Toward Effective Semi-supervised Node Classification with Hybrid Curriculum Pseudo-labeling

XIAO LUO, Department of Computer Science, University of California, Los Angeles, USA
WEI JU, YIYANG GU, and YIFANG QIN, School of Computer Science, Peking University, China
SIYU YI, School of Statistics and Data Science, Nankai University, China
DAQING WU, School of Mathematical Sciences, Peking University, China
LUCHEN LIU and MING ZHANG, School of Computer Science, Peking University, China

Semi-supervised node classification is a crucial challenge in relational data mining and has attracted increasing interest in research on graph neural networks (GNNs). However, previous approaches merely utilize labeled nodes to supervise the overall optimization, but fail to sufficiently explore the information of their underlying label distribution. Even worse, they often overlook the robustness of models, which may cause instability of network outputs to random perturbations. To address the aforementioned shortcomings, we develop a novel framework termed Hybrid Curriculum Pseudo-Labeling (HCPL) for efficient semi-supervised node classification. Technically, HCPL iteratively annotates unlabeled nodes by training a GNN model on the labeled samples and any previously pseudo-labeled samples, and repeatedly conducts this process. To improve the model robustness, we introduce a hybrid pseudo-labeling strategy that incorporates both prediction confidence and uncertainty under random perturbations, therefore mitigating the influence of erroneous pseudo-labels. Finally, we leverage the idea of curriculum learning to start from annotating easy samples, and gradually explore hard samples as the iteration grows. Extensive experiments on a number of benchmarks demonstrate that our HCPL beats various state-of-the-art baselines in diverse settings.

CCS Concepts: • **Computing methodologies** → **Semi-supervised learning settings**; • **Theory of computation** → *Semi-supervised learning;*

Additional Key Words and Phrases: Graph neural network, semi-supervised learning, curriculum learning

ACM Reference format:
Xiao Luo, Wei Ju, Yiyang Gu, Yifang Qin, Siyu Yi, Daqing Wu, Luchen Liu, and Ming Zhang. 2023. Toward Effective Semi-supervised Node Classification with Hybrid Curriculum Pseudo-labeling. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 3, Article 82 (November 2023), 19 pages.
https://doi.org/10.1145/3626528

This article is partially supported by the National Natural Science Foundation of China under NSFC Grants No. 62106008, No. 62276002, and 62306014, as well as the China Postdoctoral Science Foundation under Grant No. 2023M730057.

Authors' addresses: X. Luo, Department of Computer Science, University of California, Los Angeles 90095; e-mail: xiaoluo@cs.ucla.edu; W. Ju (Corresponding author), Y. Gu, Y. Qin, L. Liu, and M. Zhang (Corresponding author), School of Computer Science, Peking University, Beijing 100871, China; e-mails: {wudq, yiyanggu, qinyifang, liuluchen, mzhang_cs}@pku.edu.cn; S. Yi, School of Statistics and Data Science, Nankai University, Tianjin 300071 , China; e-mail: siyuyi@mail.nankai.edu.cn; D. Wu, School of Mathematical Sciences, Peking University, Beijing 100871, China; e-mail: juwei@pku.edu.cn.

1551-6857/2023/11-ART82
https://doi.org/10.1145/3626528

1 INTRODUCTION

With the increasing popularity of cloud computing technologies and the Internet of Things, as well as the expansion of social media, structured data is rising at an unprecedented rate. A graph is an effective and powerful tool for representing a large number of relational data across various domains including biology, social networks, information science, and so on [25]. Thus, the exploration of graph-structured data is very critical and necessary. In recent years, **graph neural networks (GNNs)** have been proposed and shown incredible performance in studying graph-structured data. Typically, GNNs are capable of combining vertex attributes and graph topology information to learn vertex representations for a variety of downstream graph-based tasks, including node classification [29, 31, 54, 79], graph classification [27, 35], graph clustering [26, 70], link prediction [7, 73], and traffic forecasting [23, 37]. Here in this article, we study semi-supervised node classification, which aims to forecast the categories of unlabeled nodes using a limited number of labeled nodes.

Indeed, various GNN algorithms for semi-supervised node classification have been developed [21, 29, 52], the bulk of which rely on the construction of diverse neighborhood propagation strategies for learning effective node representations. The most prominent technique is **graph convolutional networks (GCNs)** [29], which aggregate node representations of their local neighbors iteratively. Following GCNs, a number of graph convolutions have been developed sequentially using a variety of message passing algorithms. For example, **Graph Attention Network (GAT)** [52] integrates the attention mechanism into the message passing process that enables feature information to be passed adaptively. By eliminating nonlinearities and compressing weight matrices between successive layers, **Simple Graph Convolution (SGC)** [60] reduces the computational cost of a GCN. Hamilton et al. [21] propose to sample node neighborhoods for higher efficiency. **Graph Isomorphism Network (GIN)** [66] further improves the expressive capability of GCNs and is capable of capturing different graph structural information. GNNs have also been applied to various applications such as multi-view learning [69] and recommendation [40, 67].

Despite the remarkable performance in semi-supervised node classification, existing approaches go through two critical constraints that may impair model performance. On the one hand, these methods typically leverage the unlabeled nodes when propagating their attributes during the message passing process using GNNs, while ignoring the information of underlying label distribution. This problem may lead to easy underfitting, particularly in the absence of adequate annotated labels, thus limiting the performance of the network [54]. On the other hand, they usually pay less attention to the robustness of the model. Existing GNNs often have predefined attributes and neighborhood propagation patterns, which leads to each node being extremely reliant on its initial features and neighbors. When networks are attacked by noise in real-world applications [75], they may output unstable predictions, leading to considerable performance deterioration.

Numerous semi-supervised learning algorithms have been comprehensively investigated in recent years for making full use of the unlabeled datum [9, 30, 41, 50, 51]. Pseudo-labeling [30] is one of the most classic methods in this field. It needs a predictor to iteratively output the categories of unlabeled samples and involve well-classified examples in the training dataset. On this basis, further works usually encourage the classifier to make predictions with a small entropy on unlabeled samples [20]. Pseudo-labeling techniques have a lot of downstream applications in various tasks. For example, FixMatch [49] combines pseudo-labeling and consistency regularization to address the problems of label scarcity for image classification. **Cross Pseudo Supervision (CPS)** [9] introduces pseudo-labeling techniques into semantic segmentation problems. However, pseudo-labeling techniques are predominantly studied in the visual domains but have not been well applied to effectively solving node classification problems on graphs yet. Therefore, it is promising to explore unlabeled nodes on the graph with semi-supervised techniques.

Toward this end, this article develops a simple but effective approach called **Hybrid Curriculum Pseudo-Labeling (HCPL)** for semi-supervised node classification. The core of our idea is to sufficiently explore the unlabeled data through a pseudo-labeling strategy. Specifically, we iteratively annotate unlabeled nodes by training a model on the labeled samples and any previously pseudo-labeled samples, and repeat the process in a self-training way. Moreover, we not only involve a perturbed GNN predictor with the adaptive decoupling of the representation transformation and neighborhood propagation, but also introduce a hybrid pseudo-labeling strategy to increase the robustness under noise attack. We take both prediction confidence and uncertainty into account while dealing with noise, alleviating the impact of potential erroneous pseudo-labels. Note that curriculum learning [4] is a training strategy that trains a learning model from easier data to harder data. Inspired by this, we utilize the idea of curriculum learning to annotate easy samples with high confidence and robustness and then annotate hard samples with less confidence and robustness. This strategy learns the model in a meaningful order and helps the model free from error accumulation. In this way, our HCPL can sufficiently explore the unlabeled data through a pseudo-labeling strategy. Experimental results on several popular benchmark datasets demonstrate our HCPL outperforms a wide range of state-of-the-art approaches. To sum up, the contributions of this work are as follows:

- We develop a novel approach named HCPL for semi-supervised node classification, which leverages curriculum learning to produce confident pseudo-labels to make full use of the abundant unlabeled data while existing works usually do not explore semantic information in unlabeled data.
- We study the model robustness and introduce a novel hybrid pseudo-labeling strategy that takes into consideration both prediction confidence and prediction uncertainty to produce accurate pseudo-labels.
- Extensive experiments on six graph datasets show that HCPL achieves remarkable performance compared with a variety of state-of-the-arts in different settings.

