Posterior Label Smoothing for Node Classification

Jaeseung Heo[1], **MoonJeong Park**[1], **Dongwoo Kim**[1,2]
[1]Graduate School of Artificial Intelligence
[2]Department of Computer Science & Engineering
POSTECH, South Korea
{jsheo12304,mjeongp,dongwookim}@postech.ac.kr

Abstract

Label smoothing is a widely studied regularization technique in machine learning. However, its potential for node classification in graph-structured data, spanning homophilic to heterophilic graphs, remains largely unexplored. We introduce *posterior label smoothing*, a novel method for transductive node classification that derives soft labels from a posterior distribution conditioned on neighborhood labels. The likelihood and prior distributions are estimated from the global statistics of the graph structure, allowing our approach to adapt naturally to various graph properties. We evaluate our method on 10 benchmark datasets using eight baseline models, demonstrating consistent improvements in classification accuracy. Our code is available at https://anonymous.4open.science/r/PosteL.

1 Introduction

Soft label, which contains class-wise probabilities, has demonstrated remarkable success in training neural networks across various domains, including computer vision and natural language processing [16, 18, 28, 32, 33, 38]. One of the popular approaches to obtain a soft label is label smoothing, which introduces uniform noise into the ground-truth labels. Despite its simplicity, this technique effectively regularizes the output distribution and enhances generalization [21]. Knowledge distillation [8] is another effective option, which trains a teacher model with a given one-hot label and utilizes its output as a soft-label to train the student model.

One of the most convincing explanations of why knowledge distillation works is that soft label enables the learning of *"dark knowledge"* included in instances [1, 8]. Since additional information captured by the teacher model that one-hot labels cannot convey is encoded as a soft label, the student model learns richer features.

Considering the graph dataset, the relation between nodes that the graph already contains can be helpful for node classification. As the quote says, *"You can tell a person by the company they keep,"* our idea is to encode the neighbor's information into the soft label. More specifically, we utilize the posterior distribution, i.e., the probability of the node label given its neighbor nodes' labels. This principle naturally generalizes both homophilic and heterophilic settings: in homophilic graphs, a target node is likely to have the same label as its neighbors, whereas in heterophilic graphs, the target node is likely to have different labels, which is supported by our theoretical analysis.

Existing approaches that generate soft labels in the graph domain build up the method based on a more specified assumption that nodes tend to share the same label with their neighboring nodes. Based on this assumption, they construct soft labels by naively aggregating the labels of neighboring nodes [35, 41]. This approach aligns well with homophilic graphs, where nodes of the same class are likely to be connected, leading to improved generalization. However, it conflicts with the nature of heterophilic graphs, where edges frequently connect nodes with different labels.

Workshop on New Perspectives in Advancing Graph Machine Learning at NeurIPS 2025.

Based on this intuition, we propose **Poste**rior **L**abel Smoothing (PosteL), a novel method that derives the soft label as the posterior distribution. The likelihood is approximated by the product of conditional label distributions over the node's neighborhood. To estimate the prior and conditional distributions, we count label occurrences at nodes and label co-occurrences across edges, thereby constructing global statistics that capture the label dependencies encoded in the graph structure. The resulting soft label, therefore, encapsulates rich information from both the local neighborhood structure and the global label distribution.

Since PosteL needs the information of label co-occurrences and global statistics of the graph, accurate information would be important for the success of our approach. However, we can only access the labels of train nodes, while the labels of test nodes remain unknown. The lack of information can result in weakening the efficacy of our method. To address this issue, we propose an iterative pseudo-labeling procedure that utilizes pseudo-labels to re-obtain soft labels. Specifically, neighbor nodes' information is updated and recalculated by the prior and likelihood, which are also re-estimated by pseudo labels.

We apply our smoothing method to eight baseline neural network models, including a multi-layer perceptron and variants of graph neural networks, and test their performances on 10 graph benchmarks, including five homophilic and five heterophilic graphs. Across 80 model–dataset combinations, the soft label approach with iterative pseudo-labeling improves classification accuracy in 76 cases.

2 Related work

2.1 Node classification

Various works leverage graph structures in different ways to perform node classification. Early approaches such as GCN [13], GraphSAGE [6], and GAT [34] aggregate neighbor representations under the homophilic assumption. For tackling class imbalance on homophilic graphs, GraphSMOTE [40], ImGAGN [22], and GraphENS [19] have been proposed. Meanwhile, H_2GCN [42] and U-GCN [12] enhance performance on heterophilic graphs by aggregating representations from multi-hop neighbors. Other research focuses on adaptively learning the graph structure itself. For instance, GPR-GNN [2] and CPGNN [43] determine which nodes to aggregate, while ChebNet [3], APPNP [4], and BernNet [7] focus on learning appropriate filters from graph signals.

2.2 Classification with soft labels

Hinton et al. [8] demonstrate that training a small student model with soft labels derived from a large teacher model's predictions outperforms training with one-hot labels. This approach, known as knowledge distillation (KD), has proven effective for both model compression and performance improvement [11, 14, 29].

Alternatively, simpler methods for generating soft labels exist. Label smoothing [28] adds uniform noise to one-hot labels, and its benefits have been widely explored. For instance, Müller et al. [18] show that the label smoothing improves model calibration, while Lukasik et al. [15] connect the label smoothing to label-correction techniques and demonstrate its utility in addressing label noise. The label smoothing is popular in computer vision [16, 32, 38] and NLP [5, 26, 33], yet it remains relatively underexplored in the graph domain.

To our knowledge, only two studies specifically propose label smoothing techniques for node classification. SALS [35] smooths a node's label to match those of its neighbors, and ALS [41] aggregates neighborhood labels with adaptive refinements. However, neither work focuses on heterophilic graphs, where nodes often connect to dissimilar neighbors. Meanwhile, other studies have proposed smoothing the prediction output based on the graph structure [36, 39], but their motivations differ substantially from the label smoothing approach investigated in this paper (e.g., they adjust output logits rather than training labels).

3 Method

In this section, we present our label smoothing approach for node classification and propose a new training strategy that iteratively refines soft labels via pseudo-labels obtained after training.

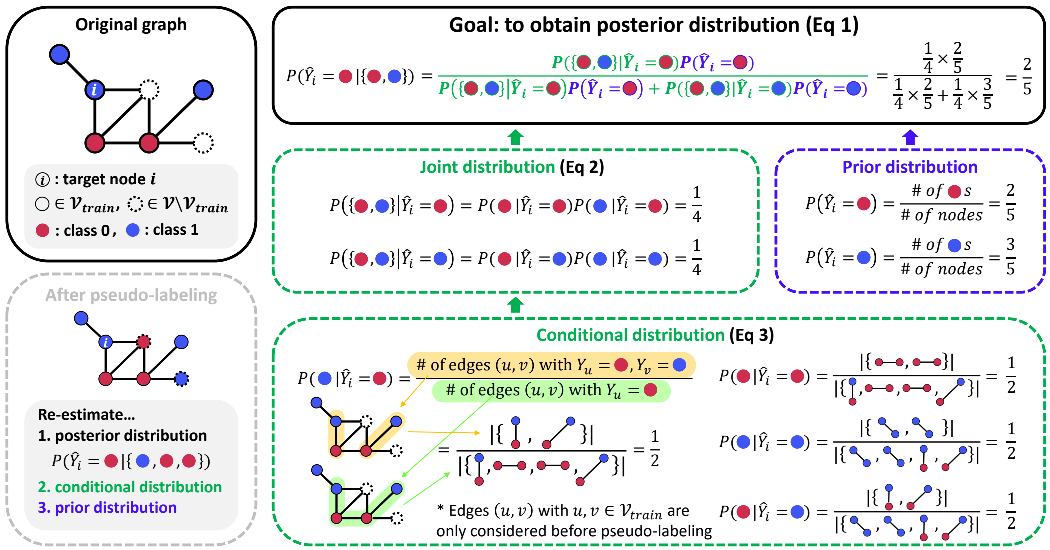


Figure 1: Overall illustration of posterior label smoothing. To relabel the node label, we compute the posterior distribution of the label given neighborhood labels. The likelihood and prior distributions are estimated from global statistics. The statistics are updated through the pseudo-labels after training, resulting in an iterative algorithm.

3.1 Posterior label smoothing

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ be a graph, where $\mathcal{V}$ is a set of nodes, and $\mathcal{E}$ is a set of edges, and $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the d-dimensional node feature matrix. We consider a transductive node classification scenario in which we observe the graph structure for all nodes, including test nodes, but only the labels of nodes in the training set. For each node i in a training set, we have a label $y_i \in [K]$, where K is the total number of classes. Let $\boldsymbol{e}_i \in \{0, 1\}^K$ be a one-hot encoding of y_i, i.e., $e_{ik} = 1$ if $y_i = k$ and $\sum_k e_{ik} = 1$.

We propose an effective relabeling method to allocate a new soft label to each node based on the *local neighborhood structure* and *global label statistics*. Let $\hat{Y}_i$ be a random variable representing the soft label of node i. Given the one-hop neighborhood $\mathcal{N}(i) = \{j \mid (i, j) \in \mathcal{E}\}$, we compute the posterior distribution of $\hat{Y}_i$ conditioned on the labels of its neighbors using Bayes' rule:

$$P(\hat{Y}_i = k \mid \{Y_j = y_j\}_{j \in \mathcal{N}(i)}) = \frac{P(\{Y_j = y_j\}_{j \in \mathcal{N}(i)} | \hat{Y}_i = k) P(\hat{Y}_i = k)}{\sum_{\ell=1}^{K} P(\{Y_j = y_j\}_{j \in \mathcal{N}(i)} | \hat{Y}_i = \ell) P(\hat{Y}_i = \ell)} . \tag{1}$$

To obtain the likelihood $P(\{Y_j = y_j\}_{j \in \mathcal{N}(i)} | \hat{Y}_i = k)$, we assume that the labels of the neighboring nodes are conditionally independent given $\hat{Y}_i$, i.e.,

$$P(\{Y_j\}_{j \in \mathcal{N}(i)} \mid \hat{Y}_i) = \prod_{j \in \mathcal{N}(i)} P(Y_j \mid \hat{Y}_i) . \tag{2}$$

We empirically verify the conditional independence assumption in appendix G.

There are multiple ways to model the individual conditionals in the factorized form of Equation (2). In this work, we use the global statistics between adjacent nodes to estimate the conditional. Specifically, we define

$$P(Y_j = m | \hat{Y}_i = n) := \frac{|\{(u, v) \mid y_v = m, y_u = n, (u, v) \in \mathcal{E}\}|}{|\{(u, v) \mid y_u = n, (u, v) \in \mathcal{E}\}|} . \tag{3}$$

We also estimate the prior distribution from global label frequencies. Concretely, we set $P(\hat{Y}_i = m) := |\{u \mid y_u = m\}| / |\mathcal{V}|$. In Appendix G, we investigate alternative designs for the likelihood and compare their performances. Figure 1 presents an example of obtaining the posterior distribution on a toy graph.

