Unraveling the Impact of Heterophilic Structures on Graph Positive-Unlabeled Learning

Yuhao Wu [1] **Jiangchao Yao** [2 3] **Bo Han** [4] **Lina Yao** [5] **Tongliang Liu** [1]

Abstract

While Positive-Unlabeled (PU) learning is vital in many real-world scenarios, its application to graph data still remains under-explored. We unveil that a critical challenge for PU learning on graph lies on the edge heterophily, which directly violates the *irreducibility assumption* for *Class-Prior Estimation* (class prior is essential for building PU learning algorithms) and degenerates the latent label inference on unlabeled nodes during classifier training. In response to this challenge, we introduce a new method, named *Graph PU Learning with Label Propagation Loss* (GPL). Specifically, GPL considers learning from PU nodes along with an intermediate heterophily reduction, which helps mitigate the negative impact of the heterophilic structure. We formulate this procedure as a bilevel optimization that reduces heterophily in the inner loop and efficiently learns a classifier in the outer loop. Extensive experiments across a variety of datasets have shown that GPL significantly outperforms baseline methods, confirming its effectiveness and superiority.

1. Introduction

Positive-Unlabeled (PU) learning (Denis, 1998; De Comité et al., 1999; Denis et al., 2005), where a binary classifier is trained from positive and unlabeled samples, has made remarkable progress on image (Kiryo et al., 2017; Niu et al., 2016; Garg et al., 2021) and text (Li et al., 2016) data. Nowadays, there is burgeoning interest in extending PU learning to graph-structured data due to the ubiquitous nature of real-world contexts (Hu et al., 2020; Gaudelet et al., 2021; Veličković, 2023; Wu et al., 2024). For instance, in online transaction networks, fraudsters are labeled as positive nodes only upon being detected (Yoo et al., 2021), and a similar case holds in pandemic graph prediction (Panagopoulos et al., 2021) where only affected nodes are identified as positive, while the remaining nodes are actually label-agnostic.

However, most current approaches to PU learning are built on the hypothesis that each sample is independently generated. Such a premise hinders these methods from readily adapting to graph data where nodes are interdependent by edges (Wu et al., 2022; Zhao et al., 2020). Our investigation reveals that a primary challenge in adapting these approaches to graph data is the prevalence of *heterophilic structure*, where positive and negative nodes are connected (Zhu et al., 2020; Zhang et al., 2019). Such heterophily widely existed in standard graph learning (Platonov et al., 2023; Luan et al., 2022), intensifies when dealing with only positive and unlabeled nodes of interest.

The heterophilic structure impedes two aspects of PU learning: (i) It results in the latent feature entanglement of positive and negative nodes due to heterophilic edges, which directly challenges *Class-Prior Estimation* (CPE), *i.e*, estimating the fraction of positive nodes among the unlabeled nodes. CPE methods rely on the *irreducibility* assumption, which holds that the patterns belonging to positive nodes could not possibly be confused with patterns from negative nodes. (Garg et al., 2021; Bekker & Davis, 2020; Ivanov, 2020). However, heterophilic structures violate this assumption, yielding an overestimated class prior. (ii) The presence of heterophilic structure complicates the latent label inference for unlabeled nodes. More precisely, the heterophilic edges between different classes mislead the message passing among same-class nodes, hindering the accurate inference of latent labels for unlabeled nodes during the classifier training process (termed as *PU classification* for simplicity) (Garg et al., 2021; Yao et al., 2020). The adverse effect to CPE and PU classification significantly challenges graph PU learning when heterophilic structure exists.

In our paper, we present a *Graph PU Learning with Label Propagation Loss* (GPL) method to alleviate the side effect of heterophilic structures on PU learning. GPL mitigates the impact of heterophilic edges by reducing their weights based

[1] Sydney AI Center, The University of Sydney [2] CMIC, Shanghai Jiao Tong University [3] Shanghai AI Laboratory [4] TMLR Group, Department of Computer Science, Hong Kong Baptist University [5] University of New South Wales. Correspondence to: Tongliang Liu <tongliang.liu@sydney.edu.au>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

on a proposed Label Propagation Loss (LPL). Specifically, optimizing LPL results in strengthening the weights of homophilic edges, *i.e*, edges linking same-class nodes, while simultaneously reducing the weights of heterophilic edges, *relying solely on the observed positive nodes*. This approach is effective because homophilic edges inherently support the label propagation process, whereas heterophilic edges between different classes are counterproductive. Therefore, by matching the given/predicted labels and the propagated labels, we can reduce the impact of heterophilic edges, which then promotes the accuracy of CPE and PU classification.

Taking LPL and the vanilla graph PU learning interactively forms the basic idea of GPL: the diminished heterophilic structure by LPL creates a conducive context for graph PU learning. The classifier trained through graph PU learning, in turn, helps refine this structure with the aid of more accurate prediction labels. We formulate the optimization of LPL and graph PU learning as an efficient bilevel optimization, where the heterophily is reduced in the inner loop and the classifier is learnt in the outer loop. In a nutshell, our contributions can be summarized into the following points:

- We first identify the challenge of learning from PU nodes with heterophilic structures. Through both theoretical analysis and empirical evidence, we formally elucidate the difficulty encountered in CPE and PU classification when the heterophilic structure exists.
- We propose an end-to-end GPL method, which considers the optimization of the graph structure to reduce the negative impact of heterophilic edges on graph PU learning. This reduction has been theoretically proved. Besides, GPL leverages a bilevel algorithm, obviating the need for a known class-prior and eliminating the assumption of a homophily graph structure.
- We conduct extensive experiments on datasets ranging from homophily to heterophily to demonstrate the superiority of our proposed framework over the baselines. Furthermore, we provide comprehensive analyses of the underlying mechanisms of our framework.

2. Preliminaries

2.1. Notation

We consider a set of positive (P) and negative (N) nodes with indices $i \in \{1, 2, \cdots, n\}$, represented by an observed undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ denotes the node set and $\mathcal{E} = \{e_{ij}\}$ denotes the edge set. The observed edges induce an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ where $\boldsymbol{A}_{ij} = 1$ if nodes i and j are connected by edges e_{ij} and $\boldsymbol{A}_{ij} = 0$ otherwise. Moreover, each node v_i has an input feature vector denoted by $\boldsymbol{x}_i \in \mathbb{R}^D$ where D is the dimension, and a latent binary label $y_i \in \{-1, +1\}$.

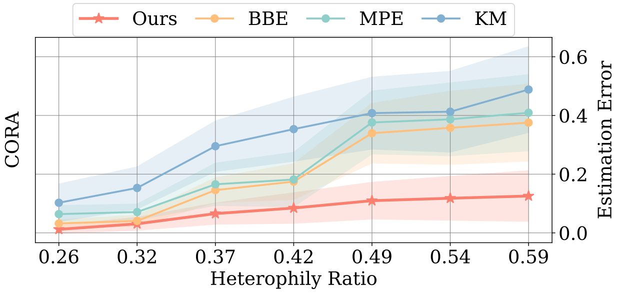

Figure 1: Estimation error of our method and the baseline estimator for Class-Prior Estimation, under varying heterophily ratio. The way to modify the graph structure with different heterophily ratios is referred to (Ma et al., 2021).

Here, assume the P and N nodes are drawn from the P and N class-conditional density $[\mathbb{P}_{\mathrm{p}}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}}), \mathbb{P}_{\mathrm{n}}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}})] := [\mathbb{P}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}}|y = +1), \mathbb{P}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}}|y = -1)]$, where $\mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}} = (\{\boldsymbol{x}_j\}_{j\in\mathcal{N}_{\boldsymbol{x}}}, \boldsymbol{A}_{\boldsymbol{x}})$ denotes the ego-graph centered at node $\boldsymbol{x}$, $\mathcal{N}_{\boldsymbol{x}}$ is the set of neighboring nodes (within a certain number of hops) in the ego-graph of $\boldsymbol{x}$ and $\boldsymbol{A}_{\boldsymbol{x}}$ is its associated adjacency matrix. In learning from positive and unlabeled (U) nodes, we only know a fraction of P nodes, denoted as a set $\mathcal{P} \sim \mathbb{P}_{\mathrm{p}}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}})$. The other $\mathcal{V}\backslash\mathcal{P}$ is unlabeled and represented as a set $\mathcal{U} \sim \mathbb{P}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}})$. Given G with $\mathcal{P}$ and $\mathcal{U}$, our goal is to learn a binary classifier $f_w : (\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}}) \to [0, 1]$ that accurately approximates $\mathbb{P}(y = +1|\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}})$ on $\mathcal{U}$ with the trainable parameters w. Our formulation is fairly different from existing PU learning works that assume *i.i.d.* inputs.