2 RELATED WORK

2.1 GNNs

Recent years have witnessed increasing attention in research to apply deep learning methods to graph-structured data. GNNshave come into the spotlight due to their superior capability for graph representation learning [32, 34, 47, 55, 64] with wide applications such as fake new detection [24, 68]. Pioneering efforts use spectral-based approaches for localized and effective graph convolution. GCN simplifies GNN into spatial-based models [29], which utilizes the adjacent matrix and increases the effectiveness of GNNs. These spatial-based approaches are typically based on a neighborhood aggregation mechanism that updates node representation via the aggregation of information from its neighbors. Following that, several GCN variants have been developed subsequently, including GAT [52], SGC [60], and GIN [66]. GAT incorporates the attention mechanism to assess the importance of different neighbors on the center node and uses the attention scores as feature aggregation weights. Inspired by the Weisfeiler-Lehman algorithm [46], GIN [66] improves the expressive capability of GCNs via capturing different graph structures. However, the majority of GNN-based methods train the model using the cross entropy on labeled nodes to optimize the model, which neglect the abundant unlabeled nodes. In contrast, our research builds on the strength of GNNs and focuses on enhancing semi-supervised node classification via curriculum learning in a pseudo-labeling way. GNNs can also be applied to various settings such as PU learning [61], open-world learning [62], unsupervised domain adaptation [63], and cross-modal retrieval [39]. For example, LSDAN [61] incorporates the attention mechanism with GNN to model node significance from both short and long terms. OpenWGL [62] utilizes a variation graph autoencoder to explore unseen nodes in the test set. GCLN [63] models both attraction and repulsion forces for consistency learning within a single graph and across graphs. DAGNN [39] adopts multi-hop GNN to investigate the relationship between labels for effective cross-modal retrieval.

2.2 Semi-supervised Learning

Semi-supervised learning (SSL) has recently drawn a lot of interest and achieved a lot of success in a variety of fields. SSL is capable of reducing the need for labeled data by using a vast volume of unlabeled data. Because unlabeled data can be quickly obtained with minimal human effort, the performance of models may be improved at a low cost using SSL. Two mainstream methodologies of SSL are self-training and consistency regularization. The pioneer semi-supervised learning works are based on self-training (so-called pseudo-labeling) [8, 30, 48, 51], which uses the class predictions of models as pseudo-labels to train unlabeled data in a supervised way. Specifically, unlabeled samples are iteratively added to the training data by annotating them with a weak model trained with labeled data. Another line of SSL is based on recent breakthroughs in consistency learning [5, 49], which encourages the network to make consistent predictions when it comes to noise perturbation on unlabeled samples. Semi-supervised learning techniques have been extensively utilized in a variety of fields such as computer vision and knowledge mining [1, 9, 13, 43, 49]. Inspired by recent advances in visual domains, our proposed HCPL effectively combines curriculum learning and semi-supervised learning, and develops a novel pseudo-labeling strategy for effective semi-supervised node classification on graphs. Our work is also related to **teaching-to-learn and learning-to-teach (TLLT)** [15], which also adopt an easy-to-hard curriculum strategy for graph-based semi-supervised learning. TLLT aims to perform the message propagation based on the teacher module and choose the following samples from the learner module in an alternative manner. ML-TLLT [18] introduces this framework to solve the problem of multi-labeled learning, which studies every possible label for the unlabeled samples along with the label dependencies. SMMCL [19] incorporates curriculum learning to learn the procedure of label propagation on graphs. MMCL [17] leverages curriculum learning to acquire the difficulty of accurately classifying each unlabeled sample for semi-supervised image classification. Gong et al. [16] adopt TLLT into the saliency detection, which successfully identifies the salient objects in images with high propagation quality.

2.3 Graph-based Semi-supervised Learning

Semi-supervised node classification is the most fundamental problem in graph data mining [14, 29, 33, 56, 59], which has various applications in social analysis [42], bioinformatics [65], image annotation [76], text generation [58], and noise cleaning [74]. For example, ASFS [72] explores the pairwise relationships in the latent and then utilizes the graph structure to guide semantics learning in a semi-supervised setting. This work utilizes a classic feature selection framework while ours adopts GNNs for semi-supervised node classification. For semi-supervised node classification, only a few annotation nodes are available in the graph to predict the labels of the remaining nodes. Traditional methods of solving this problem are typically based on graph Laplacian regularizations [3, 36, 77]. For example, Belkin et al. [3] propose to exploit the geometry of the marginal distribution and give a new form of regularization. Recently, GNNs have emerged as a powerful approach to learn from the graph. However, existing GNN methods usually focus on developing effective message passing patterns [29, 60], but neglect to sufficiently exploit the information of unlabeled nodes. Our article, by contrast, tackles this issue through effective pseudo-labeling for better semi-supervised node classification. We believe our method can be extended to tackle various graph-related semi-supervised tasks [12].

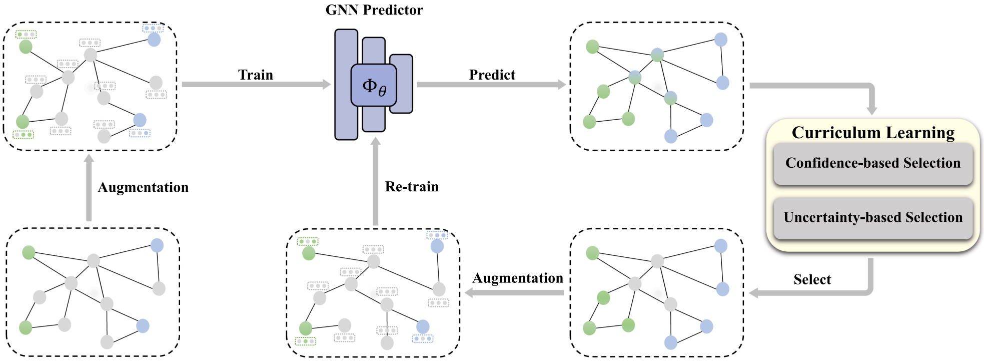


Fig. 1. Illustration of our proposed approach HCPL. The model is first trained under the supervision of labeled nodes. Then, the model is used to predict and assign pseudo-labels for unlabeled nodes. We use curriculum learning to iteratively select a subset of pseudo-labeled samples based on a hybrid selection strategy, and add them to the labeled set. Afterward, a new model is re-trained under the supervision of the expanded labeled data. The process will stop when all data is exhausted during iterative training.

3 METHOD

To begin with, we introduce the problem definition and present our approach HCPL for semi-supervised node classification on graphs. Previous methods usually neglect the label information contained in unlabeled nodes as well as the robustness of the model. To tackle the issues, our approach is based on the exploration of unlabeled data via pseudo-labeling. Namely, we annotate unlabeled nodes by training a model on labeled samples and any previously pseudo-labeled data and then repeating this procedure in a self-training fashion. Specifically, we first propose a perturbed GNN predictor and present a hybrid pseudo-labeling technique, which takes both prediction confidence and uncertainty into account under noise attack. Finally, an optimization pipeline embraced with curriculum learning is used to dynamically and automatically select pseudo-labels. The overall framework can be illustrated in Figure 1.

3.1 Problem Formulation

A graph is represented in a form of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ denotes a set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes a set of edges in the graph. $\boldsymbol{x}_i \in \mathbb{R}^F$ denotes the attribute feature of node v_i, where F is the dimension number of the node attributes. In addition, each node v_i is associated with a label vector, i.e., $\boldsymbol{y}_i \in \{0,1\}^C$ where C is the number of label categories. In our case, $M (M < N)$ nodes in $\mathcal{V}^L$ are annotated with their labels, while the labels of the other $N - M$ nodes are unknown. Our aim is to predict the unobserved labels for unlabeled nodes in $\mathcal{V}^U$ in a graph. Take the social network CORA as an example. Each node v_i corresponds to a research paper, and two research papers (i.e., v_i and v_j) are linked if the paper v_i is cited by the paper v_j. There are seven classes of topics, i.e., reinforcement learning, neural networks, case-based research, genetic algorithms, probabilistic methods, rule learning, and theory. We need to classify all papers without labels into their associated classes.