The posterior distribution serves as a soft label for model training. However, to prevent the posterior from becoming overly confident, we incorporate a small amount of uniform noise, ϵ. Additionally, because the most probable label from the posterior may not always align with the ground-truth, e.g., due to label noise or limited local information, we interpolate the posterior with the one-hot label. To this end, we obtain the target label $\hat{\boldsymbol{e}}_i$ used for actual training as

$$\hat{\boldsymbol{e}}_i = \alpha \tilde{\boldsymbol{e}}_i + (1-\alpha)\boldsymbol{e}_i \ , \tag{4}$$

where $\tilde{e}_{ik} \propto P(\hat{Y}_i = k \mid \{Y_j = y_j\}_{j \in \mathcal{N}(i)}) + \beta\epsilon$, and α and β are hyperparameters controlling the weights of interpolation and uniform noise. By enforcing $\alpha < 1/2$, we can keep the most probable label of the target label the same as the ground-truth label, but we find that this condition is not necessary for empirical experiments. We refer to our method as PosteL (**Poste**rior **L**abel smoothing). The detailed algorithm of PosteL is shown in Algorithm 1 in Appendix A.

3.2 Iterative pseudo-labeling

Posterior relabeling derives a node's soft label by leveraging the labels of its neighbors. However, its effectiveness can be limited by certain graph properties, particularly sparsity and label noise. For instance, if a node has no labeled neighbors, the likelihood term becomes uniform, making the posterior depend solely on the prior; if only a few neighbors are labeled and those labels are noisy, the posterior can become skewed. These challenges are more pronounced in sparse graphs: in the Cornell dataset, for example, 26.35% of nodes have no labeled neighbors, making posterior relabeling especially difficult.

To address these limitations, we propose updating the likelihoods and priors using pseudo-labels generated for the validation and test nodes. Specifically, we first train a graph neural network with the target labels obtained from Equation (4) and then use its predictions on the validation and test nodes to obtain pseudo-labels. We assign each unlabeled node the most probable class from the model's output. Next, we update the likelihood and prior based on these pseudo-labels, while retaining the ground-truth labels for training nodes, to recalibrate both the posterior smoothing and the resulting soft labels.

We repeat this cycle of training and re-calibration until we achieve the best validation loss, aiming to maximize node classification performance. Intuitively, if posterior label smoothing improves predictive accuracy through better likelihood and prior estimation, then the resulting pseudo-labels should, in turn, further refine these distributions, provided that the pseudo-labels contain minimal errors. The detailed algorithms for the training process involving iterative pseudo-labeling are presented in Algorithm 2 in Appendix A, while the complexity analysis is provided in Appendix F.

4 Theoretical analysis of PosteL

We analyze how PosteL behaves under different graph homophily and heterophily conditions in a binary classification setting. Specifically, we demonstrate how PosteL (i) adapts label assignments based on the neighborhood label distribution and (ii) remains robust across both homophilic and heterophilic graphs. While our focus here is on binary classification for clarity, a similar argument extends to multi-class scenarios as well.

Recall from Equation (3) that PosteL captures the adjacency relationship via empirical edge statistics. In the binary setting, let $\mathcal{N}_k(i)$ denote the set of neighbors of node i with label $k \in \{0, 1\}$. Further, we define the *class homophily* c_k for each label k as

$$c_k := \frac{|\{(i,j) \mid (i,j) \in \mathcal{E}, y_i = k, y_j = k\}|}{|\{(i,j) \mid (i,j) \in \mathcal{E}, y_i = k\}|}, \tag{5}$$

which measures how likely two adjacent nodes are both labeled k among all edges that include a node labeled k. Thus, $c_k > 0.5$ indicates that nodes labeled k tend to be adjacent to others with label k, i.e., *homophilic*, whereas $c_k < 0.5$ indicates they tend to connect to nodes with the opposite label, i.e., *heterophilic*.

The following lemma states the condition under which the posterior of label k is higher than $1-k$.

Lemma 1 (Homophilic graph)**.** *Suppose that the classes are balanced, i.e.,* $P(\hat{Y} = 0) = P(\hat{Y} = 1)$ *and the graph is homophilic, i.e.,* $c_k > 1 - c_{1-k}$*. Then, for any node* i *with neighbors* $\mathcal{N}(i)$*, the* *posterior probability satisfies,*

$$P(\hat{Y}_i = k | \{Y_j = y_j\}_{j \in \mathcal{N}(i)}) > 0.5$$

if and only if

$$|\mathcal{N}_k(i)| > |\mathcal{N}_{1-k}(i)| \cdot \frac{\log c_{1-k} - \log(1 - c_k)}{\log c_k - \log(1 - c_{1-k})}.$$

Intuitively, Lemma 1 states that if the graph is sufficiently homophilic, having more neighbors with label k than with label $1 - k$ pushes the posterior probability for k above 0.5.

A similar statement holds for heterophilic graphs.

Lemma 2 (Heterophilic graph)**.** *Under the same assumptions used in Lemma 1, but now with a heterophilic condition, i.e.,* $c_k < 1 - c_{1-k}$*, we have,*

$$P(\hat{Y}_i = k | \{Y_j = y_j\}_{j \in \mathcal{N}(i)}) > 0.5$$

if and only if

$$|\mathcal{N}_k(i)| < |\mathcal{N}_{1-k}(i)| \cdot \frac{\log c_{1-k} - \log(1 - c_k)}{\log c_k - \log(1 - c_{1-k})}.$$

Lemma 2 indicates that in a heterophilic graph, having fewer neighbors of label k than of label $1 - k$ can make the posterior favor k. Detailed proofs are provided in Appendix B. These two lemmas highlight the key difference between PosteL and the neighborhood aggregation method in Wang et al. [35]. In their method, naively aggregating neighborhood labels for smoothing on heterophilic graphs results in the soft label being dominated by the majority neighborhood label. This *majority dominance* contradicts the inherent heterophilic property, where nodes are more likely to connect to dissimilar labels. In contrast, Lemma 2 demonstrates that PosteL assigns a lower probability to the majority neighborhood label in heterophilic graphs, thereby mitigating majority dominance, while Lemma 1 shows that PosteL effectively retains similarity to the majority neighbor label in homophilic graphs. Moreover, nodes in heterophilic graphs tend to connect with nodes that have different labels, which implies that $|\mathcal{N}_{y_i}(i)| < |\mathcal{N}_{1-y_i}(i)|$. In this case, PosteL preserves the ground-truth label as the most probable label. A similar effect is also found in homophilic graphs.

5 Experiments

In this section, we evaluate our proposed method against various label smoothing methods on 10 node classification tasks, utilizing eight different backbone GNNs. A comprehensive analysis highlighting the importance of each design choice is provided in Appendix G.

5.1 Node classification

In this section, we evaluate the improvements in node classification performance achieved by our method across a range of datasets and backbone models. We aim to demonstrate the robustness and consistent effectiveness of our approach across graphs with varying structural and label characteristics.

Datasets We assess the performance of our method across 10 node classification datasets. To examine the effect of our method on diverse types of graphs, we conduct experiments on both homophilic and heterophilic graphs. For the homophilic setting, we evaluate our method on five datasets: Cora, CiteSeer, and PubMed, which are citation networks where nodes represent documents and edges correspond to citation links [25, 37], as well as the Amazon co-purchase graphs Computers and Photo [17], where nodes represent products and edges indicate frequent co-purchases. For the heterophilic setting, we use five datasets: Chameleon and Squirrel, which are Wikipedia networks where nodes represent pages and edges correspond to mutual links [24]; Actor, a co-occurrence network where nodes represent actors and edges indicate co-appearances on the same Wikipedia pages [30]; and Texas and Cornell, which are webpage graphs where nodes represent web pages and edges denote hyperlinks [20]. Detailed statistics of each dataset are illustrated in Appendix C.

Table 1: Classification accuracy on 10 node classification datasets. Δ represents the performance improvement achieved by PosteL compared to the backbone model trained with the ground-truth label. All results of the backbone model trained with the ground-truth label are sourced from He et al. [7].

	Homophilic					Heterophilic				
	Cora	CiteSeer	PubMed	Computers	Photo	Chameleon	Actor	Squirrel	Texas	Cornell
GCN	87.14±1.01	79.86±0.67	86.74±0.27	83.32±0.33	88.26±0.73	59.61±2.21	33.23±1.16	46.78±0.87	77.38±3.28	65.90±4.43
+LS	87.77±0.97	81.06±0.59	87.73±0.24	89.08±0.30	94.05±0.26	64.81±1.53	33.81±0.75	49.53±1.10	77.87±3.11	67.87±3.77
+KD	87.90±0.90	80.97±0.56	87.03±0.29	88.56±0.36	93.64±0.31	64.49±1.38	33.33±0.78	49.38±0.64	78.03±2.62	63.61±5.57
+SALS	88.10±1.08	80.52±0.85	87.23±0.13	88.88±0.54	93.80±0.31	63.00±1.75	33.24±0.92	49.16±0.77	70.00±3.93	58.36±7.54
+ALS	88.10±0.85	81.02±0.52	87.30±0.30	89.18±0.36	93.88±0.27	64.11±1.29	34.05±0.49	47.44±0.76	77.38±2.13	71.64±3.28
+PosteL	**88.56±0.90**	**82.10±0.50**	**88.00±0.25**	**89.30±0.23**	**94.08±0.35**	**65.80±1.23**	**35.16±0.43**	**52.76±0.64**	**80.82±2.79**	**80.33±1.80**
Δ	+1.42(↑)	+2.24(↑)	+1.26(↑)	+5.98(↑)	+5.82(↑)	+6.19(↑)	+1.93(↑)	+5.98(↑)	+3.44(↑)	+14.43(↑)
GAT	88.03±0.79	80.52±0.71	87.04±0.24	83.32±0.39	90.94±0.68	63.13±1.93	33.93±2.47	44.49±0.88	**80.82±2.13**	78.21±2.95
+LS	88.69±0.99	81.27±0.86	86.33±0.32	88.95±0.31	94.06±0.39	65.16±1.49	34.55±1.15	45.94±1.60	78.69±4.10	74.10±4.10
+KD	87.47±0.94	80.79±0.60	86.54±0.31	88.99±0.46	93.76±0.31	65.14±1.47	35.13±1.36	43.86±0.85	79.02±2.46	73.44±2.46
+SALS	88.64±0.94	81.23±0.59	86.49±0.25	88.75±0.36	93.74±0.37	62.76±1.42	33.91±1.41	42.29±0.94	74.92±4.43	65.57±10.00
+ALS	88.60±0.92	81.09±0.68	87.06±0.24	89.57±0.35	94.16±0.36	66.15±1.25	34.05±0.52	46.85±1.45	78.03±3.11	75.08±3.77
+PosteL	**89.21±1.08**	**82.13±0.64**	**87.08±0.19**	**89.60±0.29**	**94.31±0.31**	**66.28±1.14**	**35.92±0.72**	**49.38±1.05**	80.33±2.62	**80.33±1.81**
Δ	+1.18(↑)	+1.61(↑)	+0.04(↑)	+6.28(↑)	+3.37(↑)	+3.15(↑)	+1.99(↑)	+4.89(↑)	−0.49(↓)	+2.12(↑)
BernNet	88.52±0.95	80.09±0.79	88.48±0.41	87.64±0.44	93.63±0.35	68.29±1.58	**41.79±1.01**	51.35±0.73	93.12±0.65	92.13±1.64
+LS	88.80±0.92	80.37±1.05	87.40±0.27	88.32±0.38	93.70±0.21	69.58±0.94	39.60±0.53	52.39±0.60	91.80±1.80	90.49±1.48
+KD	87.78±0.99	81.20±0.86	87.59±0.41	87.35±0.40	93.96±0.40	67.75±1.42	41.04±0.89	51.25±0.83	93.61±1.31	90.33±2.30
+SALS	88.77±0.85	81.20±0.61	88.61±0.35	88.87±0.33	94.22±0.43	64.62±0.85	40.15±1.07	46.19±0.78	85.90±4.10	88.03±3.12
+ALS	89.13±0.79	81.17±0.67	**89.19±0.46**	89.52±0.30	**94.54±0.32**	67.92±1.07	40.51±0.61	51.83±1.31	93.77±1.31	92.79±1.48
+PosteL	**89.39±0.92**	**82.46±0.67**	89.07±0.29	**89.56±0.35**	**94.54±0.36**	**69.65±0.83**	40.40±0.67	**53.11±0.87**	**93.93±1.15**	**92.95±1.80**
Δ	+0.87(↑)	+2.37(↑)	+0.59(↑)	+1.92(↑)	+0.91(↑)	+1.36(↑)	−1.39(↓)	+1.76(↑)	+0.81(↑)	+0.82(↑)