2.2. Motivation

Class-Prior Estimation (CPE) with Heterophily Formally, CPE plays the role of estimating the class prior $\pi_{\mathrm{p}} \in (0, 1)$ for PU learning methods (Yao et al., 2020; Garg et al., 2021), given samples from the marginal distribution $\mathbb{P}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}})$ and samples from class-conditional distribution $\mathbb{P}_{\mathrm{p}}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}})$ of P nodes. Specifically, $\mathbb{P}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}})$ is mixture of class-conditional distributions of P and N nodes, *i.e*,

$$\mathbb{P}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}}) = \pi_{\mathrm{p}}\mathbb{P}_{\mathrm{p}}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}}) + (1 - \pi_{\mathrm{p}})\mathbb{P}_{\mathrm{n}}(\boldsymbol{x}, \mathcal{G}_{\boldsymbol{x}}^{\boldsymbol{A}}). \quad (1)$$

To ensure the identifiability of π_{p}, almost all CPE algorithms assume $\mathbb{P}_{\mathrm{n}}$ to be irreducible with respect to $\mathbb{P}_{\mathrm{p}}$ (Blanchard et al., 2010; Ivanov, 2020; Jain et al., 2016; Wu et al., 2023; Liu & Tao, 2015). However, this assumption is based on *independent and identically distributed* (i.i.d) data, which is not applicable to graph data. Here, we use the following theorem to specify the form of irreducible condition in our graph PU learning setting and leave its complete proof in Appendix A for reference.

Theorem 2.1 (Graph Irreducible Condition)**.** *In a graph, $\mathbb{P}_{\mathrm{n}}$ is irreducible with respect to $\mathbb{P}_{\mathrm{p}}$ iff it satisfies the following*

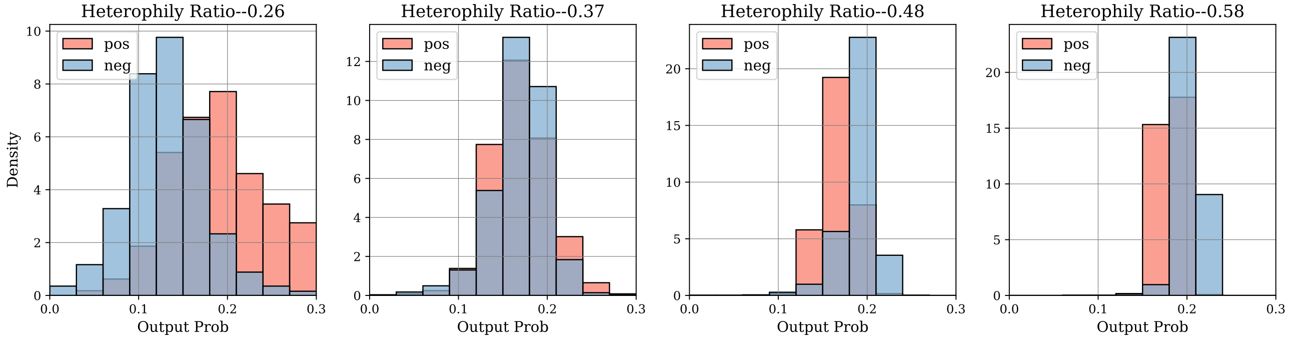

Figure 2: The distributions of the predicted probabilities (of being positive) for unlabeled nodes when training a classifier on positive and unlabeled CORA dataset, under varying heterophily ratios. The methodology employed to modify the graph structure with different heterophily ratios can be referred to (Ma et al., 2021). We show that as the heterophily ratio increases, the distribution of predicted probabilities of positives and negatives become less and less separable.

condition:

$$\operatorname*{ess\,sup}_{i\in\mathcal{V}} \mathbb{P}(y=+1|\boldsymbol{x}_i,\mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}) = 1, \tag{2}$$

where ess sup *is the essential supremum.*

With theorem 2.1, the irreducible condition on a graph essentially assumes that there exists one subset of nodes, along with their connected neighbouring nodes from $\mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}$, must all belong to the P class with probability one. However, the heterophilic structures in the graph suggest that P nodes are likely to have N nodes as neighbours, and vice versa. This characteristic violates the irreducible condition of the graph, yielding an overestimated class prior. As demonstrated in Figure 1, we can observe that the estimation error for the class prior escalates with the increase in heterophilic structure, which thus recalls extra design for graph PU learning.

PU classification with Heterophily The typical approach in PU classification is to distinguish P and N data from U data, followed by training a classifier on estimated P and N data (Garg et al., 2021; Xinrui et al., 2023). Ideally, with π_{p}, if we can distinguish all P and N data from the U data, then we can expect to achieve a well-performed classifier, akin to one trained using fully-supervised data. The core of graph PU classification hinges on the distinguishing performance from U nodes, which is complicated in the presence of heterophilic structures. As in Figure 2, with an increasing heterophilic ratio, the embeddings of P and N nodes from the GNN classifier become more indistinct, making the identification more challenging within the U nodes.

3. Methodology

Basically, a straightforward application of PU learning on graph data can be formulated as follows:

$$\begin{aligned} &\text{Stage 1:} \quad \hat{\pi}_{\mathrm{p}} = \mathcal{E}_{CPE}(\boldsymbol{A}), \\ &\text{Stage 2:} \quad w^* := \arg\min_{w} \mathcal{L}_{GNN}(\hat{\pi}_{\mathrm{p}}, w, \boldsymbol{A}). \end{aligned} \tag{3}$$

Here, w^* represents the optimal parameter for a GNN classifier. This parameter is derived by minimizing the loss using the pre-estimated class prior $\hat{\pi}_{\mathrm{p}}$, from an estimator $\mathcal{E}_{CPE}(\boldsymbol{A})$ with the adjacent matrix $\boldsymbol{A}$. As previously discussed, this process is intrinsically challenged by heterophilic structures characterized in the adjacency matrix $\boldsymbol{A}$. To address this, we propose the *Graph PU Learning with Label Propagation Loss* (GPL) method, which different from the two-stage form in Equation 3, has the following formulation:

$$\begin{aligned} &\min_{w} \mathcal{L}_{GNN}(\hat{\pi}_{\mathrm{p}}, w, \hat{\boldsymbol{A}}) \\ \text{s.t.} \quad &\hat{\pi}_{\mathrm{p}} = \mathcal{E}_{CPE}(\hat{\boldsymbol{A}}) \\ &\hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \mathcal{L}_{LPL}(\boldsymbol{A})^{(K)}, \end{aligned} \tag{4}$$

where the heterophilic structure $\boldsymbol{A}$ will be optimized to $\hat{\boldsymbol{A}}$ by the proposed *Label Propagation Loss* (LPL) in the next subsection. Intuitively, Equation 4 explores to introduce an intermediate heterophily reduction procedure to the vanilla graph PU learning, which helps reduce the negative impact in the aforementioned sections. In the following, we will discuss the detailed design and the optimization process.

3.1. Reducing Heterophily with Label Propagation Loss

Despite the straightforward idea in Equation 4, it is actually challenging in technique to directly reduce the heterophilic structures, since there are no annotated negative nodes available in PU setting. This difficulty hinders the application of previous heterophily reduction methods (Zhang et al., 2019; Huang et al., 2023). To address this issue, we incorporate the idea of *label propagation algorithm* (LPA)[1] that relies solely on observed positive nodes, characterized as follows.

Let $\boldsymbol{E}^{(0)} = [\mathbb{P}(y=+1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}), \mathbb{P}(y=-1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i})]_{i\in\mathcal{V}}$ be a vector of initial class-posterior probability for nodes in the graph and the updating rule in LPA is

$$\begin{aligned}
\boldsymbol{E}^{(k)} &= \alpha\boldsymbol{E}^{(k-1)} + (1-\alpha)\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{E}^{(k-1)},\\
\boldsymbol{E}^{(k)} &= [\mathbb{P}(y=+1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i})^{(k)}, \mathbb{P}(y=-1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i})^{(k)}]_{i\in\mathcal{V}}\\
&= [\boldsymbol{E}_i^{(k)}]_{i\in\mathcal{V}},
\end{aligned} \tag{5}$$

where $0 < \alpha < 1$ is a parameter, $k \in K$ is the label propagation iteration times and $\boldsymbol{D}$ is the associated diagonal degree matrix of $\boldsymbol{A}$. We construct $\boldsymbol{M} \odot \boldsymbol{A}$, where $\boldsymbol{M}$ is a learnable positive mask matrix that represents edge weights. Intuitively, in LPA, homophilic edges generally help positive nodes retain their positive labels, while heterophilic edges can impede these nodes from maintaining their original labels. As a result, the influence through heterophilic edges on a given P node $\boldsymbol{x}_a$ is inversely proportional to the probability of $\boldsymbol{x}_a$ being positively classified by LPA. This point will be theoretically proved in our Theorem 4.2.