3.2 Perturbed GNN Predictor

Here, we present a perturbed GNN as the backbone of our HCPL. GNNs have been frequently utilized to collect node attributive information as well as topological information on graphs using deep neural networks. We begin with the introduction of the popular message passing procedure. In formulation, the embedding of node $v_i \in V$ at the layer k is represented as $\boldsymbol{h}_i^{(k)}$. The message passing procedure in GNNs usually involves two steps: (i) Aggregation step, which collects the semantic information from the neighborhood of node v_i at the previous layer $k-1$; and (ii) Combination step, which merges the node embedding of v_i at the previous layer with the obtained neighbor embedding at the current layer. To summarize,

$$\begin{aligned} \boldsymbol{h}_{\mathcal{N}(v_i)}^{(k)} &= AGG_{\theta}^{(k)}\left(\left\{\boldsymbol{h}_j^{(k-1)}\right\}_{v_j \in \mathcal{N}(v_i)}\right), \\ \boldsymbol{h}_i^{(k)} &= COM_{\theta}^{(k)}\left(\boldsymbol{h}_i^{(k-1)}, \boldsymbol{h}_{\mathcal{N}(v_i)}^{(k)}\right), \end{aligned} \tag{1}$$

where $\mathcal{N}(v_i)$ denotes the neighborhood of v_i, and $AGG_{\theta}^{(k)}$ and $COM_{\theta}^{(k)}$ represent the aggregation and combination operators at the layer k, respectively. After performing neighborhood aggregation K times, the embedding vectors at all layers are condensed into a single embedding vector:

$$\boldsymbol{h}_i = SUM_{\theta}\left(\left\{\boldsymbol{h}_i^k\right\}_{k=1}^{K}\right), \tag{2}$$

where SUM_{θ} represents the summarization operator. Widely used mean aggregator, LSTM aggregator, and pooling aggregator [21] can also be utilized to generate informative and structure-aware representations for various downstream tasks.

In our implementation, we begin with a **Multi-Layer Perception (MLP)** to process the initial attributes. Formally, we have

$$\boldsymbol{z}_i = MLP(\boldsymbol{x}_i), \tag{3}$$

which will be concatenated into feature matrix $\boldsymbol{Z} \in \mathbb{R}^{|V| \times d}$ and d is the embedding dimension. Then, given the adjacent matrix $\boldsymbol{A} \in \mathbb{R}^{|V| \times |V|}$, we use the decoupled symmetrical normalization propagation to generate node embeddings at every layer.

$$\boldsymbol{H}_k = \widehat{\boldsymbol{A}}^k \boldsymbol{Z}, k = 1, 2, \ldots, K, \tag{4}$$

where $\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widehat{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$ and $\widehat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$. Finally, we use an attention mechanism [2, 33] to aggregate embeddings at all layers. Formally, the summarized node representation $\boldsymbol{h}_i$ for node v_i is written as

$$\begin{aligned} w_i^k &= \sigma\left(\boldsymbol{h}_i^k \boldsymbol{W}\right), \\ \boldsymbol{h}_i &= \frac{w_i^k \boldsymbol{h}_i^k}{\sum_{k'=1}^{K} w_i^{k'}}, \end{aligned} \tag{5}$$

where $\boldsymbol{h}_i^k$ is the representation of the i-th node at the layer k, $\sigma(\cdot)$ denotes an activation function, and $\boldsymbol{W}$ is a trainable matrix to acquire the weight.

Nonetheless, neighborhood propagation strategies are typically predefined in GNNs, leaving each node largely reliant on its attributes and neighbors. The neural network could be misled during graph convolution methods due to noise attacks on node properties and connection patterns. Here, we utilize two popular graph augmentation strategies [71] to make it easier to generate disturb-invariant node representations.

- **Attribute Masking:** We choose several vertices and mask a portion of their attributes afterward. The prior behind this strategy is that masking part of vertices will not change the semantics of the node much. It serves as the dropout strategy in the deep neural network.
- **Edge Deletion:** Certain edges are randomly dropped out of the graph based on an i.i.d uniform distribution. The strategy corresponds to the prior that the node semantics should be robust to the random attacks of edge connectivity patterns.

The augmented version of $\mathcal{G}$ is denoted as $\tilde{\mathcal{G}}$. After our GNN, we feed the representation $\boldsymbol{h}_i$ of each node into a two-layer MLP classifier to produce the prediction vector $\boldsymbol{p}_i \in \mathbb{R}^C$. Formally,

$$\boldsymbol{p}_i = MLP_\theta(\boldsymbol{h}_i), \tag{6}$$

where θ is the parameter of the GNN predictor.

3.3 Hybrid Pseudo-Label Selection

In our framework, we first optimize the model using labeled data and then employ the trained model to output the label distribution for unlabeled data. Furthermore, we seek to collect accurate pseudo-labels and add these unlabeled data into the training set. Previous methods [30] usually utilize the confidence scores for pseudo-labeling, which could generate biased and overconfident samples, and therefore result in error accumulation. Therefore, we propose a hybrid pseudo-label selection strategy based on both the confidence and uncertainty of the prediction, which will be elaborated as follows.

Confidence-based Selection. Intuitively, based on the label distribution, we seek to select hard samples with high confidence. Formally, let p_i^c denote the probability of the i-th example belonging to the class c, and the pseudo-label can be obtained as follows:

$$\tilde{y}_i^c = \mathbf{1}\left[p_i^c \geq \gamma\right], \tag{7}$$

where γ is a fixed threshold. Note that the use of hard labels is associated with entropy minimization [20], where the model prediction is enforced to be of low entropy on unlabeled samples.

Uncertainty-based Selection. However, the accuracy of pseudo-labels is still far from satisfactory, since it does not take the robustness of the network into consideration. From a different perspective, the prediction is unreliable if the predicted distribution is unstable to random attack [43]. At this inspirit, we re-run the perturbed GNN predictor for W times where W is the running number of the predictor, and calculate the uncertainty of pseudo-labels using the formulation of standard deviation. Let $\underline{p_i^c}$ denote the list of W predictions, and we have

$$\tilde{y}_i^c = \mathbf{1}\left[sd\left(\underline{p_i^c}\right) \leq \eta\right], \tag{8}$$

where η is another threshold and $sd(\cdot)$ denotes the standard error of the W predictions. In this way, we consider the robustness of our model by selecting pseudo-labels invariant to random attacks, and thus our model is more likely to produce correct pseudo-labels.

Finally, we combine the advantages of the two strategies by taking the intersection of selected pseudo-labels. Note that, since we forward the GNN predictor for W times, we use the mean of the prediction to replace the single output in Equation (7). In formulation,

$$\tilde{y}_i^c = \mathbf{1}\left[sd\left(\underline{p_i^c}\right) \leq \eta\right] \mathbf{1}\left[mean\left(\underline{p_i^c}\right) \geq \gamma\right], \tag{9}$$

where $mean(\underline{p_i^c})$ denotes the mean of the W predictions. In this way, the pseudo-labels with both high confidence and robustness will be selected, which greatly improves the accuracy of pseudo-labels. In summary, the motivation of our strategy is to evaluate the difficulty of classifying each sample accurately, which can guide sequential curriculum learning. Here, our hybrid pseudo-label selection strategy is introduced based on both the confidence and uncertainty of the prediction. On the one hand, we select samples with high maximal probabilities, for which the model has high confidence about the prediction. On the other hand, we re-run the perturbed GNN predictor and evaluate the variance of the predictions, which can reflect the uncertainty of the prediction. These uncertainty scores can help us to evaluate the difficulty from different views. Finally, we take the intersection of results from our hybrid strategy to detect reliable samples under both rules. Further

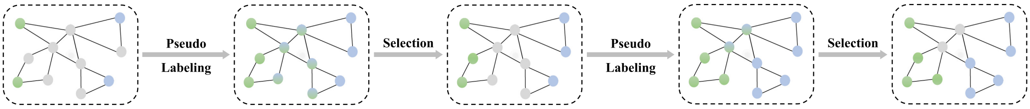


Fig. 2. An example of our optimization pipeline with curriculum learning.

ablation studies also validate the effectiveness of our hybrid strategy and we believe our method can be utilized in more semi-supervised settings such as cross-modal retrieval.

3.4 Optimization Pipeline with Curriculum Learning

In our framework, we first train the GNN predictor using labeled data. Specifically, we employ the standard cross-entropy loss to train labeled nodes on the augmented graphs. Formally,

$$\ell_s = -\frac{1}{|\mathcal{V}^L|}\sum_{x_i \in \mathcal{V}^L} \boldsymbol{y}_i^T \log \boldsymbol{p}_i. \tag{10}$$

Then, following the principle of self-training, we output the prediction for unlabeled data for W times, and then select reliable pseudo-labels based on Equation (9). These unlabeled nodes and their pseudo-labels are added to the labeled subset.