Experimental setup and baselines We evaluate the performance of PosteL across various backbone models, including a multi-layer perception (MLP) without graph structure, and seven widely used graph neural networks: GCN [13], GAT [34], APPNP [4], ChebNet [3], GPR-GNN [2], BernNet [7], and OrderedGNN [27].

We follow the experimental setup and backbone implementations of He et al. [7]. Specifically, we use fixed 10 sets of train, validation, and test splits with ratios of 60%/20%/20%, respectively, and measure the accuracy at the lowest validation loss. Each model is trained for 1,000 epochs, with early stopping applied if the validation loss does not improve over the last 200 epochs. Details of the experimental setup, including the hyperparameter search spaces and additional implementation specifics, are provided in Appendix D.

We compare our method with two domain-agnostic soft labeling methods, including label smoothing (LS) [28] and knowledge distillation (KD) [8], as well as two label smoothing methods designed for node classification: SALS [35] and ALS [41].

Results Table 1 reports the classification accuracy and 95% confidence intervals for each of the three models across ten datasets. Complete results, including the performance of additional GNNs, are presented in Table 3 of Appendix E. Our method outperforms baseline methods in 76 out of 80 experimental settings. In 41 of these cases, the performance improvements exceed the 95% confidence interval, highlighting the robustness of our approach. On the Cornell dataset, using the GCN backbone, PosteL achieves a substantial improvement of 14.43%.

Compared to other soft labeling methods, PosteL consistently achieves superior performance. In particular, our method outperforms SALS and ALS, which are label smoothing methods specifically tailored for node classification, on both homophilic and heterophilic datasets. The improvements are especially significant on heterophilic datasets, indicating that the heterophily-aware label assignment strategy of PosteL effectively enhances classification performance in heterophilic graph settings.

6 Conclusion

We introduce a novel posterior label smoothing method for node classification on graphs. By combining local neighborhoods with global label statistics, PosteL improves model generalization. Extensive experiments on multiple datasets and models confirm its effectiveness, demonstrating significant performance gains over baseline methods.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (RS-2024-00337955; RS-2023-00217286) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2024-00457882, National AI Research Lab Project; RS-2019-II191906, Artificial Intelligence Graduate School Program(POSTECH)).

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[2] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020. 2.1, 5.1, D.2

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016. 2.1, 5.1, D.2

[4] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018. 2.1, 5.1, D.2

[5] Biyang Guo, Songqiao Han, Xiao Han, Hailiang Huang, and Ting Lu. Label confusion learning to enhance text classification models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12929–12936, 2021. 2.2

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2.1

[7] Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34: 14239–14251, 2021. 2.1, 1, 5.1, D.1, D.2, 3

[8] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 1, 2.2, 5.1

[9] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019. G

[10] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020. G

[11] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174. Association for Computational Linguistics, nov 2020. 2.2

[12] Di Jin, Zhizhi Yu, Cuiying Huo, Rui Wang, Xiao Wang, Dongxiao He, and Jiawei Han. Universal graph convolutional networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2.1

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2.1, 5.1, D.2

[14] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2599–2608, 2019. 2.2

[15] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020. 2.2

[16] Tohar Lukov, Na Zhao, Gim Hee Lee, and Ser-Nam Lim. Teaching with soft label smoothing for mitigating noisy labels in facial expressions. In *European Conference on Computer Vision*, pages 648–665. Springer, 2022. 1, 2.2

[17] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015. 5.1

[18] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 1, 2.2, G

[19] Joonhyung Park, Jaeyun Song, and Eunho Yang. GraphENS: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International Conference on Learning Representations*, 2022. 2.1

[20] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020. 5.1

[21] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017. 1

[22] Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1390–1398, 2021. 2.1

[23] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020. G

[24] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021. 5.1

[25] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008. 5.1

[26] Minguang Song, Yunxin Zhao, Shaojun Wang, and Mei Han. Learning recurrent neural network language models with context-sensitive label smoothing for automatic speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6159–6163. IEEE, 2020. 2.2

[27] Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. Ordered gnn: Ordering message passing to deal with heterophily and over-smoothing. *arXiv preprint arXiv:2302.01524*, 2023. 5.1

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2.2, 5.1

[29] Jiaxi Tang and Ke Wang. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2289–2298, New York, NY, USA, 2018. Association for Computing Machinery. 2.2

[30] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009. 5.1

[31] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. G

[32] Sukesh Adiga Vasudeva, Jose Dolz, and Herve Lombaert. Geols: Geodesic label smoothing for image segmentation. In *Medical Imaging with Deep Learning*, pages 468–478. PMLR, 2024. 1, 2.2

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 2.2

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2.1, 5.1, D.2

[35] Yiwei Wang, Yujun Cai, Yuxuan Liang, Wei Wang, Henghui Ding, Muhao Chen, Jing Tang, and Bryan Hooi. Structure-aware label smoothing for graph neural networks. *arXiv preprint arXiv:2112.00499*, 2021. 1, 2.2, 4, 5.1

[36] Tian Xie, Rajgopal Kannan, and C-C Jay Kuo. Label efficient regularization and propagation for graph node classification. *IEEE transactions on pattern analysis and machine intelligence*, 2023. 2.2

[37] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016. 5.1

[38] Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. Delving deep into label smoothing. *IEEE Transactions on Image Processing*, 30: 5984–5996, 2021. 1, 2.2

[39] Wentao Zhang, Mingyu Yang, Zeang Sheng, Yang Li, Wen Ouyang, Yangyu Tao, Zhi Yang, and Bin Cui. Node dependent local smoothing for scalable graph learning. *Advances in Neural Information Processing Systems*, 34:20321–20332, 2021. 2.2

[40] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 833–841, 2021. 2.1

[41] Kaixiong Zhou, Soo-Hyun Choi, Zirui Liu, Ninghao Liu, Fan Yang, Rui Chen, Li Li, and Xia Hu. Adaptive label smoothing to regularize large-scale graph training. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 55–63. SIAM, 2023. 1, 2.2, 5.1

[42] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7793–7804. Curran Associates, Inc., 2020. 2.1

[43] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11168–11176, 2021. 2.1

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions. In particular, they explain our method and summarize the experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the method in Section 5, noting that it behaves similarly to uniform noise under certain conditions.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide the full set of assumptions in Appendix G and the proof in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide an anonymous link to the source code, and the hyperparameters used in our experiments are reported in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide an anonymous link to the source code, along with the requirements for running it and the hyperparameter space used in our experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide data splits, hyperparameters, and other details in Section 5 and Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide 95% confidence intervals for our main results in Table 1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details of the computational resources in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our paper follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper aims at node classification and does not appear to have any direct societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not pose risks of misuse or dual use.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers that introduced the code packages and datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the document via an anonymous code link.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Algorithms related to iterative pseudo-labeling

Algorithm 1 and Algorithm 2 present the detailed algorithms for PosteL using pseudo-labels and the training process involving iterative pseudo-labeling.

Algorithm 1 Posterior label smoothing using pseudo-labels

Require: The set of training nodes $\mathcal{V}_{\text{train}}$ and the set of nodes with pseudo-label $\mathcal{V}_{\text{pseudo}}$; the number of classes K; one-hot encoding of node labels $\{\boldsymbol{e}_i\}_{i\in\mathcal{V}_{\text{train}}\cup\mathcal{V}_{\text{pseudo}}}$; and the hyperparameters α and β.

Ensure: The set of soft labels $\{\hat{\boldsymbol{e}}_i\}_{i\in\mathcal{V}_{\text{train}}}$.

Initialize the set of labeled nodes: $\mathcal{V}_{\text{labeled}} = \mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{pseudo}}$

Estimate prior distribution for $m \in [K]$: $P(\hat{Y}_i = m) = \sum_{u\in\mathcal{V}_{\text{labeled}}} e_{um}/|\mathcal{V}_{\text{labeled}}|$.

Define the set of labeled neighbors for each node u: $\mathcal{N}_{\text{labeled}}(u) = \mathcal{N}(u) \cap \mathcal{V}_{\text{labeled}}$.

Estimate the empirical conditional for $n, m \in [K]$:

$$P(Y_j = m|\hat{Y}_i = n) \propto \sum_{u:u\in\mathcal{V}_{\text{labeled}}, y_u=n} \sum_{v\in\mathcal{N}_{\text{labeled}}(u)} e_{vm}.$$

for each $i \in \mathcal{V}_{\text{train}}$ **do**

Approximate the likelihood:

$$P(\{Y_j = y_j\}_{j\in\mathcal{N}_{\text{labeled}}(i)}|\hat{Y}_i = k) \approx \prod_{j\in\mathcal{N}_{\text{labeled}}(i)} P(Y_j = y_j|\hat{Y}_i = k).$$

Compute posterior distribution: $P(\hat{Y}_i = k \mid \{Y_j = y_j\}_{j\in\mathcal{N}_{\text{labeled}}(i)})$ using Equation (1).

Add uniform noise: $\tilde{e}_{ik} \propto P(\hat{Y}_i = k \mid \{Y_j = y_j\}_{j\in\mathcal{N}_{\text{labeled}}(i)}) + \beta\epsilon$.

Obtain the soft label: $\hat{\boldsymbol{e}}_i = \alpha\tilde{\boldsymbol{e}}_i + (1-\alpha)\boldsymbol{e}_i$.

end for

Algorithm 2 Training algorithm with iterative pseudo-labeling

Require: The input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$; the set of training nodes $\mathcal{V}_{\text{train}}$ and test nodes $\mathcal{V}_{\text{test}}$, where $\mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{test}} = \mathcal{V}$; one-hot encoded training labels $\{\boldsymbol{e}_i\}_{i\in\mathcal{V}_{\text{train}}}$; and PosteL, as described in algorithm 1, along with its parameters K, α, and β.