With the above discussion, we can use LPA as a surrogate to reduce the heterophily, which we term as the *Label Propagation Loss* (LPL) as follows,

$$\begin{aligned}
\hat{\boldsymbol{A}} &= \arg\min_{\boldsymbol{A}} \mathcal{L}_{LPL}(\boldsymbol{A})^{(K)}\\
&= \arg\min_{\boldsymbol{A}} \frac{1}{n_{\mathcal{P}}} \sum_{\boldsymbol{x}_i\in\mathcal{P}} \log(\mathbb{P}(y=-1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i})^{(K)}).
\end{aligned} \tag{6}$$

The LPL technique is designed to minimize the likelihood of P nodes being misclassified as negative by the LPA, which effectively matches the given labels with the labels propagated through the graph structure. Furthermore, the incorporation of predicted labels from a trained model, as detailed in Equation 9, can further enhance this matching process.

3.2. Bilevel Training

With LPL, we can solve Equation 4 by bilevel optimization. In the following, we will detail the procedures for CPE and PU classification as well as a bilevel training schedule.

Class-Prior Estimator With multiple layers of graph convolution, assume a GNN model transforms each node to a positive posterior probability $\boldsymbol{z} \in [0,1]$, i.e., $\boldsymbol{z} = f_w(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$, where w denotes the trainable parameters of the GNN model f_w. For a probability density function $\mathbb{P}_\theta$ and f_w, a cumulative distribution function can be defined as $\mathbb{Q}^{\boldsymbol{A}}_\theta(\boldsymbol{z}) = \int_{f_w(\boldsymbol{x},\mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})\geq\boldsymbol{z}} \mathbb{P}_\theta(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})\mathrm{d}x$, which captures the probability that the input $(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$ is assigned a value larger than $\boldsymbol{z}$ by the classifier f_w in the transformed space. For each probability density distribution $\mathbb{P}_{\mathrm{p}}(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$, $\mathbb{P}_{\mathrm{n}}(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$ and $\mathbb{P}_{\mathrm{u}}(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$, we can define $\mathbb{Q}^{\boldsymbol{A}}_{\mathrm{p}}(\boldsymbol{z})$, $\mathbb{Q}^{\boldsymbol{A}}_{\mathrm{n}}(\boldsymbol{z})$ and $\mathbb{Q}^{\boldsymbol{A}}_{\mathrm{u}}(\boldsymbol{z})$ respectively. The class prior can be estimated with the optimized graph structure $\hat{\boldsymbol{A}}$ (after Equation 6):

$$\hat{\pi}_{\mathrm{p}} = \min_{\boldsymbol{c}\in[0,1]} \frac{\mathbb{Q}^{\hat{\boldsymbol{A}}}_{\mathrm{u}}(\boldsymbol{c})}{\mathbb{Q}^{\hat{\boldsymbol{A}}}_{\mathrm{p}}(\boldsymbol{c})} = \mathcal{E}_{CPE}(\hat{\boldsymbol{A}}). \tag{7}$$

Algorithm 1 Algorithm flow of GPL.

```
Input: Labeled positive training nodes $\mathcal{P}$ and unlabeled training nodes $\mathcal{U}$, Class prior estimate $\pi_{\mathrm{p}}$, Original adjacent matrix $\boldsymbol{A}$
Initialize a training GNN classifier $f_w$ with $\boldsymbol{A}$
$\hat{\boldsymbol{A}} \leftarrow \boldsymbol{A}$
for training iteration $t = 1, 2 \ldots E$ do
  while $k < K$ or not converge do
    $\hat{\boldsymbol{A}} = \arg\min_{\hat{\boldsymbol{A}}} \mathcal{L}_{LPL}(\hat{\boldsymbol{A}})^{(k)}$ (Equation 9)
    // i.e., optimizing $\hat{\boldsymbol{A}}$ to reduce heterophilic edges
  end while
  $\hat{\pi}_{\mathrm{p}} = \mathcal{E}_{CPE}(\hat{\boldsymbol{A}})$ (Equation 7)
  // i.e., estimating $\pi_{\mathrm{p}}$ with optimized $\hat{\boldsymbol{A}}$
  Minimizing $\mathcal{L}_{GNN}(\hat{\pi}_{\mathrm{p}}, w, \hat{\boldsymbol{A}})$ (Equation 8)
  // i.e., training $f_w$ with $\hat{\boldsymbol{A}}$ and $\hat{\pi}_{\mathrm{p}}$
end for
Output:Trained GNN classifier $f_w$
```

PU Classification Given a training set of P nodes $\mathcal{P}$ and U nodes $\mathcal{U}$ with the estimated class prior $\hat{\pi}_{\mathrm{p}}$, we begin by ranking $\mathcal{U}$ according to their positive class-posterior probability $\mathbb{P}(y_i=+1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}), \boldsymbol{x}_i \in \mathcal{U}$. Then, in every epoch of training, we define $\mathcal{U} \setminus \mathcal{S}_{\hat{\pi}_{\mathrm{p}}}$ as a (temporary) set of provisionally N nodes, created by excluding the subset $\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}$. This subset $\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}$ consists of U nodes representing the top $\hat{\pi}_{\mathrm{p}}$ fraction with the highest positive class-posterior probability. Next, we update our GNN classifier by minimising the loss with the $\hat{\boldsymbol{A}}$:

$$\begin{aligned}
\mathcal{L}_{GNN}(\hat{\pi}_{\mathrm{p}}, w, \hat{\boldsymbol{A}}) = &\frac{1}{|\mathcal{P}\cup\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}|} \sum_{\boldsymbol{x}_i\in\mathcal{P}\cup\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}} \mathcal{L}(f_w(\boldsymbol{x}_i, \mathcal{G}^{\hat{\boldsymbol{A}}}_{\boldsymbol{x}_i}), +1)\\
&+ \frac{1}{|\mathcal{U}\setminus\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}|} \sum_{\boldsymbol{x}_i\in\mathcal{U}\setminus\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}} \mathcal{L}(f_w(\boldsymbol{x}_i, \mathcal{G}^{\hat{\boldsymbol{A}}}_{\boldsymbol{x}_i}), -1).
\end{aligned} \tag{8}$$

Note that in PU classification, the identified P and N nodes are not only utilized to train a binary classifier but also contribute to our heterophily reduction. This dual functionality is possible because the predicted labels of these identified P and N nodes can be matched with the propagated labels,

[1] It is a classical algorithm that iteratively propagates estimated labels of nodes across edges within observed graph structures.

enabling the optimization of the graph structure. This approach parallels the use of original given labels, and thus we enhance Equation 6 with the identified P and N nodes:

$$\begin{aligned}\hat{\boldsymbol{A}} &= \arg\min_{\boldsymbol{A}} \mathcal{L}_{LPL}(\boldsymbol{A})^{(K)} \\ &= \arg\min_{\boldsymbol{A}} \Bigg(\frac{1}{n_{\mathcal{P}\cup\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}}} \sum_{\boldsymbol{x}_i \in \mathcal{P}\cup\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}} \log \mathbb{P}(y=-1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i})^{(K)} \\ &+ \frac{1}{n_{\mathcal{U}\backslash\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}}} \sum_{\boldsymbol{x}_i \in \mathcal{U}\backslash\mathcal{S}_{\hat{\pi}_{\mathrm{p}}}} \log \mathbb{P}(y=+1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i})^{(K)} \Bigg). \end{aligned} \tag{9}$$

Putting all things together, this entire procedure can be elegantly formulated as a bi-level optimization: In the inner loop, the identified P and N nodes (initially comprising only observed P nodes) are fixed while the graph structure is optimized to maximize label propagation within the same category. In the outer loop, while maintaining a constant optimized graph structure and utilizing CPE, a binary classifier is strategically trained and helps to differentiate N nodes from U nodes. We include the convergence of GPL in Appendix C for completeness, which is an application of standard bilevel optimization. Summary in Algorithm 1.

4. Theoretical Analysis

In this section, we delve into the theoretical properties of GPL. We mainly target to answer the basic question about Equation 4 in terms of $\mathcal{L}_{LPL}$, CPE and PU classification.