Note that a fixed threshold in selection is not optimal. For example, a large threshold γ may lead to too few pseudo-labels while a small value may bring in too many inaccurate pseudo-labels otherwise. As a result, we involve in a novel pipeline by adopting curriculum learning, resulting in dynamic thresholds in the selection for multiple iterations. To be specific, we gradually select more unlabeled samples from easy to difficult by increasing η and decreasing γ. In our implementation, we use the percentile of scores to decide the thresholds. Assume the total number of iterations is T and $argmax_{c \in C}\{mean(\underline{p_i^c})\} = c_i'$. For the t-th iteration, the thresholds are adjusted with

$$\begin{aligned} \gamma_t &= Percentile\left(\left\{mean\left(\underline{p_i^{c_i'}}\right)\right\}_{\boldsymbol{x}_i \in V^U}, 100 - 100/T * t\right), \\ \eta_t &= Percentile\left(\left\{sd\left(\underline{p_i^{c_i'}}\right)\right\}_{\boldsymbol{x}_i \in V^U}, 100/T * t\right), \end{aligned} \tag{11}$$

where $Percentile(S, m)$ denotes the values of the m-th percentile of set S. Then, we select pseudo-labels with dynamic thresholds as follows:

$$\tilde{y}_i^c = \mathbf{1}\left[sd\left(\underline{p_i^c}\right) \leq \eta_t\right] \mathbf{1}\left[mean\left(\underline{p_i^c}\right) \geq \gamma_t\right]. \tag{12}$$

Note that at T-th iteration, all the unlabeled nodes will be exhausted, i.e., annotated in self-training. An example is illustrated in Figure 2. Through curriculum learning [8], we start from easy samples with high confidence and low uncertainty, and gradually explore hard unlabeled samples. We are also involved in two strategies. First, after each iteration, we restore the labeled set and re-annotate every node in the unlabeled set. This enables pseudo-annotated nodes to enter or leave the updated set. Second, we train the GNN predictor from scratch, i.e., reinitialize the parameters in the GNN predictor after each iteration instead of popular fine-tuning. These two strategies can discourage concept drift or confirmation bias introduced at the early stage of self-training to be accumulated, improving the performance of our proposed HCPL. The whole pipeline of the optimization process is illustrated in Algorithm 1.

4 EXPERIMENTS

In this part, by conducting extensive experiments on six real-world datasets to show the effectiveness of our HCPL, we highlight the following results:

ALGORITHM 1: Training pipeline for our HCPL

```
Input: Attribute feature set $\{\boldsymbol{x}_i\}_{x_i \in \mathcal{V}}$; Graph edge set $\mathcal{E}$; Set of labeled nodes $\mathcal{V}^L$; Stepping times $T$;
Output: Prediction result $\{\boldsymbol{p}_i\}_{x_i \in \mathcal{V}^U}$.
Train GNN predictor using $\mathcal{V}^L$ only;
for t = 1 to $T$ do
    Forward the networks for $W$ times to get the mean together with the standard deviation of the predictions for all nodes in $\mathcal{V}^U$, i.e., $mean(p_i^c)$ and $sd(p_i^c)$
    Calculate the percentiles $r_1$ and $r_2$ for thresholds $\gamma_t$ and $\eta_t$, respectively, with Equation (12)
    Restore current labeled set $\tilde{\mathcal{V}}^L$ using $\mathcal{V}^L$
    for $x_i \in \mathcal{V}^U$ do
        if $\exists c, mean(p_i^c) > \gamma_t$ and $sd(p_i^c) < \eta_t$ then
            $\tilde{\mathcal{V}}^L \leftarrow \tilde{\mathcal{V}}^L \cup (x_v, c)$
        end if
        Train GNN model using updated $\tilde{\mathcal{V}}^L$ from scratch
    end for
end for
```

- HCPL significantly outperforms all competing baselines that are compared to all experimental settings.
- Ablation studies demonstrate the efficiency of the different components of HCPL.
- The performance of our methods is stable to main hyper-parameters in proper ranges.
- Our HCPL is robust to random attack compared with baselines.

4.1 Experimental Setup

Datasets. Our HCPL is accessed on six widely used benchmark node classification datasets including three paper citation datasets [6, 44], i.e., Cora, CiteSeer, and PubMed, two purchasing graph datasets [45], i.e., Amazon Computers and Amazon Photo, and one co-author network dataset [45], i.e., Coauthor CS. In three paper citation datasets, nodes denote publication and edges denote citation links. The purpose is to classify these nodes into different areas. Both purchasing graph datasets are collected from Amazon, where nodes represent goods and edges are constructed when two goods are often bought at the same time. CoauthorCS is a co-author network dataset where nodes denote authors and edges indicate co-author relationships. The statistics of these datasets are summarized in Table 1.

We utilize the same splits in the previous work [54] to construct train/validation/test datasets for three citation datasets, while for the other three datasets, we randomly choose 30 nodes from each category as labeled training data, 30 nodes as validation data, and other nodes as the test data. For a fair comparison, we adopt the same dataset splits on all datasets for all baseline methods.

Compared Methods. To evaluate the effectiveness of our developed HCPL, we compare it with the following state-of-the-art baseline models for semi-supervised node classification as follows.

- *Chebyshev* [10]: It is a formulation of CNNs that leverages the idea of spectral graph theory to devise fast localized convolutional filters suitable for graph data.
- *GCN* [29]: It is a classic semi-supervised GNN model based on the spectral theory that generates node representations via aggregating information from neighbors.
- *GAT* [52]: It is a GNN model that improves GCN by incorporating the attention mechanism to assign different weights to each neighboring node of a node.

Table 1. Statistics of Six Datasets

Dataset	#Nodes	#Edges	#Features	#Classes	Edge density	Type
Cora	2,708	5,278	1,433	7	0.0004	Citation
CiteSeer	3,327	4,552	3,703	6	0.0004	Citation
PubMed	19,717	44,324	500	3	0.0001	Citation
Amazon Computers	13,752	245,861	767	10	0.0007	Co-purchase
Amazon Photo	7,650	119,081	745	8	0.0011	Co-purchase
Coauthor CS	18,333	81,894	6,805	15	0.0001	Coauthor

- *SGC* [60]: It is a fast algorithm that lowers the unnecessary computational cost of GCN via removing the nonlinearity between layers and compressing the weight matrix.
- *DGI* [53]: It is an unsupervised approach for learning node representations, which focuses on the mutual information between node-level representations and their associated graph-level representations.
- *GMI* [38]: It presents a new method for measuring the similarity degree between input graphs and hidden node embeddings, generalizing the concept of mutual information computations to the graph domain.
- *MVGRL* [22]: It introduces a self-supervised approach, which learns node-level and graph-level representations by maximizing mutual information between representations encoded from different topological views of graphs.
- *GRACE* [78]: It is a novel framework based on contrastive learning for unsupervised graph representation learning via a hybrid scheme for generating graph views on both topology and feature levels.
- *CG^3* [54]: It is a novel GCN-based semi-supervised learning algorithm that enriches the label information via leveraging node similarities and structural knowledge from two different perspectives.
- *AM-GCN* [57]: It fuses multi-view information from topological structures and features using the attention mechanism.

Parameter Settings. We implement all the compared methods using PyTorch 1.8.0 and Pytorch Geometric 1.7.2, which are capable of smoothly training GNNs for a range of applications connected to graph-structured data. Extensive experiments are performed on an NVIDIA GeForce GTX 1080 Ti. For simplicity, we adopt a two-layer GCN [29] as the GCN backbone as default and include a model variant HCPL-A that utilizes GAT [52] as the backbone. The dimension number of hidden embedding is set to 256 for all datasets and the number of iterations is set to 20. These two hyper-parameters will be discussed in Section 4.4. Adam [28] is employed during optimization due to its effectiveness. We set the learning rate to 0.01 and it decays with the rate 0.0005. For all experiments, we present the mean accuracy with standard deviations from five runs. The validation dataset is utilized to tune all hyper-parameters, and the test dataset can provide the final results. For the parameters in the adopted baselines, we refer to their original papers and utilize their tuning strategies for the best performance.

4.2 Experimental Results

Table 2 displays the compared results on six datasets. From the table, the following observations can be obtained:

- GCN-based algorithms (i.e., GCN, GAT, and SGC) overall perform better than the traditional method (i.e., Chebyshev), which shows that the superior representation-learning ability of GCN helps to enhance the performance for semi-supervised node classification.