Ensure: Trained GNN model f with pseudo-labeled nodes.

Initialize the pseudo-labeled node set: $\mathcal{V}_{\text{pseudo}} = \emptyset$.

Initialize pseudo-labels: $\{\boldsymbol{e}_i\}_{i\in\mathcal{V}_{\text{pseudo}}} = \emptyset$.

while validation loss is decreasing **do**

Apply posterior label smoothing:

$$\{\hat{\boldsymbol{e}}_i\}_{i\in\mathcal{V}_{\text{train}}} = \text{PosteL}(\mathcal{V}_{\text{train}}, \mathcal{V}_{\text{pseudo}}, \{\boldsymbol{e}_i\}_{i\in\mathcal{V}_{\text{training}}\cup\mathcal{V}_{\text{pseudo}}}, K, \alpha, \beta).$$

Train the GNN model f to predict soft labels for the training nodes $\{\hat{\boldsymbol{e}}_i\}_{i\in\mathcal{V}_{\text{train}}}$.

Obtain pseudo-labels $\{\bar{y}_i\}_{i\in\mathcal{V}_{\text{test}}}$ and their one-hot encodings $\{\bar{\boldsymbol{e}}_i\}_{i\in\mathcal{V}_{\text{test}}}$ for test nodes:

$$\{\bar{y}_i\}_{i\in\mathcal{V}_{\text{test}}} = \{\arg\max f(\mathcal{G})_i\}_{i\in\mathcal{V}_{\text{test}}}.$$

Update the pseudo-labeled node set: $\mathcal{V}_{\text{pseudo}} = \mathcal{V}_{\text{test}}$.

Update pseudo-labels: $\{\boldsymbol{e}_i\}_{i\in\mathcal{V}_{\text{pseudo}}} = \{\bar{\boldsymbol{e}}_i\}_{i\in\mathcal{V}_{\text{test}}}$.

end while

B The proof of the lemmas

Lemma 1 (Homophilic graph)**.** *Suppose that the classes are balanced, i.e., $P(\hat{Y}=0)=P(\hat{Y}=1)$ and the graph is homophilic, i.e., $c_k > 1-c_{1-k}$. Then, for any node i with neighbors $\mathcal{N}(i)$, the posterior probability satisfies,*

$$P(\hat{Y}_i = k|\{Y_j = y_j\}_{j\in\mathcal{N}(i)}) > 0.5$$

if and only if

$$|\mathcal{N}_k(i)| > |\mathcal{N}_{1-k}(i)| \cdot \frac{\log c_{1-k} - \log(1-c_k)}{\log c_k - \log(1-c_{1-k})}.$$

Lemma 2 (Heterophilic graph)**.** *Under the same assumptions used in Lemma 1, but now with a heterophilic condition, i.e., $c_k < 1-c_{1-k}$, we have,*

$$P(\hat{Y}_i = k|\{Y_j = y_j\}_{j\in\mathcal{N}(i)}) > 0.5$$

if and only if

$$|\mathcal{N}_k(i)| < |\mathcal{N}_{1-k}(i)| \cdot \frac{\log c_{1-k} - \log(1-c_k)}{\log c_k - \log(1-c_{1-k})}.$$

Proof. In binary classification, the conditional probabilities can be expressed in terms of class homophily c_k as follows:

$$P(Y_j = k|\hat{Y}_i = k) = c_k, \tag{6}$$
$$P(Y_j = 1-k|\hat{Y}_i = k) = 1-c_k. \tag{7}$$

By substituting these conditional probabilities into Equation (1), the posterior probability of $\hat{Y}_i$ is given by:

$$P(\hat{Y}_i = k|\{Y_j = y_j\}_{j\in\mathcal{N}(i)}) = \frac{c_k^{|\mathcal{N}_k(i)|}(1-c_k)^{|\mathcal{N}_{1-k}(i)|}}{c_k^{|\mathcal{N}_k(i)|}(1-c_k)^{|\mathcal{N}_{1-k}(i)|} + c_{1-k}^{|\mathcal{N}_{1-k}(i)|}(1-c_{1-k})^{|\mathcal{N}_k(i)|}}, \tag{8}$$

$$P(\hat{Y}_i = 1-k|\{Y_j = y_j\}_{j\in\mathcal{N}(i)}) = \frac{c_{1-k}^{|\mathcal{N}_{1-k}(i)|}(1-c_{1-k})^{|\mathcal{N}_k(i)|}}{c_k^{|\mathcal{N}_k(i)|}(1-c_k)^{|\mathcal{N}_{1-k}(i)|} + c_{1-k}^{|\mathcal{N}_{1-k}(i)|}(1-c_{1-k})^{|\mathcal{N}_k(i)|}}, \tag{9}$$

where $\mathcal{N}_k(i) = \{y_j = k | j \in \mathcal{N}(i)\}$ and $\mathcal{N}_{1-k}(i) = \{y_j = 1-k | j \in \mathcal{N}(i)\}$.

The condition under which the posterior probability of the soft label $\hat{Y}_i$ for k is higher than that for $1-k$ is given by the inequality:

$$c_k^{|\mathcal{N}_k(i)|}(1-c_k)^{|\mathcal{N}_{1-k}(i)|} > c_{1-k}^{|\mathcal{N}_{1-k}(i)|}(1-c_{1-k})^{|\mathcal{N}_k(i)|}. \tag{10}$$

Taking the logarithm of both sides, the inequality expands as follows:

$$|\mathcal{N}_k(i)|\log c_k + |\mathcal{N}_{1-k}(i)|\log(1-c_k) > |\mathcal{N}_{1-k}(i)|\log c_{1-k} + |\mathcal{N}_k(i)|\log(1-c_{1-k}). \tag{11}$$

Rearranging terms yields:

$$|\mathcal{N}_k(i)|\left(\log c_k - \log(1-c_{1-k})\right) > |\mathcal{N}_{1-k}(i)|\left(\log c_{1-k} - \log(1-c_k)\right). \tag{12}$$

Finally, dividing through by $\log c_k - \log(1-c_{1-k})$, assuming it is nonzero, we obtain the condition:

$$|\mathcal{N}_k(i)| \begin{cases} > |\mathcal{N}_{1-k}(i)| \cdot \frac{\log c_{1-k} - \log(1-c_k)}{\log c_k - \log(1-c_{1-k})}, & \text{if } c_k > 1-c_{1-k}, \\ < |\mathcal{N}_{1-k}(i)| \cdot \frac{\log c_{1-k} - \log(1-c_k)}{\log c_k - \log(1-c_{1-k})}, & \text{if } c_k < 1-c_{1-k}. \end{cases} \tag{13}$$

Thus, the condition in Lemma 1 holds in the first case of Equation (13), while the condition in Lemma 2 holds in the second case of Equation (13). □

Table 2: Statistics of the dataset utilized in the experiments.

Dataset	# nodes	# edges	# features	# classes
Cora	2,708	5,278	1,433	7
CiteSeer	3,327	4,552	3,703	6
PubMed	19,717	44,324	500	3
Computers	13,752	245,861	767	10
Photo	7,650	119,081	745	8
Chameleon	2,277	31,396	2,325	5
Actor	7,600	30,019	932	5
Squirrel	5,201	198,423	2,089	5
Texas	183	287	1,703	5
Cornell	183	277	1,703	5

C Dataset statistics

We provide detailed statistics and explanations about the dataset used for the experiments in Table 2 and the paragraphs below.

Cora, CiteSeer, and PubMed Each node represents a paper, and an edge indicates a reference relationship between two papers. The task is to predict the research subjects of the papers.

Computers and Photo Each node represents a product, and an edge indicates a high frequency of concurrent purchases of the two products. The task is to predict the product category.

Chameleon and Squirrel Each node represents a Wikipedia page, and an edge indicates a link between two pages. The task is to predict the monthly traffic for each page. We use the classification version of the dataset, where labels are converted by dividing monthly traffic into five bins.

Actor Each node represents an actor, and an edge indicates that two actors appear on the same Wikipedia page. The task is to predict the category of the actors.

Texas and Cornell Each node represents a web page from the computer science department of a university, and an edge indicates a link between two pages. The task is to predict the category of each web page as one of the following: student, project, course, staff, or faculty.

D Detailed experimental setup

In this section, we provide the computer resources and search space for hyperparameters. Our experiments are executed on AMD EPYC 7513 32-core Processor and a single NVIDIA RTX A6000 GPU with 48GB of memory.

D.1 Learning Hyperparameters

We largely follow the search space outlined in He et al. [7]:

- Learning Rate: {0.001, 0.002, 0.01, 0.05}
- Weight Decay: {0, 0.0005}
- Model Depth: All GNNs have 2 layers.
- Linear Layer Dropout: Fixed at 0.5.

D.2 Model-Specific Hyperparameters

- **GCN** [13]
  - 2 layers

- Hidden dimension: 64

- **GAT** [34]
  - 2 layers
  - 8 attention heads, each with hidden dimension 8
- **APPNP** [4]
  - 2-layer MLP for feature extraction
  - Hidden dimension: 64
  - Power iteration steps: 10
  - Teleport probability: $\{0.1, 0.2, 0.5, 0.9\}$
- **MLP**
  - 2 layers
  - Hidden dimension: 64
- **ChebNet** [3]
  - 2 layers
  - Hidden dimension: 32
  - 2 propagation steps
- **GPR-GNN** [2]
  - 2-layer MLP for feature extraction
  - Hidden dimension: 64
  - Random walk path length: 10
  - PPR teleport probability: $\{0.1, 0.2, 0.5, 0.9\}$
  - Dropout ratio (propagation layers): $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
- **BernNet** [7]
  - 2-layer MLP for feature extraction
  - Hidden dimension: 64
  - Polynomial approximation order: 10
  - Dropout ratio (propagation layers): $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$

D.3 PosteL Hyperparameters

- Posterior Label Ratio α: {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}
- Uniform Noise Ratio β: {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

These two parameters control the interpolation weight between the posterior and one-hot labels (α) and the magnitude of uniform noise added to the posterior (β).

E Complete experimental results

Table 3: Classification accuracy on 10 node classification datasets. Δ represents the performance improvement achieved by PosteL compared to the backbone model trained with the ground-truth label. All results of the backbone model trained with the ground-truth label are sourced from He et al. [7].