4.1. Why $\mathcal{L}_{LPL}$ reduces the heterophilic structure?

We analyze our LPL with the concept of *influence distribution* (Xu et al., 2018; Koh & Liang, 2017; Chen et al., 2022). Specifically, we investigate how the output probability of a node $\boldsymbol{x}_a$ changes when the initial probability of another node $\boldsymbol{x}_b$ is perturbed slightly through k iterations of label propagation. When $\boldsymbol{x}_a$ represents a positive node and $\boldsymbol{x}_b$ represents a negative node, we assess the influence of $\boldsymbol{x}_b$ on $\boldsymbol{x}_a$ using label propagation applied to the graph structure. This assessment allows us to quantify the extent of the heterophilic structure between $\boldsymbol{x}_b$ and $\boldsymbol{x}_a$, signifying the presence of edges connecting positive and negative nodes. We refer to this influence of one class of nodes on another class of nodes as *heterophily influence*(HI), which represents the existence of the heterophilic structure. According to (Koh & Liang, 2017; Xu et al., 2018; Wang & Leskovec, 2020), the influence can be measured by the gradient of the output probability of $\boldsymbol{x}_a$ with respect to the initial probability of $\boldsymbol{x}_b$. Thus, we have

Definition 4.1. (Heterophily influence with label propagation) The heterophily influence of a negative nodes $\boldsymbol{x}_b$ on positive nodes $\boldsymbol{x}_a$ after k iterations of label propagation is:

$$\begin{aligned} &HI((\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a}), (\boldsymbol{x}_b, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_b}); k) \\ &= \frac{\partial \left| \mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)} - \mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(0)} \right|}{\partial \mathbb{P}(y_b=-1|\boldsymbol{x}_b, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_b})}. \end{aligned} \tag{10}$$

According to the definition provided above, we can obtain the total heterophily influence on $\boldsymbol{x}_a$ as follows:

Theorem 4.2. *Considering a given positive node $(\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})$, the total heterophily influence of all other nodes on node $(\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})$ is proportional to the negatively class-posterior probability of $(\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})$ by label propagation:*

$$\begin{aligned} &\sum_{\boldsymbol{x}_i \in \mathcal{V}, \boldsymbol{x}_i \neq \boldsymbol{x}_a} HI((\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a}), (\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}); k) \\ &:= \left| \mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)} - \mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(0)} \right| \\ &\propto \mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)}, \end{aligned} \tag{11}$$

where $\mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)}$ is negative class-posterior probability of $(\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})$ by k-iteration label propagation.

We give a detailed proof of Theorem 4.2 in Appendix B. Generally, Theorem 4.2 indicates that, if the adjacency matrix $\boldsymbol{A}$ minimizes the $\mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)}$ through a k-iteration label propagation, they also minimize the heterophily influence on a given positive node. Consequently, we can make the adjacency matrix $\boldsymbol{A}$ trainable and learn to minimize the heterophily influence for each observed positive node by adjusting $\boldsymbol{A}$, which is equivalent to reducing the heterophilic structures within the graph.

4.2. Why reducing the heterophilic structure boosts CPE?

When estimating the CPE, we have

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}\in[0,1]} \frac{\mathbb{Q}_{\mathrm{u}}(\boldsymbol{c})}{\mathbb{Q}_{\mathrm{p}}(\boldsymbol{c})} \tag{12}$$

and $\hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \mathcal{L}_{LPL}(\boldsymbol{A})^{(K)}$. We define the estimation error for estimating class prior on the ordinary graph structure $\boldsymbol{A}$ as $e^{\boldsymbol{A}} = \left|\hat{\pi}^{\boldsymbol{A}}_{\mathrm{p}} - \pi^{\boldsymbol{A}}_{\mathrm{p}}\right|$ and the estimation error for estimating class prior on the optimized graph structure $\hat{\boldsymbol{A}}$ as $e^{\hat{\boldsymbol{A}}} = \left|\hat{\pi}^{\hat{\boldsymbol{A}}}_{\mathrm{p}} - \pi^{\hat{\boldsymbol{A}}}_{\mathrm{p}}\right|$. Then, we have the following lemma.

Lemma 4.3 (Theorem 1. of (Garg et al., 2021))**.** *For $\min(n_{\mathrm{p}}, n_{\mathrm{u}}) \geq \frac{2\log(4/\delta)}{\mathbb{Q}_{\mathrm{p}}(c^*)}$ and for every $\delta > 0$, the mixture proportion estimator $\hat{\pi}_p$ satisfies with probability $1-\delta$:*

$$|\hat{\pi}_{\mathrm{p}} - \pi_{\mathrm{p}}| \leq \frac{c}{\mathbb{Q}_{\mathrm{p}}(c^*)} \left(\sqrt{\frac{\log(4/\delta)}{n_{\mathrm{u}}}} + \sqrt{\frac{\log(4/\delta)}{n_{\mathrm{p}}}} \right) \tag{13}$$

for some constant $c \geq 0$.

Furthermore, according to the Lemma 4.3, we have the theorem on estimation error, which characterizes the benefits.

Theorem 4.4 (Estimation Error)**.** *Let the class prior can be estimated by* $\hat{\pi}_{\mathrm{p}} = \min_{\boldsymbol{c} \in [0,1]} \mathbb{Q}_{\mathrm{u}}^{\boldsymbol{A}}(\boldsymbol{c}) / \mathbb{Q}_{\mathrm{p}}^{\boldsymbol{A}}(\boldsymbol{c})$ *and the graph structure can be optimized through* $\hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \mathcal{L}_{LPA}(\boldsymbol{A})^{(K)}$*. Then, the upper bound of the estimation error with optimized graph structure* $\hat{\boldsymbol{A}}$ *is lower than ordinary graph structure* $\boldsymbol{A}$*.*

Please refer to Appendix D for the complete proof. According to Theorem 4.4, we can find that our heterophily minimization makes the upper bound of class prior estimation tighten, which proves the benefits of reducing heterophilic structures to the vanilla graph PU learning.

4.3. Why heterophilic structure degenerates PU Classification?

In this part, we perform a theoretical analysis of the heterophilic structure, elucidating how it leads P and N nodes to converge closer to each other within the embedding space. Let $\boldsymbol{h}^{(l)}$ be the embedding of node $\boldsymbol{x}^{(l)}$ after trained graph classifier. Through the following theorem, we illustrate that the aggregation step in graph classifier training, influenced by the heterophilic structure, reduces the distance in the embedding space between P and N nodes.

Theorem 4.5. *Let* $D_{\mathrm{PN}}(\boldsymbol{x}) = \frac{1}{2} \sum_{\boldsymbol{x}_i \in \mathrm{P}, \boldsymbol{x}_j \in \mathrm{N}} \tilde{\boldsymbol{A}}_{ij} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$ *be a distance metric between P and N node embeddings* $\boldsymbol{x}$ *and* $\tilde{\boldsymbol{A}}_{ij}(\boldsymbol{x}_i \in \mathrm{P}, \boldsymbol{x}_j \in \mathrm{N})$ *is the normalized weight of heterophilic edges. Then we have*

$$D_{\mathrm{PN}}(\boldsymbol{h}^{(l)}) \leq D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}).$$

Proof of Theorem 4.5 is in Appendix E. Theorem 4.5 indicates that following one aggregation step within the heterophilic structure, there is a reduction in the overall distance between P and N nodes. Thus, the heterophilic edges between P and N nodes hinder the distinguishability of these nodes, thereby making the task of identifying P and N nodes from U nodes more challenging during PU classification. Fortunately, the GPL method effectively minimizes heterophilic edge weights by utilizing LPL, which ensures that P and N nodes maintain a more discernible distinction than that they would be under the original structure.

5. Experiment

5.1. Experiment Setting

Datasets We evaluate the performance of our method on various real-world datasets, each characterized by an edge homophily ratio h ranging from strong homophily to strong heterophily, as defined in Zhu et al. (2020). We have summarized the dataset details in Table 1. To transform these datasets into binary classification tasks, we follow the previous approach (Yoo et al., 2021; Yang et al., 2023), where we treat the label with the largest number of nodes as positive and the rest as negative. The resulting numbers of positive and negative nodes are reported in the Appendix Table 4.