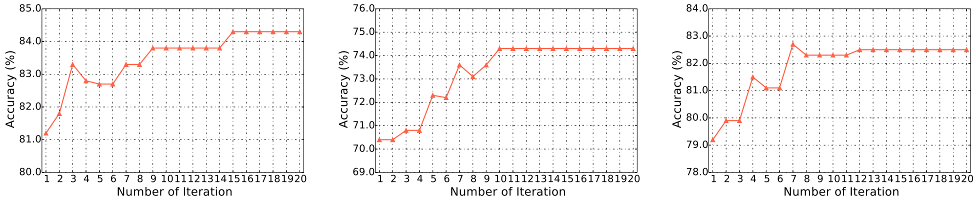
Table 2. Results on Six Datasets in Terms of Accuracy (in %) Over Five Runs

Methods	Cora	CiteSeer	PubMed	Amazon Computers	Amazon Photo	Coauthor CS
Chebyshev [10]	80.7 ± 0.2	70.2 ± 0.6	77.4 ± 0.1	72.5 ± 0.0	88.4 ± 0.1	90.4 ± 0.2
GCN [29]	81.3 ± 0.4	71.5 ± 0.2	78.8 ± 0.6	77.7 ± 0.7	88.1 ± 0.8	91.6 ± 0.7
GAT [52]	82.7 ± 0.1	70.7 ± 0.4	78.5 ± 0.2	79.5 ± 0.2	88.0 ± 0.6	91.2 ± 0.5
SGC [60]	77.7 ± 0.0	72.6 ± 0.0	76.4 ± 0.0	74.8 ± 0.1	87.9 ± 0.1	90.2 ± 0.2
DGI [53]	80.9 ± 0.3	71.4 ± 0.2	76.3 ± 1.1	77.7 ± 0.8	85.3 ± 0.9	90.6 ± 0.5
GMI [38]	81.6 ± 0.4	71.9 ± 0.5	81.8 ± 0.4	78.9 ± 0.1	84.9 ± 0.0	90.7 ± 0.0
MVGRL [22]	81.3 ± 0.4	71.9 ± 0.1	79.3 ± 0.1	79.5 ± 0.8	88.1 ± 0.2	91.7 ± 0.1
AM-GCN [57]	81.0 ± 0.3	72.8 ± 0.4	OOM	80.9 ± 0.7	91.3 ± 0.2	OOM
GRACE [78]	82.8 ± 0.3	71.3 ± 0.7	79.0 ± 0.2	75.1 ± 0.1	83.2 ± 0.1	91.2 ± 0.2
CG^3 [54]	83.5 ± 0.3	73.7 ± 0.2	79.2 ± 0.6	80.5 ± 0.1	90.0 ± 0.2	92.4 ± 0.1
HCPL (Ours)	84.2 ± 0.6	**74.4 ± 0.7**	**82.4 ± 0.7**	82.2 ± 0.8	92.3 ± 0.5	**93.2 ± 0.4**
HCPL-A (Ours)	**84.5 ± 0.4**	73.6 ± 0.5	81.6 ± 0.4	**83.4 ± 1.2**	**92.6 ± 0.7**	92.5 ± 0.3

OOM means out-of-memory.

- The methods (i.e., DGI, GMI, MVGRL, GRACE, CG^3, and HCPL) that explore the representations or label distribution of unlabeled data perform better than other methods, showing that utilizing additional unlabeled datum by unsupervised or semi-supervised learning is an important complement to supervised learning, enhancing model performance.
- In all of the datasets, our approach produces the best results. In particular, on the large-scale datasets Amazon Computers and Amazon Photo, our HCPL outperforms the best baseline CG^3 by 2.1% and 2.6%, respectively, validating the efficiency of our HCPL. We claim this improvement can be attributed to two reasons: (i) Our hybrid pseudo-labeling strategy incorporates both prediction confidence and uncertainty to generate accurate pseudo-labels for unlabeled nodes. (ii) Our curriculum learning pipeline gradually explores unlabeled samples to avoid overconfident annotations.
- We have conducted one-sample paired t-tests to justify that the improvements with the best baseline are statistically significant with p-value < 0.05 on all the datasets. However, the variance of our HCPL is a little larger than that of baselines on several datasets. A potential reason is that in some cases, wrong pseudo-labels could make the performance a little unstable when studying unlabeled nodes. In practice, we suggest running multiple times and selecting the best model based on validation datasets. We calculate the classification accuracy on all test nodes.
- The performance improvement compared with the best baseline (CG^3) is limited in Cora. The potential reason could be the high homophily ratio of Cora and thus low risk of biased pseudo-labeling, which makes curriculum learning less important. In practice, we can measure the risk of biased pseudo-labeling by active learning and then design the algorithm accordingly.

Moreover, we can validate that GAT can still benefit from our hybrid curriculum pseudo-labeling by comparing GAT and HCPL-A. Moreover, our HCPL-A can perform better than all the baselines in most cases, which validates our superiority again. Of note, our HCPL is trained in an iterative way. Hence, we access the accuracy of our HCPL at each iteration to observe whether the performance will improve as the number of iterations grows under curriculum learning. The results of the three datasets are shown in Figure 3. We can observe that the performance increases in most cases after each iteration, which validates that our proposed HCPL benefits from pseudo-labeling and curriculum learning.



(a) Cora (b) CiteSeer (c) PubMed

Fig. 3. Results on three datasets Cora, CiteSeer, and PubMed *w.r.t.* the number of iteration. We can observe that model performance improves as the number of iterations grows before saturation in most cases.

Table 3. Results on Cora Dataset with Different Label Rates in Terms of Classification Accuracies (in %)

Label Rate	0.5%	1%	2%	3%	5%	10%	20%	50%
Chebyshev	37.9	59.4	73.5	76.1	80.7	82.6	82.4	82.9
GCN	47.8	63.9	72.7	76.4	81.3	82.1	85.0	86.5
GAT	57.1	70.9	74.3	78.2	82.7	83.4	85.3	87.2
SGC	48.4	66.5	69.7	73.9	77.7	78.9	81.2	79.9
DGI	68.0	73.4	76.7	78.3	80.9	81.2	81.3	81.6
GMI	67.8	71.6	75.5	77.6	81.6	84.0	84.2	84.7
MVGRL	57.6	67.6	76.2	77.8	81.3	83.8	84.5	84.9
GRACE	63.8	73.5	75.2	76.2	82.8	83.6	84.4	85.9
CG^3	68.1	74.2	77.3	79.1	83.5	84.3	85.1	86.6
HCPL (Ours)	**71.7**	**75.4**	**78.2**	**81.1**	**84.2**	**84.9**	**86.3**	**88.4**

Table 4. Results on CiteSeer Dataset with Different Label Rates in Terms of Classification Accuracies (in %)

Label Rate	0.5%	1%	2%	3%	5%	10%	20%	50%
Chebyshev	34.0	58.3	64.6	67.2	71.3	71.7	72.2	75.7
GCN	47.6	55.8	65.3	69.2	71.7	72.6	73.4	77.6
GAT	53.2	63.9	68.3	69.5	71.2	72.1	75.1	79.0
SGC	46.8	59.3	67.1	68.6	72.7	73.0	74.5	78.8
DGI	61.0	65.8	67.5	68.8	71.6	72.3	73.1	76.5
GMI	54.4	63.5	66.7	68.5	72.5	74.8	75.0	75.9
MVGRL	61.3	65.1	68.5	70.3	71.2	72.8	73.1	74.8
GRACE	61.8	62.5	70.7	71.4	71.9	73.0	74.2	76.6
CG^3	62.9	70.1	70.9	71.7	73.9	74.5	74.8	77.2
HCPL (Ours)	**64.4**	**71.4**	**71.9**	**73.0**	**74.6**	**75.1**	**75.7**	**80.4**

Further, we experiment in the cases where the labeled samples are changed to access the performance of the HCPL in handling different supervision. We choose a proportion of labeled samples for model training in each run following [54]. We first choose the Cora dataset as an example where the label rates are varying in 0.5%, 1%, 2%, 3%, 5%, 10%, 20%, and 50%. The result is summarized in Table 3. Again, we can see that our HCPL consistently beats other baselines in different settings, demonstrating the superiority of our HCPL in tackling scarce supervision. We also conduct similar experiments on datasets CiteSeer and PubMed following the setting (i.e., label rate) in [54]. The result is shown in Tables 4 and 5 and similar results can be detected in two datasets.