	Homophilic					Heterophilic				
	Cora	CiteSeer	PubMed	Computers	Photo	Chameleon	Actor	Squirrel	Texas	Cornell
GCN	87.14±1.01	79.86±0.67	86.74±0.27	83.32±0.33	88.26±0.73	59.61±2.21	33.23±1.16	46.78±0.87	77.38±3.28	65.90±4.43
+LS	87.77±0.97	81.06±0.59	87.73±0.24	89.08±0.30	94.05±0.26	64.81±1.53	33.81±0.75	49.53±1.10	77.87±3.11	67.87±3.77
+KD	87.90±0.90	80.97±0.56	87.03±0.29	88.56±0.36	93.64±0.31	64.49±1.38	33.33±0.78	49.38±0.64	78.03±2.62	63.61±5.57
+SALS	88.10±1.08	80.52±0.85	87.23±0.13	88.88±0.54	93.80±0.31	63.00±1.75	33.24±0.92	49.16±0.77	70.00±3.93	58.36±7.54
+ALS	88.10±0.85	81.02±0.52	87.30±0.30	89.18±0.36	93.88±0.27	64.11±1.29	34.05±0.49	47.44±0.76	77.38±2.13	71.64±3.28
+PosteL	**88.56±0.90**	**82.10±0.50**	**88.00±0.25**	**89.30±0.23**	**94.08±0.35**	**65.80±1.23**	**35.16±0.43**	**52.76±0.64**	**80.82±2.79**	**80.33±1.80**
Δ	+1.42(↑)	+2.24(↑)	+1.26(↑)	+5.98(↑)	+5.82(↑)	+6.19(↑)	+1.93(↑)	+5.98(↑)	+3.44(↑)	+14.43(↑)
GAT	88.03±0.79	80.52±0.71	87.04±0.24	83.32±0.39	90.94±0.68	63.13±1.93	33.93±2.47	44.49±0.88	**80.82±2.13**	78.21±2.95
+LS	88.69±0.99	81.27±0.86	86.33±0.32	88.95±0.31	94.06±0.39	65.16±1.49	34.55±1.15	45.94±1.60	78.69±4.10	74.10±4.10
+KD	87.47±0.94	80.79±0.60	86.54±0.31	88.99±0.46	93.76±0.31	65.14±1.47	35.13±1.36	43.86±0.85	79.02±2.46	73.44±2.46
+SALS	88.64±0.94	81.23±0.59	86.49±0.25	88.75±0.36	93.74±0.37	62.76±1.42	33.91±1.41	42.29±0.94	74.92±4.43	65.57±10.00
+ALS	88.60±0.92	81.09±0.68	87.06±0.24	89.57±0.35	94.16±0.36	66.15±1.25	34.05±0.52	46.85±1.45	78.03±3.11	75.08±3.77
+PosteL	**89.21±1.08**	**82.13±0.64**	**87.08±0.19**	**89.60±0.29**	**94.31±0.31**	**66.28±1.14**	**35.92±0.72**	**49.38±1.05**	80.33±2.62	**80.33±1.81**
Δ	+1.18(↑)	+1.61(↑)	+0.04(↑)	+6.28(↑)	+3.37(↑)	+3.15(↑)	+1.99(↑)	+4.89(↑)	−0.49(↓)	+2.12(↑)
APPNP	88.14±0.73	80.47±0.74	88.12±0.31	85.32±0.37	88.51±0.31	51.84±1.82	39.66±0.55	34.71±0.57	90.98±1.64	91.81±1.96
+LS	89.01±0.64	81.58±0.61	88.90±0.32	87.28±0.27	94.34±0.23	**53.98±1.47**	39.44±0.78	**36.81±0.98**	91.31±1.48	89.51±1.81
+KD	89.16±0.74	81.88±0.61	88.04±0.39	86.28±0.44	93.85±0.26	52.17±1.23	**41.43±0.95**	35.28±1.10	90.33±1.64	91.48±1.97
+SALS	88.97±0.90	81.53±0.56	88.50±0.31	86.49±0.50	93.74±0.38	52.82±1.95	39.66±0.64	36.34±0.65	83.44±3.93	89.51±3.77
+ALS	88.93±0.94	81.75±0.59	**89.30±0.30**	87.32±0.23	94.33±0.24	53.44±1.99	39.89±0.67	36.11±0.81	90.82±2.62	92.13±1.48
+PosteL	**89.62±0.84**	**82.47±0.66**	89.17±0.26	**87.46±0.29**	**94.42±0.24**	53.83±1.66	40.18±0.70	36.71±0.60	**92.13±1.48**	**93.44±1.64**
Δ	+1.48(↑)	+2.00(↑)	+1.05(↑)	+2.14(↑)	+5.91(↑)	+1.99(↑)	+0.52(↑)	+2.00(↑)	+1.15(↑)	+1.63(↑)
MLP	76.96±0.95	76.58±0.88	85.94±0.22	82.85±0.38	84.72±0.34	46.85±1.51	40.19±0.56	31.03±1.18	91.45±1.14	90.82±1.63
+LS	77.21±0.97	76.82±0.66	86.14±0.35	83.62±0.88	89.46±0.44	48.23±1.23	39.75±0.63	31.10±0.80	90.98±1.64	90.98±1.31
+KD	76.32±0.94	77.75±0.75	85.10±0.29	83.89±0.53	88.23±0.38	47.40±1.75	**41.32±0.75**	32.58±0.83	89.34±1.97	91.80±1.15
+SALS	77.29±1.05	77.00±0.90	85.78±0.33	82.55±0.51	89.11±0.52	43.68±1.69	39.47±0.73	30.88±0.68	86.39±5.09	89.11±0.52
+ALS	77.59±0.69	77.24±0.82	86.43±0.43	**84.26±0.66**	89.86±0.43	48.03±1.38	39.98±0.94	31.33±0.89	91.64±3.44	91.64±1.31
+PosteL	**78.39±0.94**	**78.40±0.71**	**86.51±0.33**	84.20±0.55	**89.90±0.27**	**48.51±1.66**	40.15±0.46	**33.11±0.60**	**92.95±1.31**	**93.61±1.80**
Δ	+1.43(↑)	+1.82(↑)	+0.57(↑)	+1.35(↑)	+5.18(↑)	+1.66(↑)	−0.04(↓)	+2.08(↑)	+1.50(↑)	+2.79(↑)
ChebNet	86.67±0.82	79.11±0.75	87.95±0.28	87.54±0.43	93.77±0.32	59.28±1.25	37.61±0.89	40.55±0.42	86.22±2.45	83.93±2.13
+LS	87.22±0.99	79.70±0.63	88.48±0.29	89.55±0.38	94.53±0.37	66.41±1.16	39.39±0.73	42.55±1.11	**87.21±2.62**	84.59±2.30
+KD	87.36±0.95	80.80±0.72	88.41±0.20	89.81±0.30	94.76±0.30	61.47±1.23	**40.68±0.50**	43.88±1.97	84.75±3.61	83.61±2.30
+SALS	87.31±0.94	79.71±0.83	88.46±0.30	89.52±0.35	94.19±0.27	56.94±2.52	39.25±0.67	41.61±0.93	74.26±3.61	73.44±6.89
+ALS	87.39±0.97	79.81±0.81	88.80±0.33	89.88±0.36	**95.21±0.23**	61.09±0.63	39.61±1.12	41.98±0.85	85.57±3.28	86.39±2.30
+PosteL	**88.57±0.92**	**82.48±0.52**	**89.20±0.31**	**89.95±0.40**	94.87±0.25	**66.83±0.77**	39.56±0.51	**50.87±0.90**	86.39±2.46	**88.52±2.63**
Δ	+1.90(↑)	+3.37(↑)	+1.25(↑)	+2.41(↑)	+1.10(↑)	+7.55(↑)	+1.95(↑)	+10.32(↑)	+0.17(↑)	+4.59(↑)
GPR-GNN	88.57±0.69	80.12±0.83	88.46±0.33	86.85±0.25	93.85±0.28	67.28±1.09	39.92±0.67	50.15±1.92	92.95±1.31	91.37±1.81
+LS	88.82±0.99	79.78±1.06	88.24±0.42	88.39±0.48	93.97±0.33	67.90±1.01	39.72±0.70	53.39±1.80	92.79±1.15	90.49±2.46
+KD	**89.33±1.03**	**81.24±0.85**	89.85±0.56	87.88±1.11	94.23±0.51	66.76±1.31	**42.00±0.63**	53.26±1.07	**94.26±1.48**	88.52±1.97
+SALS	88.78±0.90	80.71±0.91	90.12±0.46	88.63±0.35	94.23±0.65	65.16±1.49	39.67±0.73	44.75±1.45	73.61±3.44	82.46±2.95
+ALS	88.93±1.31	80.31±0.71	90.23±0.50	89.14±0.48	94.55±0.53	67.79±1.07	40.09±0.72	51.34±1.00	92.95±1.31	89.18±2.13
+PosteL	89.20±1.07	81.21±0.64	**90.57±0.31**	**89.84±0.43**	**94.76±0.38**	**68.38±1.12**	40.08±0.69	**53.54±0.79**	93.28±1.31	**92.46±0.99**
Δ	+0.63(↑)	+1.09(↑)	+2.11(↑)	+2.99(↑)	+0.91(↑)	+1.10(↑)	+0.16(↑)	+3.39(↑)	+0.33(↑)	+1.09(↑)
BernNet	88.52±0.95	80.09±0.79	88.48±0.41	87.64±0.44	93.63±0.35	68.29±1.58	**41.79±1.01**	51.35±0.73	93.12±0.65	92.13±1.64
+LS	88.80±0.92	80.37±1.05	87.40±0.27	88.32±0.38	93.70±0.21	69.58±0.94	39.60±0.53	52.39±0.60	91.80±1.80	90.49±1.48
+KD	87.78±0.99	81.20±0.86	87.59±0.41	87.35±0.40	93.96±0.40	67.75±1.42	41.04±0.89	51.25±0.83	93.61±1.31	90.33±2.30
+SALS	88.77±0.85	81.20±0.61	88.61±0.35	88.87±0.33	94.22±0.43	64.62±0.85	40.15±1.07	46.19±0.78	85.90±4.10	88.03±3.12
+ALS	89.13±0.79	81.17±0.67	**89.19±0.46**	89.52±0.30	**94.54±0.32**	67.92±1.07	40.51±0.61	51.83±1.31	93.77±1.31	92.79±1.48
+PosteL	**89.39±0.92**	**82.46±0.67**	89.07±0.29	**89.56±0.35**	**94.54±0.36**	**69.65±0.83**	40.40±0.67	**53.11±0.87**	**93.93±1.15**	**92.95±1.80**
Δ	+0.87(↑)	+2.37(↑)	+0.59(↑)	+1.92(↑)	+0.91(↑)	+1.36(↑)	−1.39(↓)	+1.76(↑)	+0.81(↑)	+0.82(↑)
OrderedGNN	88.62±1.05	80.11±0.86	88.74±0.56	89.72±0.50	94.76±0.36	58.27±1.33	39.73±1.15	38.70±1.10	90.16±2.63	90.33±2.46
+LS	88.52±0.94	80.23±0.80	88.16±0.33	89.59±0.47	94.49±0.45	58.86±1.62	40.01±0.66	40.12±0.82	88.20±3.61	91.15±1.31
+KD	88.26±1.07	80.52±0.83	88.23±0.21	89.35±0.34	94.40±0.23	58.21±1.18	40.17±0.45	40.92±0.87	**90.49±1.48**	91.31±1.80
+SALS	88.44±0.97	80.93±0.72	88.08±0.62	88.94±0.51	93.87±0.35	59.30±1.25	39.52±0.41	40.85±0.86	77.70±4.75	84.75±4.10
+ALS	87.96±0.74	80.60±0.57	88.69±0.57	89.84±0.48	94.76±0.36	59.39±1.23	**40.28±0.79**	40.37±1.05	90.00±2.62	89.84±2.95
+PosteL	**88.97±1.15**	**82.54±0.64**	**88.85±0.61**	**90.13±0.29**	**94.96±0.34**	**60.15±1.20**	39.99±1.00	**43.72±0.85**	87.70±5.25	**91.97±1.15**
Δ	+0.35(↑)	+2.43(↑)	+0.11(↑)	+0.41(↑)	+0.20(↑)	+1.88(↑)	+0.26(↑)	+5.02(↑)	−2.46(↓)	+1.64(↑)

F Complexity analysis

In this section, we analyze the time complexity of Section 3.1 in detail. Specifically, we first show the complexities of deriving the prior and likelihood distributions independently, and then combine these results to determine the overall complexity of computing the posterior distribution.

