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E* **2004**, *69*, 066138. [[CrossRef](#)] [[PubMed](#)]
23. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv* **2018**, arXiv:1808.06670.
24. Velickovic, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep Graph Infomax. *ICLR Poster* **2019**, *2*, 4.
25. Sun, F.Y.; Hoffmann, J.; Verma, V.; Tang, J. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv* **2019**, arXiv:1908.01000.
26. Mavromatis, C.; Karypis, G. Graph infoclust: Leveraging cluster-level node information for unsupervised graph representation learning. *arXiv* **2020**, arXiv:2009.06946.
27. Park, C.; Han, J.; Yu, H. Deep multiplex graph infomax: Attentive multiplex network embedding using global information. *Knowl.-Based Syst.* **2020**, *197*, 105861. [[CrossRef](#)]
28. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015; Volume 2.
29. Linsker, R. Self-organization in a perceptual network. *Computer* **1988**, *21*, 105–117. [[CrossRef](#)]
30. Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective classification in network data. *AI Mag.* **2008**, *29*, 93. [[CrossRef](#)]
31. Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; Su, Z. Arnetminer: Extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2009; pp. 990–998.
32. McAuley, J.; Pandey, R.; Leskovec, J. Inferring networks of substitutable and complementary products. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 785–794.

33. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
34. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.