Dataset Processing To ensure a fair comparison with related work (Yoo et al., 2021; Wu et al., 2021; Yang et al., 2023), we randomly split each positive and negative dataset into positive and unlabeled sets. For each dataset, we use $r_{\mathrm{p}} = 50\%$ of all positive nodes as the observed positive nodes $\mathcal{P}$ and treat the rest as unobserved positive nodes $\mathcal{P}_u$. All negative nodes are treated as unobserved and denoted by $\mathcal{N}_u$, as we assume graph PU learning. Then, our objective is to predict the labels of each node in $\mathcal{U} = \mathcal{P}_u \cup \mathcal{N}_u$ as test data. This is achieved by training a classifier using the labeled nodes $\mathcal{P}$ and the unlabeled nodes $\mathcal{U}$; All nodes are accessible during training, but the labels of only $\mathcal{P}$ are observable as training data. Additionally, the true prior π_p, which is defined as $\pi_p = \mathcal{P}_u / (\mathcal{P}_u + \mathcal{N}_u)$ in our experiments.

Experimental Setup In the GPL, the backbone model is a graph convolutional network (GCN), with the number of layers set to 2 and the size of hidden layers set to 16. We train each model using Adam optimizer with a learning rate of 0.01. Owing to the imbalance between positive and negative nodes in our dataset, we adopt F1 score as our measure. For each experiment, we run five trials with different random seeds and compute the average and standard deviation.

5.2. Main Results

Classification Evaluation In this experiment, we benchmarked our method against several established approaches. This includes (1) Native GCN and MLP, which represent standard baselines for a range of graph structures, from homophily to heterophily; (2) GCN+TED, MLP+TED, GCN+NNPU, and MLP+NNPU: These are two popular methods adapted for PU learning in i.i.d data, *i.e*, TED (Garg et al., 2021) and NNPU (Kiryo et al., 2017) implemented with GCN and MLP as the backbone models. (3) LS-DAN (Ma & Zhang, 2017), GRAB (Yoo et al., 2021), and PU-GNN (Yang et al., 2023): These are three current methods tailored for graph PU learning. Further details about these baseline methods can be found in Appendix F.2.

The results presented in Table 1 show that our GPL method consistently outperforms others, achieving the highest classification performance across a variety of datasets, from homophilic to heterophilic datasets. Remarkably, our method shows particularly substantial improvements in datasets characterized by a high degree of heterophilic structure. Popular methods that excel in PU learning with i.i.d. data, such as TED and NNPU, do not demonstrate the same level of effectiveness on graph data, especially in the presence of pronounced heterophilic structures. This discrepancy

Table 1: Mean F1 score ± stdev over different datasets. The best model is highlighted in LightGreen. The heterophily (Hete.) ratios of all datasets are collected from (Zhu et al., 2020). The "*" methods mean using the knowledge of class prior.

	Cora	**Pubmed**	**Citeseer**	**Wiki-CS**	**Cornell**	**Chameleon**	**Squirrel**	**Actor**	**Wisconsin**	**Texas**
Hete. ratio h	**0.19**	**0.20**	**0.26**	**0.35**	**0.70**	**0.77**	**0.78**	**0.78**	**0.79**	**0.89**
#Nodes $\|\mathcal{V}\|$	2,708	19,717	3,327	11,701	183	2,277	5,201	7,600	251	183
#Edges $\|\mathcal{E}\|$	13,264	108,365	12,431	431,206	298	36,101	217,073	30,019	515	325
GCN	25.2±5.2	19.7±3.1	25.7±3.1	35.4±4.1	1.4±2.9	0.7±0.6	0.8±0.7	0.6±0.3	0.9±1.8	1.3±3.0
MLP	16.6±8.1	8.0±9.0	4.2±8.4	5.4±8.4	11.3±9.1	18.8±6.9	14.8±6.4	17.6±8.1	15.9±8.9	11.3±9.2
GCN+TED	80.1±0.8	75.4±0.4	70.0±1.4	80.6±0.5	13.9±5.5	17.5±3.5	22.8±1.3	27.6±1.0	19.6±6.1	11.9±5.0
MLP+TED	24.4±9.2	15.6±19.1	10.6±9.5	16.2±7.5	29.2±9.5	26.5±4.1	31.4±2.6	43.3±6.7	40.4±9.1	41.0±7.9
GCN+NNPU*	76.7±0.9	76.5±2.3	66.2±1.1	71.1±0.6	11.7±6.3	28.2±9.2	15.5±12.6	25.2±21.7	15.5±8.6	22.8±19.7
MLP+NNPU*	25.2±5.6	34.2±2.3	12.9±9.8	24.4±5.1	32.9±4.9	21.0±3.7	32.1±5.6	43.2±2.6	35.6±5.2	40.1±6.2
LSDAN*	63.5±4.1	69.6±0.4	47.0±19.	80.4±0.9	26.8±4.3	30.2±8.5	25.5±9.3	38.4±9.2	25.5±8.6	41.8±9.7
GRAB	80.4±0.2	71.6±0.3	69.7±0.4	79.4±1.0	30.6±9.0	15.5±2.2	29.0±8.4	25.2±0.7	39.5±9.6	40.2±8.2
PU-GNN*	79.8±0.3	73.0±0.4	69.5±0.4	80.3±1.8	28.3±3.9	31.5±6.3	29.5±4.8	32.6±8.4	27.8±8.2	42.3±9.4
GPL	**81.9±0.5**	**79.1±0.4**	**74.1±0.9**	**82.3±0.8**	**37.9±1.3**	**36.2±1.4**	**37.7±5.8**	**48.0±4.2**	**45.2±2.7**	**46.3±3.2**
	↑ 2.2%	↑ 3.4%	↑ 5.8%	↑ 2.1%	↑ 15.2%	↑ 14.5%	↑ 17.4%	↑ 10.9%	↑ 12.5%	↑ 10.2%

Table 2: Absolute estimation error with the true class prior in the first row over different datasets. Results were reported by meaning absolute error over five experiments. The best model with the smallest estimation error is highlighted.

	Cora	**Pubmed**	**Citeseer**	**Wiki-CS**	**Cornell**	**Chameleon**	**Squirrel**	**Actor**	**Wisconsin**	**Texas**
Hete. ratio h	**0.19**	**0.20**	**0.26**	**0.35**	**0.70**	**0.77**	**0.78**	**0.78**	**0.79**	**0.89**
#True π_{p}	0.1779	0.2496	0.1179	0.1293	0.2887	0.1294	0.1113	0.1485	0.3073	0.3835
KM	0.09±0.02	0.08±0.04	0.10±0.05	0.06±0.03	0.21±0.04	0.18±0.05	0.39±0.10	0.16±0.06	0.19±0.08	0.31±0.05
DEDPUL	0.05±0.03	0.05±0.05	0.07±0.04	0.05±0.02	0.19±0.05	0.16±0.04	0.32±0.09	0.14±0.07	0.17±0.06	0.25±0.07
MPE	0.04±0.03	0.04±0.02	0.08±0.04	0.02±0.01	0.13±0.06	0.10±0.03	0.36±0.17	0.08±0.02	0.14±0.06	0.26±0.03
ReMPE	0.03±0.03	0.04±0.01	0.06±0.03	0.02±0.02	0.14±0.07	0.12±0.05	0.34±0.12	0.11±0.04	0.12±0.05	0.25±0.03
BBE	0.03±0.01	0.05±0.01	0.04±0.02	0.03±0.01	0.27±0.01	0.12±0.01	0.10±0.08	0.11±0.01	0.23±0.05	0.27±0.02
TED	0.02±0.01	0.04±0.00	0.02±0.01	0.02±0.01	0.18±0.04	0.25±0.03	0.13±0.04	0.25±0.10	0.16±0.02	0.26±0.03
GRAB	0.07±0.13	0.08±0.09	0.06±0.08	0.09±0.12	0.35±0.26	0.18±0.07	0.36±0.19	0.30±0.07	0.37±0.18	0.33±0.20
GPL	**0.01±0.01**	**0.008±0.01**	**0.01±0.01**	**0.01±0.00**	**0.04±0.03**	**0.07±0.01**	**0.02±0.01**	**0.02±0.01**	**0.03±0.02**	**0.08±0.03**
	↓ 0.01	↓ 0.03	↓ 0.01	↓ 0.01	↓ 0.09	↓ 0.03	↓ 0.08	↓ 0.06	↓ 0.09	↓ 0.17

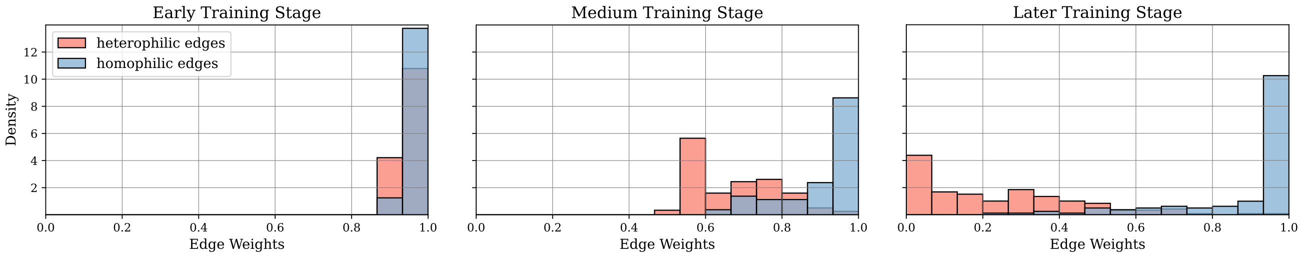

Figure 3: Distribution of edge weights for heterophilic and homophilic edges on Cornell dataset as training process in GPL.

underscores the unique challenges posed by heterophilic structures and proves the necessity and novelty of our GPL approach. Moreover, existing graph-based PU learning methods also show a marked decline in performance under heterophilic conditions, further confirming this point.