First, the prior distribution $P(\hat{Y}_i = m)$ can be obtained as follows:

$$\hat{P}(Y_i = m) = \frac{|\{u \mid y_u = k\}|}{|\mathcal{V}|} = \frac{\sum_{u \in \mathcal{V}} e_{um}}{|\mathcal{V}|}. \tag{14}$$

The time complexity of calculating Equation (14) is $O(|\mathcal{V}|)$, so the time complexity of calculating the prior distribution for K classes is $O(|\mathcal{V}|K)$.

Next, calculating the empirical conditional $P(Y_j = m|\hat{Y}_i = n)$ from Equation (3) can be performed as follows:

$$P(Y_j = m|\hat{Y}_i = n) \propto \sum_{u: u \in \mathcal{V}, y_u = n} \sum_{v \in \mathcal{N}(u)} e_{vm}. \tag{15}$$

The time complexity of calculating Equation (15) for all possible pairs of m and n is $O(\sum_{u \in \mathcal{V}} |\mathcal{N}(u)|K)$. Since $\sum_{u \in \mathcal{V}} \mathcal{N}(u) = 2|\mathcal{E}|$, the time complexity for calculating empirical conditional is $O(|\mathcal{E}|K)$.

The likelihood is approximated through the product of empirical conditional distributions, denoted as $P(\{Y_j = y_j\}_{j \in \mathcal{N}(i)}|\hat{Y}_i = k) \approx \prod_{j \in \mathcal{N}(i)} P(Y_j = y_j|\hat{Y}_i = k)$. Likelihood calculation for all training nodes operates in $O(\sum_{u \in \mathcal{V}}|\mathcal{N}(u)|K)$ time complexity. So the overall computational complexity for likelihood calculation is $O(|\mathcal{E}|K)$.

After obtaining the prior distribution and likelihood, the posterior distribution is obtained by Bayes' rule in Equation (1). Applying Bayes' rule for $|\mathcal{V}|$ nodes and K classes can be done in $O(|\mathcal{V}|K)$. So the overall time complexity is $O\left(\left(|\mathcal{E}| + |\mathcal{V}|\right)K\right)$. In most cases, $|\mathcal{V}| < |\mathcal{E}|$, so the time complexity of PosteL is $O(|\mathcal{E}|K)$.

In Section 3.2, we introduce an iterative pseudo-labeling procedure that repeatedly refines the pseudo-labels of validation and test nodes to compute posterior labels. Because each iteration requires retraining the model from scratch, the number of iterations can become a significant bottleneck in terms of runtime. Consequently, we evaluate the iteration counts to assess this overhead. The average number of iterations for each backbone and dataset in Table 3 is presented in Table 4. With an overall mean iteration count of 1.13, we argue that this level of additional time investment is justifiable for the sake of performance enhancement.

Table 4: Average iteration counts of iterative pseudo-labeling for each backbone and dataset used to report Table 3.

	Cora	CiteSeer	PubMed	Computers	Photo	Chameleon	Actor	Squirrel	Texas	Cornell
GCN+PosteL	2.5	2.2	1.5	1	0.9	0.9	1.1	0.7	1.8	2.5
GAT+PosteL	1.6	1.8	1	1.2	0.7	0.8	2	1.1	3.1	2.4
APPNP+PosteL	1.9	2	1.1	0.8	1.1	1	1.1	0.9	1.4	2.9
MLP+PosteL	1.7	2.2	0.4	0.7	0.7	0.1	0.8	0.6	0.9	2.4
ChebNet+PosteL	1.6	2.1	1.2	0.6	0.6	1	0.7	0.7	2	2
GPR-GNN+PosteL	0.8	1.1	0.8	0.5	1.3	1	0.3	0.7	1.1	1
BernNet+PosteL	1.5	1.8	0.9	0.8	1	1.5	1.5	0.5	1.2	2.1

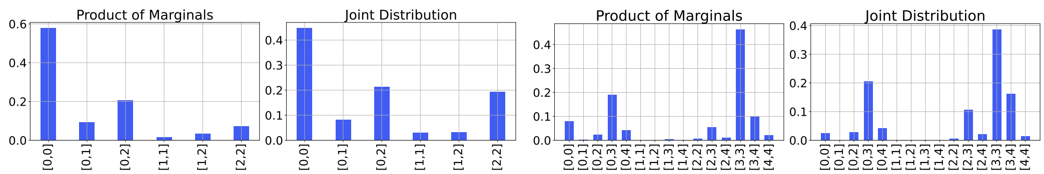
Table 5 presents the average training times of PosteL and other baselines across all datasets in Section 5, using a GCN backbone. When iterative pseudo-labeling is employed, PosteL is approximately four times slower than using the ground-truth labels, while requiring a training time comparable to knowledge distillation and ALS. If this overhead is excessive, PosteL can be applied without iterative pseudo-labeling or with only a single pseudo-labeling iteration. In particular, PosteL without pseudo-labeling trains approximately three times faster than knowledge distillation and ALS, and one pseudo-labeling iteration is still faster than those methods while achieving comparable accuracy. Table 8 summarizes the accuracy results for each variant.

Table 5: Overall training time for each smoothing method. PosteL (w/o) refers to PosteL without iterative pseudo-labeling, and PosteL (1) refers to PosteL with one iteration of pseudo-labeling.

	GCN	+LS	+KD	+SALS	+ALS	+PosteL	+PosteL (w/o)	+PosteL (1)
time (s)	0.83	0.84	3.54	0.90	2.87	3.38	1.08	1.94

G Empirical analysis

Empirical validation of the conditional independence in Equation (2) In Equation (2), we approximate the joint conditional distribution of neighborhood labels using the product of individual conditional distributions. Although this factorization is exact when the neighborhood labels are conditionally independent given the central node's label, this assumption is often violated in real-world datasets. To empirically validate our approximation, we compare the true joint distribution $P(Y_j = n, Y_k = m|Y_i = l)$ to the product of marginals $P(Y_j = n|Y_i = l) \times P(Y_k = m|Y_i = l)$. Figure 2 illustrates these distributions for the case $l = 0$. We observe that the product of marginals closely approximates the joint distribution, supporting the validity of our approximation.



(a) PubMed (b) Cornell

Figure 2: Estimated likelihood via product of marginals $P(Y_j|Y_i = 0, j \in \mathcal{N}(i)) \times P(Y_k|Y_i = 0, k \in \mathcal{N}(i))$ and empirical joint distribution $P(Y_j, Y_k|Y_i = 0, j, k \in \mathcal{N}(i))$.

Loss curves analysis We examine how soft labels affect GNN training dynamics by plotting the loss curves of GCN on the Squirrel dataset. Figure 3 compares training, validation, and test losses when using ground-truth labels, SALS labels, and PosteL labels. With PosteL, the gap between training and validation/test losses is noticeably smaller, indicating reduced overfitting; while other methods overfit after 50 epochs, PosteL remains stable through 200 epochs.

We hypothesize that correctly predicting PosteL labels, which encode local neighborhood information, enhances the model's understanding of the graph structure and thereby improves generalization. Similar context-prediction strategies have been used as pretraining methods in previous studies [9, 23]. Loss curves for homophilic datasets are provided in Figure 8 and heterophilic in Figure 9 in Appendix H, showing consistent patterns across datasets.

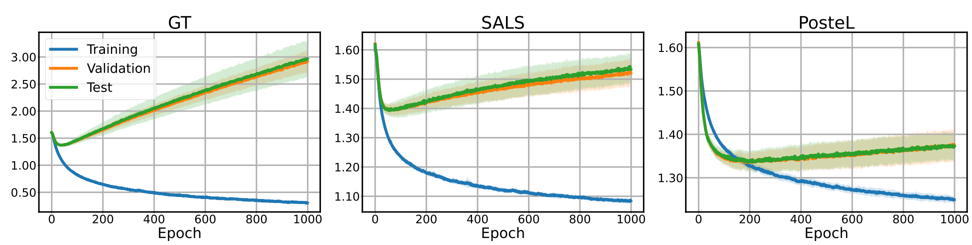


Figure 3: Loss curve of GCN trained on PosteL, SALS, and ground-truth labels on the Squirrel dataset.

Effect of iterative pseudo-labeling We analyze the impact of iterative pseudo-labeling by examining the loss curves across iterations. Figure 4 shows the loss curves on the Cornell dataset, where validation and test losses consistently decrease with each iteration. In this example, the model achieves its best performance after four iterations. On average, the best validation performance is observed at 1.13 iterations. The average number of iterations used to report the results in Table 3 is detailed in Appendix F.

Hyperparameter sensitivity analysis Figure 5 shows the performance with varying values of α and β on GCN. The blue line indicates the performance with varying α, and the green line shows the performance with varying β. The red dotted line represents the performance with the ground-truth label. Regardless of the values of α and β, the performance consistently outperforms the case using ground-truth labels, indicating that PosteL is insensitive to α and β. We observe that α values greater than 0.8 may harm training, suggesting the necessity of interpolating ground-truth labels.

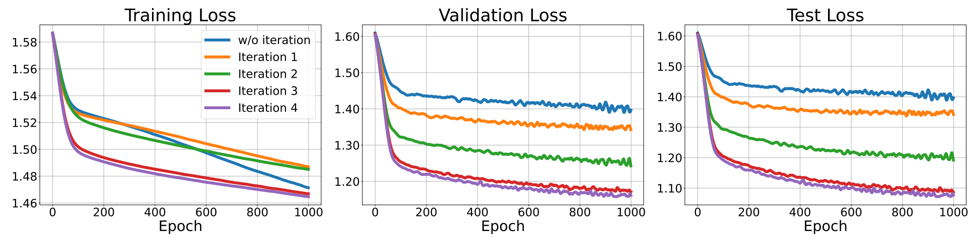


Figure 4: The loss curves of GCN on the Cornell dataset with the iterative pseudo-labeling.