Table 5. Results on PubMed Dataset with Different Label Rates in Terms of Classification Accuracies (in %)

Label Rate	0.03%	0.05%	0.1%	0.3%	0.5%	3%	10%
Chebyshev	58.9	67.2	71.5	77.4	80.1	82.1	82.9
GCN	61.3	65.6	72.3	78.8	80.8	83.9	86.1
GAT	62.8	66.7	71.1	78.5	80.1	83.6	84.8
SGC	61.0	64.3	68.5	76.4	77.8	78.6	79.2
DGI	61.5	66.2	71.4	76.3	79.9	80.2	80.4
GMI	58.7	65.2	76.3	81.8	82.5	83.2	83.7
MVGRL	60.3	67.3	73.4	79.3	81.9	82.7	83.6
GRACE	64.9	68.6	73.6	79.0	80.4	81.4	82.5
CG^3	67.0	71.1	74.5	79.2	81.7	82.3	82.9
HCPL (Ours)	**70.9**	**74.0**	**77.6**	**82.4**	**82.8**	**84.7**	**87.2**

Table 6. Comparison with Variants for Ablation Study (in %)

Methods	Cora	CiteSeer	PubMed	Amazon Computers	Amazon Photo	Coauthor CS
HCPL w/o aug	83.5 ± 0.7	73.4 ± 1.1	82.1 ± 0.7	81.6 ± 0.4	91.8 ± 0.4	92.9 ± 0.3
HCPL w/o cur	83.2 ± 0.6	73.2 ± 0.5	81.3 ± 0.6	81.4 ± 0.6	91.6 ± 0.3	92.6 ± 0.5
HCPL - inv cur	82.8 ± 0.8	72.5 ± 0.9	80.9 ± 1.1	80.8 ± 1.5	91.3 ± 0.8	92.1 ± 0.7
HCPL - random	83.5 ± 0.7	73.4 ± 0.7	81.5 ± 0.9	81.7 ± 1.0	91.6 ± 0.9	92.7 ± 0.5
HCPL w/o unc	83.9 ± 0.3	73.3 ± 0.8	81.5 ± 0.8	81.8 ± 0.7	91.9 ± 0.6	92.8 ± 0.7
HCPL (Ours)	**84.2 ± 0.6**	**74.4 ± 0.7**	**82.4 ± 0.7**	**82.2 ± 0.8**	**92.3 ± 0.5**	**93.2 ± 0.4**

4.3 Ablation Study

In this part, we perform extensive experiments over core components of the proposed HCPL. In particular, five model variants are compared with the full model, which only remove one part of our framework with the other components kept:

- HCPL w/o aug: We delete the perturbation over the input in the GNN-based predictor.
- HCPL w/o cur: We remove the curriculum strategy and annotate all the unlabeled nodes for self-training.
- HCPL w/o unc: We remove the uncertainty-based selection strategy and only use confidence to select pseudo-labels.
- HCPL - inv cur: We annotate from the hard examples to easy examples.
- HCPL - random: We annotate samples randomly during iterations.

The results are in Table 6. First, we can see a decline in the performance of HCPL w/o aug, demonstrating the necessity of augmentation strategies, which also improve the robustness of HCPL. Second, HCPL performs better than HCPL w/o cur and HCPL inv cur, validating that pseudo-labeling may introduce some biases to deteriorate the performance while our curriculum learning strategy is capable of releasing this issue. Moreover, we analyze the homophily ratio of each dataset, which denotes the fraction of edges that connect nodes from the same category in a graph. A lower homophily ratio will increase the challenges of semi-supervised learning under label scarcity, which makes curriculum learning more important. In particular, the homophily ratios for Cora and CiteSeer are 0.810, and 0.736, respectively. From ablation studies, it can be observed that when we remove curriculum learning, the performance will drop 1.18% and 1.61% for Cora and CiteSeer, respectively. This validates our analysis that the curriculum learning strategy is more suitable for

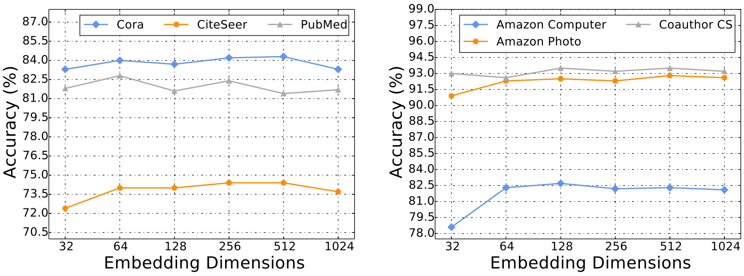


Fig. 4. Performance *w.r.t.* embedding dimension of hidden layers.

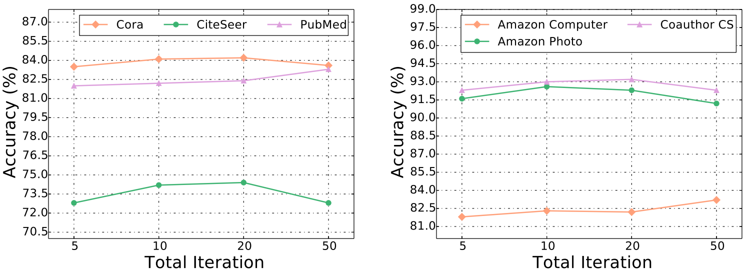


Fig. 5. Performance *w.r.t.* total iteration.

challenging tasks. Third, removing the uncertainty-based selection strategy leads to a decline in performance, which shows it can produce more accurate pseudo-labels with the consideration of model robustness. Fourth, although these model variants can still perform well with the effectiveness of the remaining components, we can always observe a decline in these model variants compared with the full model, which validates the effectiveness of every component.

4.4 Sensitivity Analysis

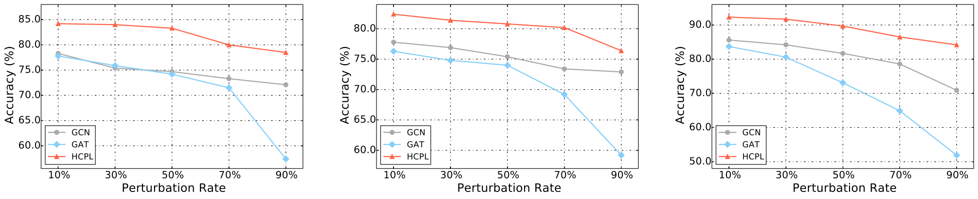
In this part, we study the sensitivity of hyper-parameters in HCPL, i.e., embedding dimension in the hidden layer and the total number of iterations, respectively.

We first study the influence of different hidden embeddings by varying the dimension in [32, 64, 128, 256, 512, 1024] with other settings fixed. We plot the result on all the datasets in Figure 4 and observe that the performance almost first increases and then stays stable as the embedding dimension grows. The potential reason is that a large hidden dimension would improve representation, but the model will tend to be saturated when the dimension is above a certain value.

Next, we study the effect of different numbers of iterations. Specifically, we fix all the other hyper-parameters and vary the iteration number in $\{5, 10, 20, 50\}$. The results are plotted in Figure 5. We can observe that in most cases increasing the iteration number leads to a gain in performance. Perhaps it is because a larger iteration number brings in fewer pseudo-labels at each iteration, which is usually reliable for self-training. However, a too-large number of iterations accompanies a higher computational cost. As a result, we set the number of iterations to 20 as the default.

4.5 Robustness Analysis

In this part, we test the robustness of our HCPL by perturbing the graph, i.e., deleting edges or masking node attributes at random. Figure 6 illustrates the performance of three methods (GCN, GAT, and HCPL) when varying the perturbation rate from 10% to 90% on three datasets Cora,



(a) Cora (b) PubMed (c) Amazon Photo

Fig. 6. Robustness analysis on the datasets Cora, PubMed, and Amazon Photo in terms of classification accuracy (in %).