Class Prior Estimation In this section, we discuss the results of CPE. We compare our method with KM (Ramaswamy et al., 2016), DEDPUL (Ivanov, 2020), MPE (Scott, 2015), REMPE (Yao et al., 2020), BBE, TED (Garg et al., 2021) and GRAB (Yoo et al., 2021). The summarized results are shown in Table 2. Overall, using the same backbone model, our method consistently outperforms other baseline models. Notably, as the heterophily ratio increases, we observe a corresponding decrease in the estimation error. This trend clearly demonstrates that our method effectively reduces estimation errors by mitigating the impact of the heterophilic structure.

5.3. Ablation Study

Different Ratios of Labeled Nodes in PU learning. We summarize the performance of GPL and the baselines with different ratios of observed positive nodes to unlabeled ones in Figure 4. We gradually decrease the ratio r_{p} from 0.5 to 0.2. The problem becomes more difficult with smaller r_{p}.

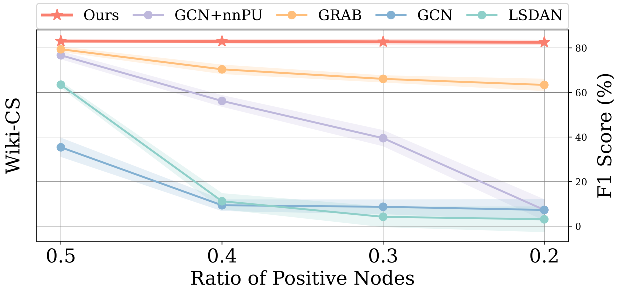

Figure 4: The F1 scores of GPL and the baseline approaches for PU learning. We change the ratio r_{p} of observed positive nodes among all positive ones.

Table 3: Ablation study for GPL.

Variant Hete. ratio h	Citeseer 0.26	Wiki-CS 0.35	Cornell 0.70	Chameleon 0.77
GPL	**74.1**$\pm$0.9	**82.3**$\pm$0.8	**37.9**$\pm$1.3	**36.2**$\pm$1.4
- w/o LPL	70.0 ↓ 5.5%	79.6 ↓ 3.3%	13.9 ↓ 63.3%	17.5 ↓ 51.6%
- w/o Bilevel	71.5 ↓ 3.5%	80.1 ↓ 2.7%	32.4 ↓ 14.5%	28.9 ↓ 20.2%
- w/o Selected	71.1 ↓ 5.3%	80.2 ↓ 3.7%	34.6 ↓ 8.7%	32.4 ↓ 10.5%

Despite this challenge, our method consistently maintains high F1 scores across all values of r_{p}, unlike the baselines whose performance deteriorates. This indicates the robustness of GPL even in imbalanced PU learning scenarios, where the count of observed nodes is significantly lower than the total number of nodes.

The distribution of Learned weights In Figure 3, we showcase the distribution of weights for both heterophilic and homophilic edges. During training with GPL, heterophilic edges tend to receive lower weights, whereas the weights of homophilic edges remain larger. This empirical observation demonstrates the substantial effectiveness of our GPL in mitigating the influence of heterophilic structures, thereby contributing to graph PU learning.

Contribution of Each Component We conducted a comprehensive analysis to evaluate the individual contributions of various components in our GPL method. This analysis in Table 3 includes (1) GPL without the LPL, focusing solely on reducing the heterophilic structure; (2) GPL without bilevel optimization, employing a two-stage approach of first optimizing the graph structure and then learning from PU nodes; (3) GPL without selected nodes, omitting the use of extracted nodes to enhance the convergence of the LPL. The results elucidate the distinct contributions of each component and underscore the importance of the LPL, especially in scenarios with a high heterophily ratio.

6. Related Work

Positive-Unlabeled Learning Positive-Unlabeled (PU) learning is broadly categorized into two subtasks: (i) Class-Prior Estimation (CPE) — determining the fraction of positive examples in the unlabeled data; and (ii) PU classification—given such an estimate, learning the desired positive-versus-negative classifier. Research on CPE and PU classification date to (De Comité et al., 1999; Denis, 1998; Letouzey et al., 2000). Without specific assumptions for CPE, class prior remains unidentifiable. To ensure identifiability, Du Plessis & Sugiyama (2014) and Elkan & Noto (2008) posited that positive and negative examples must have disjoint support. Following this, the concept of the irreducibility assumption was introduced by Blanchard et al. (2010), forming the basis for nearly all subsequent CPE algorithms (Blanchard et al., 2010; Ivanov, 2020; Jain et al., 2016). Recently, Yao et al. (2020) and Zhu et al. (2023) have begun exploring approaches to CPE that transcend the irreducibility assumption. However, these assumptions are based on i.i.d data, making them ill-suited for application to graph data. Our work is the first effort to adapt these assumptions for graph-based CPE.

Given the estimated class prior, the PU classification methods can be divided into two categories based on how unlabeled data is handled (Xinrui et al., 2023). The first category focuses on the sample-selection task to form a reliable negative set and further yield the semi-supervised learning framework, where the quantity of selected N data depends on the class prior (Liu et al., 2002; Li & Liu, 2003); the second regards unlabeled data as weighted positive and negative data simultaneously, as determined by the class-prior (Niu et al., 2016; Kiryo et al., 2017). Presently, methods specific to graph-based PU classification all fall into the latter category (Ma & Zhang, 2017; Yoo et al., 2021; Yang et al., 2023). However, these methods typically rely on the homophily characteristics of graphs to modulate the training loss for positive and unlabeled nodes, restricting their effectiveness in heterophilic structures, which are prevalent in graph data (Zhu et al., 2022; Li et al., 2022).

Graph Learning with Heterophily Heterophilic structures, prevalent in many graph data scenarios, have garnered considerable attention lately, leading to the emergence of various models designed to tackle this challenge. For instance, Geom-GCN (Pei et al., 2020) employs a bilevel aggregation process in the embedding space to manage heterophily. CPGNN (Zhu et al., 2021) addresses heterophilic signals by modelling label correlations using a compatibility matrix. Other methods (Luan et al., 2022; Chien et al., 2020; Bo et al., 2021) using the high-frequency graph signals in supervised embedding space to address heterophily. However, a common limitation of these methods is their dependency on complete label information to address heterophily. This dependency is not available in our PU learning setting, where negative labels are absent.

7. Conclusion

In this paper, we address the challenge of employing Positive and Unlabeled (PU) learning on graph-structured data, primarily due to the presence of heterophilic edges. We introduce the *G̲raph P̲U Learning with L̲abel P̲ropagation L̲oss* (GPL) method, specifically engineered to mitigate the adverse effects of heterophilic structures. This mitigation hinges on the newly proposed *Label Propagation Loss* (LPL), which effectively lowers the weights of heterophilic edges. Assisted by LPL, GPL refines the graph structure and develops a well-performing binary classifier for positive-unlabeled nodes. Our approach encapsulates this process in a bilevel optimization framework, where heterophilic reduction occurs in the inner loop, and classifier learning takes place in the outer loop. We have validated the efficacy of this framework through rigorous theoretical analysis and comprehensive experiments. Looking forward, we aim to extend our exploration to other forms of imperfect graph data, such as imbalanced or out-of-distribution graphs, further demonstrating the versatility of our approach.

Acknowledgements

TLL is partially supported by the following Australian Research Council projects: FT220100318, DP220102121, LP220100527, LP220200949, and IC190100031. JCY is supported by the National Key R&D Program of China (No. 2022ZD0160703), 111 plan (No. BP0719010) and National Natural Science Foundation of China (No. 62306178). BH is supported by the NSFC General Program No. 62376235 and Guangdong Basic and Applied Basic Research Foundation Nos. 2022A1515011652 and 2024A1515012399. The authors would give special thanks to Suqin Yuan and Muyang Li for helpful discussions and comments. The authors thank the reviewers and the meta-reviewer for their helpful and constructive comments on this work.