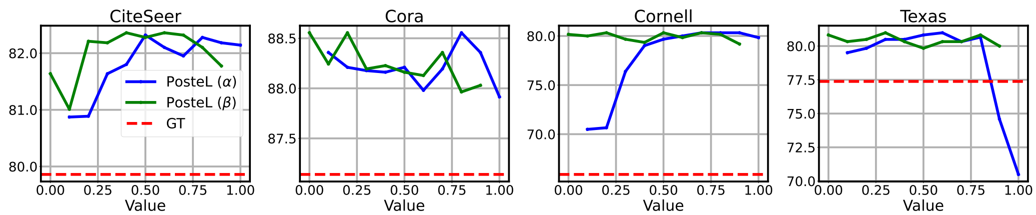


Figure 5: Hyperparameter sensitivity analysis on GCN.

Scalability to large-scale graphs We evaluated the runtime of PosteL on the ogbn-products dataset [10], which contains 2,449,029 nodes and 61,859,140 edges, to assess its computational efficiency on a large-scale graph. We measured the time excluding the training time for iterative pseudo-labeling. Generating soft labels with PosteL takes 5.65 seconds, whereas a single training epoch requires 19.11 seconds, indicating that PosteL can generate soft labels efficiently even on large-scale graphs.

Table 6: The accuracy of label smoothing methods on the ogbn-products dataset using GCN.

	GCN	+LS	+SALS	+ALS	+PosteL
ogbn-products	80.62±0.68	80.99±0.50	81.12±0.13	80.46±0.38	**81.20±0.68**

Table 6 presents the performance of PosteL on the ogbn-products dataset using GCN. Although the improvement is not statistically significant, PosteL achieves the highest performance compared to other smoothing methods.

Design Choices of Likelihood Model We investigate alternative likelihood designs and introduce two PosteL variants: PosteL (normalized) and PosteL (local-H). In Equation (3), each edge equally contributes to the conditional probability, which can over-rely on high-degree nodes. To mitigate this, PosteL (normalized) adjusts edge contributions based on node degrees:

$$P^{\text{norm.}}(Y_j = m|\hat{Y}_i = n) := \frac{\sum_{y_u=n}\sum_{v\in\mathcal{N}(u)} \frac{1}{|\mathcal{N}(u)|}\cdot \mathbb{1}[y_v = m]}{|\{y_u = n \mid u \in \mathcal{V}\}|}\ , \tag{16}$$

where $\mathbb{1}$ is an indicator function.

In PosteL (local-H), likelihoods and priors are estimated from H-hop ego graphs, emphasizing local neighborhood statistics:

$$P^{\text{local-}H}(Y_j = m|\hat{Y}_i = n) := \frac{|\{(u,v)|y_v = m, y_u = n, (u,v)\in\mathcal{E}_i^{(H)}|}{|\{(u,v)|y_u = n, (u,v)\in\mathcal{E}_i^{(H)}|}\ , \tag{17}$$

where $\mathcal{E}_i^{(H)}$ is the set of edges in the H-hop neighborhood of node i, denoted as $\mathcal{N}^{(H)}(i)$. These variants allow us to assess the importance of global versus local statistics in the smoothing process.

Table 7 shows the comparison between these variants. The likelihood with global statistics, e.g., PosteL and PosteL (normalized), performs better than the local likelihood methods, e.g., PosteL (local-1) and PosteL (local-2) in general, highlighting the importance of simultaneously utilizing global statistics. Especially in the Cornell dataset, a significant performance gap between PosteL and PosteL (local) is observed. PosteL (normalized) demonstrates similar performance to PosteL.

Table 7: Classification accuracy with various choices of likelihood model. PosteL (local-1) and (local-2) indicate that the likelihood is estimated within one- and two-hop neighbors of a target node, respectively. PosteL (norm.), shortened from PosteL (normalized), indicates that the likelihood is normalized based on the degree of a node.

	Cora	CiteSeer	Computers	Photo	Chameleon	Actor	Texas	Cornell
GCN	87.14±1.01	79.86±0.67	83.32±0.33	88.26±0.73	59.61±2.21	33.23±1.16	77.38±3.28	65.90±4.43
+PosteL (local-1)	88.26±1.07	81.42±0.46	89.08±0.31	93.61±0.40	65.36±1.25	33.48±1.03	79.02±3.11	71.97±4.10
+PosteL (local-2)	88.62±0.97	81.92±0.42	88.62±0.48	93.95±0.37	65.10±1.55	34.63±0.46	78.20±2.79	73.28±4.10
+PosteL (norm.)	**89.00±0.99**	81.86±0.70	**89.30±0.39**	**94.13±0.39**	**66.00±1.14**	34.90±0.63	80.33±2.95	80.00±1.97
+PosteL	88.56±0.90	**82.10±0.50**	**89.30±0.23**	94.08±0.35	65.80±1.23	**35.16±0.43**	**80.82±2.79**	**80.33±1.80**

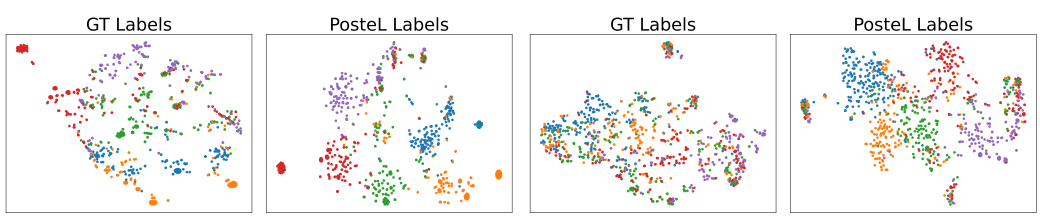
Ablation Studies We conduct ablation studies on three components of PosteL: posterior smoothing (PS), uniform noise (UN), and iterative pseudo-labeling (IPL), to evaluate their individual contributions. Table 8 summarizes the results.

Table 8: Ablation studies on three main components of PosteL on GCN. PS stands for posterior label smoothing, UN stands for uniform noise, and IPL stands for iterative pseudo-labeling. We use ✓to indicate the presence of the corresponding component in training and ✗ to indicate its absence. IPL with one indicates the performance with a single pseudo-labeling step.

PS	UN	IPL	Cora	CiteSeer	Computers	Photo	Chameleon	Actor	Texas	Cornell
✗	✗	✗	87.14±1.01	79.86±0.67	83.32±0.33	88.26±0.73	59.61±2.21	33.23±1.16	77.38±3.28	65.90±4.43
✓	✗	✗	88.11±1.22	80.95±0.52	88.86±0.40	93.55±0.30	64.53±1.23	33.48±0.62	78.52±2.46	68.52±4.43
✗	✓	✗	87.77±0.97	81.06±0.59	89.08±0.30	94.05±0.26	64.81±1.53	33.81±0.75	77.87±3.11	67.87±3.77
✓	✗	✓	**88.56±0.90**	81.64±0.57	88.70±0.27	93.70±0.37	64.25±1.93	34.71±0.76	**80.82±2.79**	80.16±1.97
✓	✓	✗	87.83±0.92	82.09±0.44	89.17±0.31	93.98±0.34	**66.19±1.60**	34.91±0.48	79.51±3.61	71.97±5.25
✓	✓	1	87.96±0.90	**82.33±0.52**	89.16±0.30	94.06±0.27	65.89±1.51	34.96±0.48	80.16±2.79	**80.33±1.97**
✓	✓	✓	**88.56±0.90**	82.10±0.50	**89.30±0.23**	**94.08±0.35**	65.80±1.23	**35.16±0.43**	**80.82±2.79**	**80.33±1.80**

The best performance is achieved when all components are included, highlighting their collective importance. IPL consistently improves performance across most datasets, especially Cornell, although omitting IPL still yields competitive results. Adding uniform noise further enhances performance on several datasets. Notably, PosteL surpasses using only uniform noise, a common label smoothing baseline.

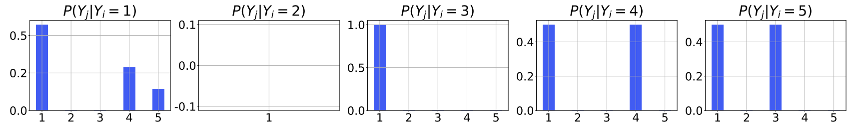
Visualization of node embeddings Figure 6 presents the t-SNE [31] plots of node embeddings from the GCN with the Chameleon and Squirrel datasets. The node color represents the label. For each dataset, the left plot visualizes the embeddings with the ground-truth labels, while the right plot visualizes the embeddings with PosteL labels. The visualization shows that the embeddings from the soft labels form tighter clusters compared to those trained with the ground-truth labels. This visualization results coincide with the t-SNE visualization of the previous work of Müller et al. [18].



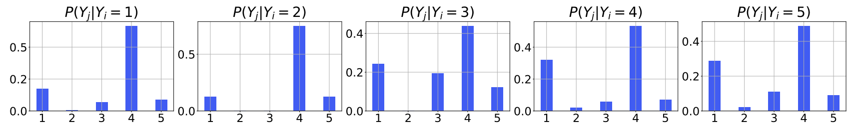
(a) Chameleon (b) Squirrel

Figure 6: t-SNE plots of the final layer representation of the Chameleon and Squirrel datasets. For each dataset, the left figure displays the representations trained on the ground-truth labels, while the right figure displays the representations trained on the PosteL labels.

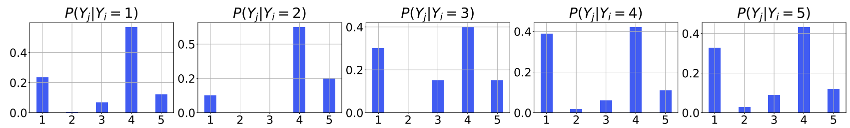
Posterior estimation with limited training labels Our method estimates posterior probabilities from training set statistics. However, when training labels are limited, these estimated distributions may substantially deviate from the oracle distributions, potentially leading to inaccurate posterior probabilities. To examine this issue, we evaluate the quality of the estimated distributions using only 10% of the training data described in Section 5.1.



(a) Training labels



(b) Training labels with pseudo-labels



(c) Ground-truth labels (Oracle)

Figure 7: Estimated conditional distributions obtained from (a) training labels only, (b) training labels combined with pseudo-labels, and (c) all ground-truth labels.

Table 9: Accuracy of the model trained on a limited number of labeled nodes.

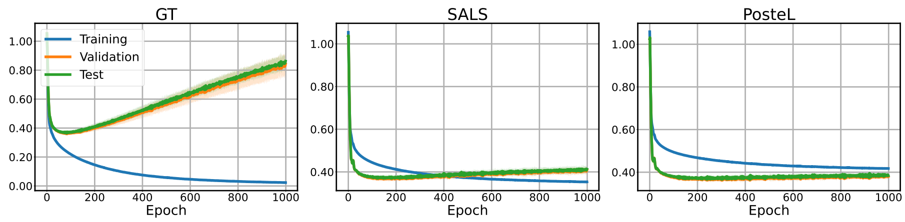
	Cora	CiteSeer	Computers	Photo	Chameleon	Actor	Texas	Cornell
GCN	80.66±0.89	73.52±1.43	84.47±0.99	92.38±0.41	45.01±3.52	24.62±5.83	67.05±14.92	58.36±19.19
+PosteL (w/o IPL)	81.59±1.23	74.97±1.62	**85.56±0.57**	92.62±0.37	51.05±2.30	30.08±2.65	69.67±14.76	64.59±15.25
+PosteL	**82.33±1.28**	**76.15±1.05**	85.50±0.50	**92.99±0.31**	**51.49±2.28**	**31.25±2.59**	**71.48±13.93**	**67.54±16.40**

Figure 7 compares the conditional distributions on the Cornell dataset estimated using (1) training labels only, (2) training labels combined with pseudo-labels for validation and test nodes, and (3) all ground-truth labels. The conditional distributions estimated from limited training data show substantial deviation from the oracle distributions derived from all labels. In contrast, incorporating pseudo-labels reduces this discrepancy, yielding conditional distributions that closely match the oracle. We provide the same analysis on the other datasets in Appendix I.