Table 7. The Compared Running Time Cost of the Compared Methods (Seconds)

Methods	Cora	CiteSeer	PubMed	Amazon Computers	Amazon Photo	Coauthor CS
Chebyshev	7.9	8.2	11.1	17.6	10.4	17.5
GCN	6.6	7.2	10.7	19.9	11.2	21.4
GAT	8.7	8.8	6.8	16.1	8.3	14.7
SGC	3.8	3.9	3.9	3.7	3.6	7.5
DGI	16.8	19.1	59.6	56.3	31.7	90.2
GMI	90.5	85.5	520.8	624.8	396.5	812.6
MVGRL	287.1	296.9	489.9	554.0	472.2	578.7
AM-GCN	18.5	7.9	OOM	1055.0	237.6	OOM
GRACE	145.7	69.6	209.8	297.8	215.6	590.6
CG^3	1156.0	1036.1	1326.8	1702.4	1563.7	3512.4
HCPL (Ours)	52.4	59.9	70.4	125.5	75.8	189.2

OOM means out-of-memory.

PubMed, and Amazon Photo, respectively. It can be shown that our HCPL obtains the best results under various random attack perturbation rates. Moreover, our HCPL decreases less as the perturbation rate grows, demonstrating the robustness of our HCPL.

4.6 Efficiency Analysis

In this part, we analyze the efficiency of competing methods by comparing their running time. The compared results on six datasets are collected in Table 7. From the results, we can find that our method has better efficiency compared with various recent works (i.e., MVGCL, CG^3, and AMGCN). Actually, besides GNNs, these current works, i.e., MVGCL, CG^3, and AMGCN utilize additional complex techniques, which could bring in huge computational cost. MVGCL introduces different data augmentation strategies to generate multiple topological views for mutual information maximization. CG^3 needs to calculate data similarities and involves both local graph convolution and global hierarchical graph convolution. AMGCN extracts node embeddings from different views and then fuses them using the attention mechanism. These techniques bring more computational cost than our curriculum learning. Although some of the early methods have better efficiency, their performance is much worse than ours. Therefore, our HCPL exhibits competitive model scalability from the comparable running time.

4.7 Visualization

In this subsection, we demonstrate the t-SNE visualization [11] of the node embeddings generated by four methods on Cora, CiteSeer, and Amazon Photo. The compared results can be found in

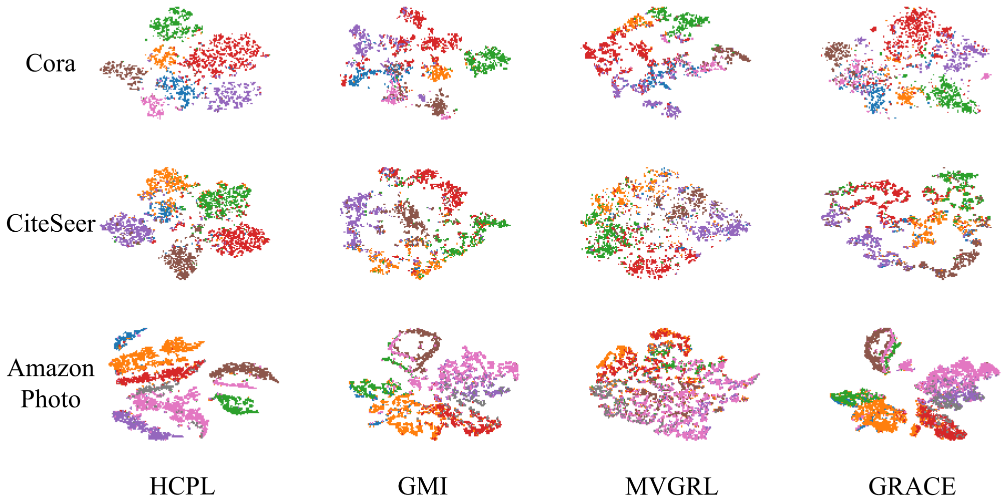


Fig. 7. t-SNE visualization on three datasets Cora, CiteSeer, and Amazon Photo.

Figure 7. From the results, the embeddings generated by our HCPL are more discriminative on these three datasets since these embeddings belonging to different categories can be better separated. This finding can result from our hybrid pseudo-labeling strategy, which provides extra high-quality supervision for the neural network, therefore validating our superiority again.

5 CONCLUSION

In this research, we investigate the problem of semi-supervised node classification on graphs and propose a simple yet effective model HCPL. Note that pseudo-labeling techniques are predominantly studied in the visual domains but have not been well applied to effectively solving node classification problems on graphs yet. In this article, our HCPL annotates unlabeled samples by training a classification model on the labeled nodes as well as pseudo-labeled nodes, and repeats the procedure with self-training. We propose a hybrid pseudo-label selection strategy for reliable guidance. Moreover, the concept of curriculum learning is introduced to progressively learn from simple pseudo-labels to hard pseudo-labels in terms of confidence and uncertainty. In this way, our HCPL can sufficiently explore the unlabeled data through our pseudo-labeling strategy. Extensive experiments on six well-known benchmarks validate the effectiveness of the proposed HCPL. In future work, we will attempt to introduce our pseudo-labeling strategy into other graph-related tasks such as link prediction and graph classification.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for critically reading the manuscript and for giving important suggestions to improve their paper.

REFERENCES

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. 2021. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *CVPR*.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

[3] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research* 7, 11 (2006), 2399–2434.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. 41–48.

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. MixMatch: A holistic approach to semi-supervised learning. In *NeurIPS*.

[6] Aleksandar Bojchevski and Stephan Günnemann. 2017. Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking. https://arxiv.org/abs/1707.03815

[7] L. Cai, J. Li, J. Wang, and S. Ji. 2022. Line graph neural networks for link prediction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2022), 5103–5113. DOI: 10.1109/TPAMI.2021.3080635

[8] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. 2021. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *AAAI*.

[9] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. 2021. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*.

[10] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*.

[11] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*. 647–655.

[12] Xiao Dong, En Yu, Min Gao, Lei Zhu, Jiande Sun, and Huaxiang Zhang. 2017. Semi-supervised distance consistent cross-modal retrieval. In *Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities*. 25–31.

[13] Fernando P. Dos Santos, Cemre Zor, Josef Kittler, and Moacir A. Ponti. 2020. Learning image features with fewer labels using a semi-supervised deep convolutional network. *Neural Networks* 132 (2020), 131–143.

[14] Sichao Fu, Weifeng Liu, Weili Guan, Yicong Zhou, Dapeng Tao, and Changsheng Xu. 2021. Dynamic graph learning convolutional networks for semi-supervised classification. *ACM Transactions on Multimedia Computing, Communications, and Applications* 17, 1s (2021), 1–13.

[15] Chen Gong, Dacheng Tao, Wei Liu, Liu Liu, and Jie Yang. 2016. Label propagation via teaching-to-learn and learning-to-teach. *IEEE Transactions on Neural Networks and Learning Systems* 28, 6 (2016), 1452–1465.

[16] Chen Gong, Dacheng Tao, Wei Liu, Stephen J. Maybank, Meng Fang, Keren Fu, and Jie Yang. 2015. Saliency propagation from simple to difficult. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2531–2539.

[17] Chen Gong, Dacheng Tao, Stephen J. Maybank, Wei Liu, Guoliang Kang, and Jie Yang. 2016. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing* 25, 7 (2016), 3249–3260.

[18] Chen Gong, Dacheng Tao, Jie Yang, and Wei Liu. 2016. Teaching-to-learn and learning-to-teach for multi-label propagation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[19] Chen Gong, Jian Yang, and Dacheng Tao. 2019. Multi-modal curriculum learning over graphs. *ACM Transactions on Intelligent Systems and Technology* 10, 4 (2019), 1–25.

[20] Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised learning by entropy minimization. In *NeurIPS*.

[21] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

[22] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*.

[23] Feihu Huang, Peiyu Yi, Jince Wang, Mengshi Li, Jian Peng, and Xi Xiong. 2022. A dynamical spatial-temporal graph neural network for traffic demand prediction. *Information Sciences* 594 (2022), 286–304.

[24] Yiqiao Jin, Xiting Wang, Ruichao Yang, Yizhou Sun, Wei Wang, Hao Liao, and Xing Xie. 2022. Towards fine-grained reasoning for fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 5746–5754.

[25] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, Junwei Yang, Jingyang Yuan, Yusheng Zhao, Xiao Luo, and Ming Zhang. 2023. A comprehensive survey on deep graph representation learning. https://arxiv.org/abs/2304.05055

[26] Wei Ju, Yiyang Gu, Binqi Chen, Gongbo Sun, Yifang Qin, Xingyuming Liu, Xiao Luo, and Ming Zhang. 2023. GLCC: A general framework for graph-level clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4391–4399.