Impact Statement

In the era of big data, the prevalence of incomplete (positive-unlabeled) labels presents significant reliability challenges for traditional supervised learning algorithms. This issue is particularly acute in graph data, where it impedes the effective training of graph models. Learning models accurately from positive and unlabeled nodes is a crucial issue, drawing increasing attention in both research and industry circles due to its significant impact on practical applications involving graph data.

In this study, we propose the GPL method as a novel approach to learning from positive and unlabeled nodes. This method is specifically designed to counter the negative effects of heterophilic edges. The effectiveness of GPL is substantiated by the extensive evidence presented in our paper. The findings of this research contribute to a deeper understanding of how to handle incomplete labels in graph data. They mark a significant step forward in enhancing the robustness and accuracy of graph models, paving the way for more reliable and precise graph-based learning.

References

Bekker, J. and Davis, J. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109:719–760, 2020.

Bianchi, F. M., Grattarola, D., Livi, L., and Alippi, C. Graph neural networks with convolutional arma filters. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3496–3507, 2021.

Blanchard, G., Lee, G., and Scott, C. Semi-supervised novelty detection. *The Journal of Machine Learning Research*, 11:2973–3009, 2010.

Bo, D., Wang, X., Shi, C., and Shen, H. Beyond low-frequency information in graph convolutional networks. In *AAAI*, volume 35, pp. 3950–3957, 2021.

Chen, Z., Li, P., Liu, H., and Hong, P. Characterizing the influence of graph elements. *arXiv preprint arXiv:2210.07441*, 2022.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020.

De Comité, F., Denis, F., Gilleron, R., and Letouzey, F. Positive and unlabeled examples help learning. In *ALT*, pp. 219–230. Springer, 1999.

Denis, F. Pac learning from positive statistical queries. In *ALT*, pp. 112–126. Springer, 1998.

Denis, F., Gilleron, R., and Letouzey, F. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348(1):70–83, 2005.

Du Plessis, M. C. and Sugiyama, M. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.

Elkan, C. and Noto, K. Learning classifiers from only positive and unlabeled data. In *KDD*, pp. 213–220, 2008.

Garg, S., Wu, Y., Smola, A. J., Balakrishnan, S., and Lipton, Z. Mixture proportion estimation and pu learning: a modern approach. *NeurIPS*, 34:8532–8544, 2021.

Gasteiger, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Regep, C., Liu, G., Hayter, J. B., Vickers, R., Roberts, C., Tang, J., et al. Utilizing graph machine learning within drug discovery and development. *Briefings in bioinformatics*, 22(6):bbab159, 2021.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *NeurIPS*, 33: 22118–22133, 2020.

Huang, J., Li, P., Huang, R., Chen, N., and Zhang, A. Revisiting the role of heterophily in graph representation learning: An edge classification perspective. *ACM Transactions on Knowledge Discovery from Data*, 2023.

Ivanov, D. Dedpul: Difference-of-estimated-densities-based positive-unlabeled learning. In *ICMLA*, pp. 782–790. IEEE, 2020.

Jain, S., White, M., Trosset, M. W., and Radivojac, P. Nonparametric semi-supervised learning of class proportions. *arXiv preprint arXiv:1601.01944*, 2016.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kiryo, R., Niu, G., Du Plessis, M. C., and Sugiyama, M. Positive-unlabeled learning with non-negative risk estimator. *NeurIPS*, 30, 2017.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *ICML*, pp. 1885–1894. PMLR, 2017.

Letouzey, F., Denis, F., and Gilleron, R. Learning from positive and unlabeled examples. In *ALT*, pp. 71–85. Springer, 2000.

Li, M., Pan, S., Zhang, Y., and Cai, X. Classifying networked text data with positive and unlabeled examples. *Pattern Recognition Letters*, 77:1–7, 2016.

Li, X. and Liu, B. Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pp. 587–592. Citeseer, 2003.

Li, X., Zhu, R., Cheng, Y., Shan, C., Luo, S., Li, D., and Qian, W. Finding global homophily in graph neural networks when meeting heterophily. In *ICML*, pp. 13242–13256. PMLR, 2022.

Liu, B., Lee, W. S., Yu, P. S., and Li, X. Partially supervised classification of text documents. In *ICML*, volume 2, pp. 387–394. Sydney, NSW, 2002.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X.-W., and Precup, D. Revisiting heterophily for graph neural networks. *NeurIPS*, 35:1362–1375, 2022.

Ma, S. and Zhang, R. Pu-lp: A novel approach for positive and unlabeled learning by label propagation. In *ICMEW*, pp. 537–542. IEEE, 2017.

Ma, Y., Liu, X., Shah, N., and Tang, J. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021.

Niu, G., Du Plessis, M. C., Sakai, T., Ma, Y., and Sugiyama, M. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. *NeurIPS*, 29, 2016.

Panagopoulos, G., Nikolentzos, G., and Vazirgiannis, M. Transfer graph neural networks for pandemic forecasting. In *AAAI*, volume 35, pp. 4838–4845, 2021.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *ICML*, pp. 737–746. PMLR, 2016.

Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., and Yang, B. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Platonov, O., Kuznedelev, D., Babenko, A., and Prokhorenkova, L. Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. In *NeurIPS*, 2023.

Ramaswamy, H., Scott, C., and Tewari, A. Mixture proportion estimation via kernel embeddings of distributions. In *ICML*, pp. 2052–2060. PMLR, 2016.

Scott, C. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Artificial Intelligence and Statistics*, pp. 838–846. PMLR, 2015.

Scott, C., Blanchard, G., and Handy, G. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, pp. 489–511. PMLR, 2013.

Veličković, P. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology*, 79:102538, 2023.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wang, H. and Leskovec, J. Unifying graph convolutional neural networks and label propagation. *arXiv preprint arXiv:2002.06755*, 2020.

Wu, M., Pan, S., Du, L., and Zhu, X. Learning graph neural networks with positive and unlabeled nodes. *TKDD*, 15 (6):1–25, 2021.

Wu, Q., Zhang, H., Yan, J., and Wipf, D. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*, 2022.

Wu, Y., Xia, X., Yu, J., Han, B., Niu, G., Sugiyama, M., and Liu, T. Making binary classification from multiple unlabeled datasets almost free of supervision. *arXiv preprint arXiv:2306.07036*, 2023.

Wu, Y., Yao, J., Xia, X., Yu, J., Wang, R., Han, B., and Liu, T. Mitigating label noise on graph via topological sample selection. *arXiv preprint arXiv:2403.01942*, 2024.

Xia, X., Liu, J., Zhang, S., Wu, Q., and Liu, T. Coreset selection with prioritized multiple objectives. *arXiv preprint arXiv:2311.08675*, 2023.

Xinrui, W., Geng, C., Li, S.-Y., Chen, S., et al. Beyond myopia: Learning from positive and unlabeled data through holistic predictive trends. In *NeurIPS*, 2023.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *ICML*, pp. 5453–5462. PMLR, 2018.

Yan, Y., Hashemi, M., Swersky, K., Yang, Y., and Koutra, D. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *ICDM*, pp. 1287–1292. IEEE, 2022.

Yang, H., Zhang, Y., Yao, Q., and Kwok, J. Positive-unlabeled node classification with structure-aware graph learning. In *CIKM*, pp. 4390–4394, 2023.

Yao, Y., Liu, T., Han, B., Gong, M., Niu, G., Sugiyama, M., and Tao, D. Rethinking class-prior estimation for positive-unlabeled learning. *arXiv preprint arXiv:2002.03673*, 2020.

Yong, L., Pi, R., Zhang, W., Xia, X., Gao, J., Zhou, X., Liu, T., and Han, B. A holistic view of label noise transition matrix in deep learning and beyond. In *The Eleventh International Conference on Learning Representations*, 2022.

Yoo, J., Kim, J., Yoon, H., Kim, G., Jang, C., and Kang, U. Accurate graph-based pu learning without class prior. In *ICDM*, pp. 827–836. IEEE, 2021.

Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. Heterogeneous graph neural network. In *KDD*, pp. 793–803, 2019.

Zhao, X., Chen, F., Hu, S., and Cho, J.-H. Uncertainty aware semi-supervised learning on graph data. *NeurIPS*, 33:12827–12836, 2020.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. *NeurIPS*, 33: 7793–7804, 2020.

Zhu, J., Rossi, R. A., Rao, A., Mai, T., Lipka, N., Ahmed, N. K., and Koutra, D. Graph neural networks with heterophily. In *AAAI*, volume 35, pp. 11168–11176, 2021.