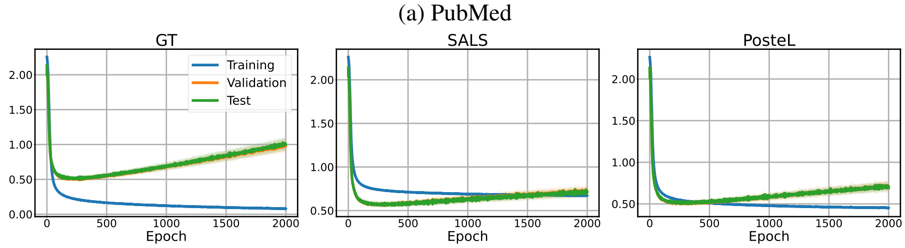
Table 9 reports the classification accuracy of GCNs trained on 10% of the training data. Despite the limited availability of training labels, PosteL consistently enhances predictive accuracy. Particularly in the Texas and Cornell datasets, where pseudo-labeling substantially improves conditional distribution estimation, iterative pseudo-labeling achieves greater improvements compared to other datasets. This highlights the importance of refining conditional distributions to estimate posterior probabilities accurately.

H Learning curves analysis for all datasets

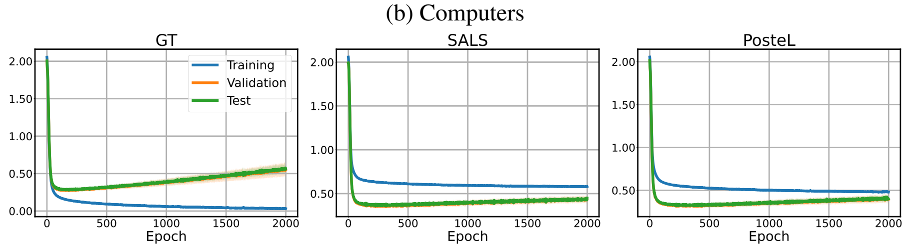
The learning curves for all datasets are provided in Figure 8 and Figure 9.



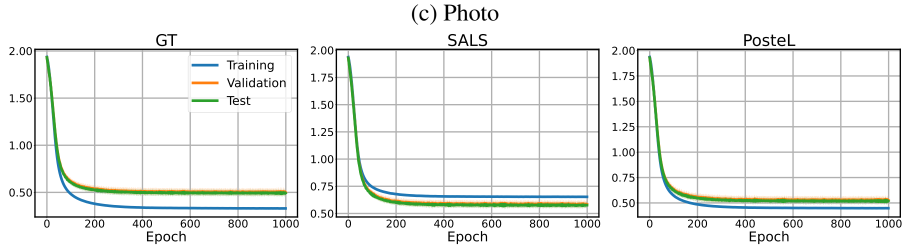
(a) PubMed



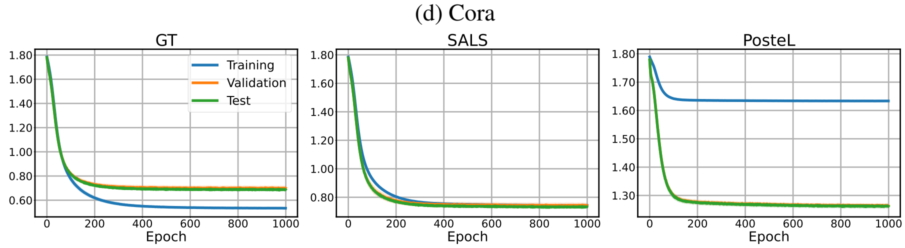
(b) Computers



(c) Photo

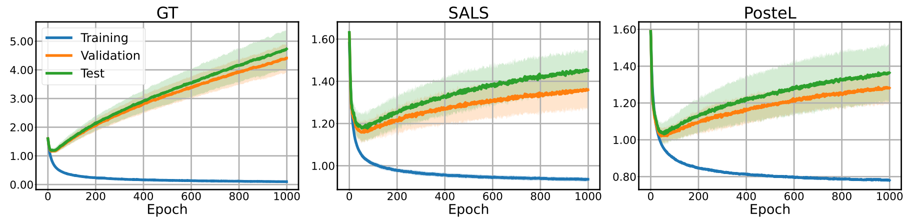


(d) Cora

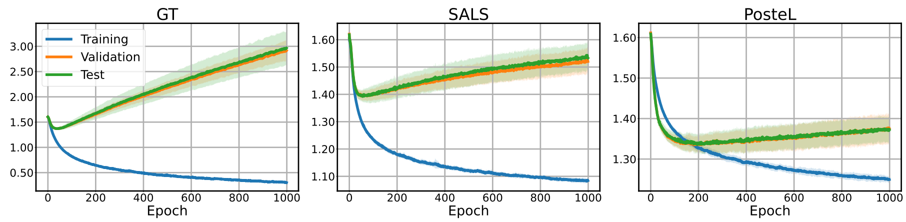


(e) CiteSeer

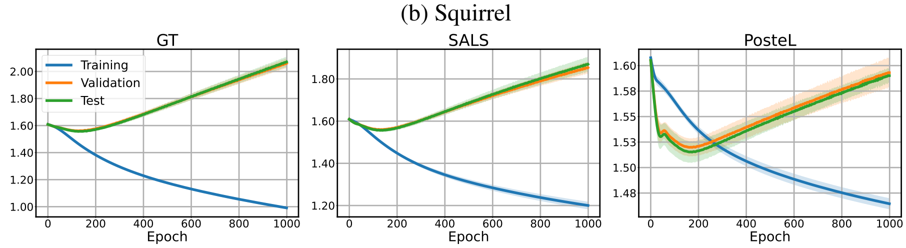
Figure 8: Loss curve of GCN trained on PosteL labels, SALS labels, and ground truth labels on homophilic datasets.



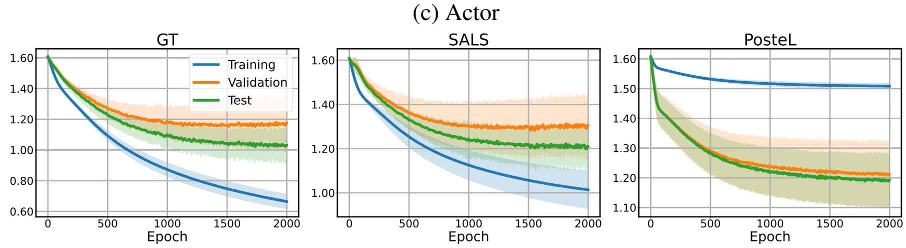
(a) Chameleon



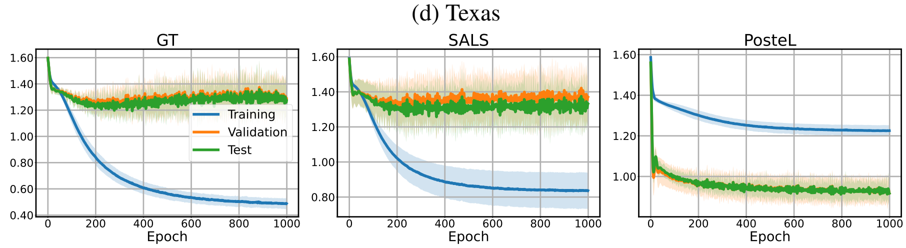
(b) Squirrel



(c) Actor



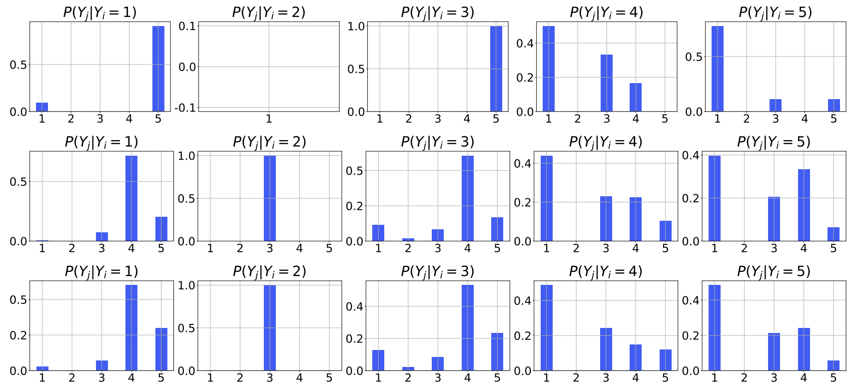
(d) Texas



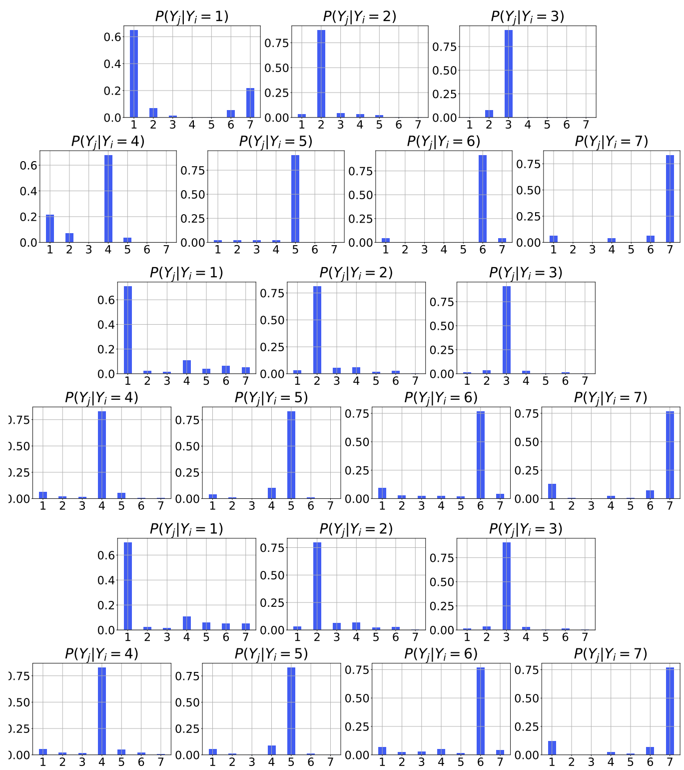
(e) Cornell

Figure 9: Loss curve of GCN trained on PosteL labels, SALS labels, and ground truth labels on heterophilic datasets.

I Empirical conditional distributions for the Cora and Texas datasets



(a) Texas



(b) Cora

Figure 10: Estimated conditional distributions based on training labels only (top), training labels with pseudo-labels (middle), and all ground-truth labels (bottom).