[27] Wei Ju, Xiao Luo, Meng Qu, Yifan Wang, Chong Chen, Minghua Deng, Xian-Sheng Hua, and Ming Zhang. 2023. TGNN: A joint semi-supervised framework for graph-level classification. https://arxiv.org/abs/2304.11688

[28] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

[29] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[30] Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*.

[31] Haifeng Li, Jun Cao, Jiawei Zhu, Yu Liu, Qing Zhu, and Guohua Wu. 2022. Curvature graph neural network. *Information Sciences* 592 (2022), 50–66.

[32] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. 2022. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning*. 13242–13256.

[33] Meng Liu, Hongyang Gao, and Shuiwang Ji. 2020. Towards deeper graph neural networks. In *KDD*.

[34] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2021. Is heterophily a real nightmare for graph neural networks to do node classification? https://arxiv.org/abs/2109.05641

[35] Xiao Luo, Yusheng Zhao, Yifang Qin, Wei Ju, and Ming Zhang. 2023. Towards semi-supervised universal graph classification. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[36] Qiaozhu Mei, Duo Zhang, and ChengXiang Zhai. 2008. A general optimization framework for smoothing language models on graph structures. In *SIGIR*.

[37] Hao Peng, Hongfei Wang, Bowen Du, Md Zakirul Alam Bhuiyan, Hongyuan Ma, Jianwei Liu, Lihong Wang, Zeyu Yang, Linfeng Du, Senzhang Wang, and Philip S. Yu. 2020. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences* 521 (2020), 277–290.

[38] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*.

[39] Shengsheng Qian, Dizhan Xue, Huaiwen Zhang, Quan Fang, and Changsheng Xu. 2021. Dual adversarial graph neural networks for multi-label cross-modal retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 2440–2448.

[40] Yifang Qin, Hongjun Wu, Wei Ju, Xiao Luo, and Ming Zhang. 2023. A diffusion model for POI recommendation. *ACM Transactions on Information Systems* (2023). Early access.

[41] Yuanyuan Qing, Yijie Zeng, and Guang-Bin Huang. 2021. Label propagation via local geometry preserving for deep semi-supervised image recognition. *Neural Networks* 143 (2021), 303–313.

[42] Hongyan Ran, Caiyan Jia, Pengfei Zhang, and Xuanya Li. 2022. MGAT-ESM: Multi-channel graph attention neural network with event-sharing module for rumor detection. *Information Sciences* 592 (2022), 402–416.

[43] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. https://arxiv.org/abs/2101.06329

[44] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine* 29, 3 (2008), 93–93.

[45] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. https://arxiv.org/abs/1811.05868

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

[47] Feng Shi, Ahren Yiqiao Jin, and Song-Chun Zhu. 2021. VersaGNN: A versatile accelerator for graph neural networks. https://arxiv.org/abs/2105.01280

[48] Zhixin Shi, Frederick Kiefer, John Schneider, and Venu Govindaraju. 2008. Modeling biometric systems using the general Pareto distribution (GPD). In *Biometric Technology for Human Identification V*, Vol. 6944. 69440O.

[49] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A. Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*.

[50] Enmei Tu, Zihao Wang, Jie Yang, and Nikola Kasabov. 2022. Deep semi-supervised learning via dynamic anchor graph embedding in latent space. *Neural Networks* 146 (2022), 350–360.

[51] Jesper E. Van Engelen and Holger H. Hoos. 2020. A survey on semi-supervised learning. *Machine Learning* 109, 2 (2020), 373–440.

[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

[53] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep graph infomax. In *ICLR*.

[54] Sheng Wan, Shirui Pan, Jian Yang, and Chen Gong. 2021. Contrastive and generative graph convolutional networks for graph-based semi-supervised learning. In *AAAI*.

[55] Fei Wang, Lei Zhu, Liang Xie, Zheng Zhang, and Mingyang Zhong. 2021. Label propagation with structured graph learning for semi-supervised dimension reduction. *Knowledge-Based Systems* 225 (2021), 107130.

[56] Jie Wang, Jianqing Liang, Junbiao Cui, and Jiye Liang. 2021. Semi-supervised learning with mixed-order graph convolutional networks. *Information Sciences* 573 (2021), 171–181.

[57] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. 2020. AM-GCN: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1243–1253.

[58] Yifan Wang, Yiping Song, Shuai Li, Chaoran Cheng, Wei Ju, Ming Zhang, and Sheng Wang. 2022. DisenCite: Graph-based disentangled representation learning for context-specific citation generation. In *AAAI.* (2022).

[59] Fei Wu, Xiao-Yuan Jing, Pengfei Wei, Chao Lan, Yimu Ji, Guo-Ping Jiang, and Qinghua Huang. 2022. Semi-supervised multi-view graph convolutional networks with application to webpage classification. *Information Sciences* 591, (2022), 142–154.

[60] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *ICML.*

[61] Man Wu, Shirui Pan, Lan Du, and Xingquan Zhu. 2021. Learning graph neural networks with positive and unlabeled nodes. *ACM Transactions on Knowledge Discovery from Data* 15, 6 (2021), 1–25.

[62] Man Wu, Shirui Pan, and Xingquan Zhu. 2020. OpenWGL: Open-world graph learning. In *IEEE International Conference on Data Mining.* 681–690.

[63] Man Wu, Shirui Pan, and Xingquan Zhu. 2022. Attraction and repulsion: Unsupervised domain adaptive graph contrastive learning network. *IEEE Transactions on Emerging Topics in Computational Intelligence* 6, 5 (2022), 1079–1091.

[64] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.

[65] Ze Xiao and Yue Deng. 2020. Graph embedding-based novel protein interaction prediction via higher-order graph convolutional network. *PloS One* 15, 9 (2020), e0238915.

[66] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks?. In *ICLR.*

[67] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S. Yu. 2021. ConsisRec: Enhancing GNN for social recommendation via consistent neighbor aggregation. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2141–2145.

[68] Ruichao Yang, Xiting Wang, Yiqiao Jin, Chaozhuo Li, Jianxun Lian, and Xing Xie. 2022. Reinforcement subgraph reasoning for fake news detection. In *The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 2253–2262.

[69] Kaixuan Yao, Jiye Liang, Jianqing Liang, Ming Li, and Feilong Cao. 2022. Multi-view graph convolutional networks with attention mechanism. *Artificial Intelligence* 307 (2022), 103708.

[70] Si-Yu Yi, Wei Ju, Yifang Qin, Xiao Luo, Luchen Liu, Yong-Dao Zhou, and Ming Zhang. 2023. Redundancy-free self-supervised relational learning for graph clustering. *IEEE Transactions on Neural Networks and Learning Systems* (2023). Early access.

[71] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. (n.d.). In *34th Conference on Neural Information Processing Systems (NeurIPS'20).*

[72] En Yu, Jiande Sun, Jing Li, Xiaojun Chang, Xian-Hua Han, and Alexander G. Hauptmann. 2018. Adaptive semi-supervised feature selection for cross-modal retrieval. *IEEE Transactions on Multimedia* 21, 5 (2018), 1276–1288.

[73] Jiarui Zhang, Jian Huang, Jialong Gao, Runhai Han, and Cong Zhou. 2022. Knowledge graph embedding by logical-default attention graph convolution neural network for link prediction. *Information Sciences* 593 (2022), 201–215.

[74] Yaobin Zhang, Weihong Deng, Mei Wang, Jiani Hu, Xian Li, Dongyue Zhao, and Dongchao Wen. 2020. Global-local GCN: Large-scale label noise cleansing for face recognition. In *CVPR.*

[75] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. 2021. Backdoor attacks to graph neural networks. In *The 26th ACM Symposium on Access Control Models and Technologies.* 15–26.

[76] Mingbo Zhao, Tommy W. S. Chow, Zhao Zhang, and Bing Li. 2015. Automatic image annotation via compact graph based semi-supervised learning. *Knowledge-Based Systems* 76 (2015), 148–165.

[77] Dengyong Zhou, Olivier Bousquet, Thomas N. Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *NeurIPS.*

[78] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Deep graph contrastive representation learning. In *ICLR Workshop.*

[79] Haodong Zou, Zhen Duan, Xinru Guo, Shu Zhao, Jie Chen, Yanping Zhang, and Jie Tang. 2021. On embedding sequence correlations in attributed network for semi-supervised node classification. *Information Sciences* 562 (2021), 385–397.

Received 27 March 2023; revised 29 August 2023; accepted 27 September 2023