Zhu, J., Jin, J., Loveland, D., Schaub, M. T., and Koutra, D. How does heterophily impact the robustness of graph neural networks? theoretical connections and practical implications. In *KDD*, pp. 2637–2647, 2022.

Zhu, Y., Fjeldsted, A., Holland, D., Landon, G., Lintereur, A., and Scott, C. Mixture proportion estimation beyond irreducibility. *arXiv preprint arXiv:2306.01253*, 2023.

A. Proof of the Theorem 2.1

Up to now, almost all CPE algorithms assume $\mathbb{P}_{\mathrm{n}}$ to be irreducible with respect to $\mathbb{P}_{\mathrm{p}}$ (Blanchard et al., 2010; Ivanov, 2020; Jain et al., 2016), or stricter conditions like the anchor set assumption (Ramaswamy et al., 2016; Yao et al., 2020). In this section, we briefly review the irreducibility assumptions used for existing CPE estimators on i.i.d data. Then we understand the irreducibility assumption on the graph data when estimating the class prior on a graph.

The so-called irreducibility assumption was proposed by (Blanchard et al., 2010):

Definition A.1 (Irreducibility)**.** $\mathbb{P}_{\mathrm{n}}$ and $\mathbb{P}_{\mathrm{p}}$ are said to satisfy the irreducibility assumption if $\mathbb{P}_{\mathrm{n}}$ is not a mixture containing $\mathbb{P}_{\mathrm{p}}$. That is, there does not exist a decomposition $\mathbb{P}_{\mathrm{n}} = (1-\beta)Q + \beta\mathbb{P}_{\mathrm{p}}$, where Q is some probability distribution and $0 < \beta \leq 1$.

Equivalently, the irreducibility assumption assumes the support of $\mathbb{P}_{\mathrm{p}}$ is hardly contained in the support of $\mathbb{P}_{\mathrm{n}}$ and implies the following fact (Scott et al., 2013; Zhu et al., 2023):

Lemma A.2. *We say that $\mathbb{P}_{\mathrm{n}}$ is irreducible with respect to $\mathbb{P}_{\mathrm{p}}$ if the infimum of the likelihood ratio*

$$\operatorname*{ess\,inf}_{i\in\mathcal{V}} \frac{\mathbb{P}(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i} | y=-1)}{\mathbb{P}(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i} | y=+1)} = 0. \tag{14}$$

Now, we consider a way of understanding irreducibility in terms of a latent graph label model on a graph. The conditional probability of y given $(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$ to be defined via

$$\mathbb{P}(y=+1|\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}}) = \begin{cases} \frac{\pi_{\mathrm{p}}\mathbb{P}(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}}|y=+1)}{\mathbb{P}(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})}, & \mathbb{P}(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}}) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

This latent graph label model is inspired by the commonly used label model in the PU learning literature (Bekker & Davis, 2020). y may be viewed as a label indicating which component an observation from $\mathbb{P}(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$ was drawn from. Going forward, we use this latent graph label model in addition to the CPE.

Proposition A.3. *Under the latent label model, the essential supremum*

$$\begin{aligned} \operatorname*{ess\,sup}_{i\in\mathcal{V}} \mathbb{P}(y=+1|\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}) &= \frac{\pi_{\mathrm{p}}}{\operatorname{ess\,inf}_{i\in\mathcal{V}} \frac{\mathbb{P}(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i})}{\mathbb{P}(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}|y=+1)}} \\ &= \frac{\pi_{\mathrm{p}}}{\pi_{\mathrm{p}} + (1-\pi_{\mathrm{p}}) \operatorname{ess\,inf}_{i\in\mathcal{V}} \frac{\mathbb{P}(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}|y=-1)}{\mathbb{P}(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}|y=+1)}}. \end{aligned} \tag{16}$$

Combining Proposition A.3 and Lemma A.2, we conclude the irreducible condition for satisfying the irreducibility assumption on a graph as in the Theorem 2.1.

B. Proof of the Theorem 4.2

Inspired from the (Koh & Liang, 2017; Wang & Leskovec, 2020; Xu et al., 2018), we can propose

Lemma B.1. *Let $\mathcal{U}^{a\to b}_j$ be a path $[\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(j-1)}, \cdots, \boldsymbol{x}^{(0)}]$ of length j from positive node $(\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})$ to negative node $(\boldsymbol{x}_b, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_b})$, where $\boldsymbol{x}^{(j)} = \boldsymbol{x}_a$, $\boldsymbol{x}^{(0)} = \boldsymbol{x}_b$, $\boldsymbol{x}^{(i-1)} \in \mathcal{N}(\boldsymbol{x}^{(i)})$ for $i = j, \cdots, 1$, and all nodes along the path are unlabeled except $\boldsymbol{x}^{(0)}$. Then we have*

$$HI((\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a}), (\boldsymbol{x}_b, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_b}); k) = \sum_{j=1}^{k} \sum_{\mathcal{U}^{a\to b}_j} \prod_{i=j}^{1} \tilde{a}_{\boldsymbol{x}^{(i-1)}, \boldsymbol{x}^{(i)}}, \tag{17}$$

where $\tilde{a}_{\boldsymbol{x}^{(i-1)}, \boldsymbol{x}^{(i)}}$ is the normalized weight of edge $(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(i-1)})$.

Proof. When running LPA, the value of the negative class-posterior probability in $y_a^{(\cdot)}$ (denoted by $\mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(\cdot)}$) mainly comes from the nodes with initial negative label. It is clear that

$$\left|\mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)} - \mathbb{P}(y_a=-1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(0)}\right| := \sum_{\boldsymbol{x}_b: y_b=-1} \sum_{j=1}^{k} \sum_{\mathcal{U}^{a\to b}_j} \prod_{i=j}^{1} \tilde{a}_{\boldsymbol{x}^{(i-1)}, \boldsymbol{x}^{(i)}}, \tag{18}$$

which equals $\sum_{\boldsymbol{x}_b:y_b=-1} HI((\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a}), (\boldsymbol{x}_b, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_b}); k)$ according to Lemma B.1. Therefore, we have

$$\left|\mathbb{P}(y_a = -1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)} - \mathbb{P}(y_a = -1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(0)}\right| := \sum_{\boldsymbol{x}_b:y_b=-1} HI((\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a}), (\boldsymbol{x}_b, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_b}); k), \tag{19}$$

which is the same as

$$\left|\mathbb{P}(y_a = -1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(k)} - \mathbb{P}(y_a = -1|\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a})^{(0)}\right| := \sum_{\boldsymbol{x}_i \in \mathcal{V}, \boldsymbol{x}_i \neq \boldsymbol{x}_a} HI((\boldsymbol{x}_a, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_a}), (\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}); k) \tag{20}$$

□

C. Convergence of GPL

The convergence of the bilevel optimization (Xia et al., 2023) using approximated solution of inner loop was established in (Pedregosa, 2016; Yong et al., 2022). We restate it here for completeness.

Theorem C.1 (Convergence, Theorem 3.3 of (Pedregosa, 2016))**.** *Suppose $\mathcal{L}_{GNN}(\hat{\pi}_{\mathrm{p}}, w, \hat{\boldsymbol{A}})$ is smooth w.r.t. w, $\mathcal{L}_{LPA}(\boldsymbol{A})^{(k)}$ is β-smooth and α-strongly convex w.r.t. $\boldsymbol{A}$. We solve the inner loop by unrolling J steps, choose the learning rate in the outer loop as $\eta_\tau = 1/\sqrt{\tau}$, then we arrive at an approximately stationary point as follows after R steps:*

$$\mathbb{E}\left[\sum_{\tau=1}^{R} \frac{\eta_\tau \left\|\nabla_w \mathcal{L}_{GNN}(\hat{\pi}_{\mathrm{p}}, w, \hat{\boldsymbol{A}})\right\|_2^2}{\sum_{\tau=1}^{R} \eta_\tau}\right] \leq \tilde{O}\left(\epsilon + \frac{\epsilon^2 + 1}{\sqrt{R}}\right), \tag{21}$$

where $\tilde{O}$ absorbs constants and logarithmic terms and $\epsilon = (1 - \alpha/\beta)^J$.

D. Proof of the Theorem 4.4

According to the Lemma 4.3, the upper bound of $e^{\boldsymbol{A}}$ is

$$b^{\boldsymbol{A}} = \frac{c}{\mathbb{Q}^{\boldsymbol{A}}_{\mathrm{p}}(c^*)}\left(\sqrt{\frac{\log(4/\delta)}{n_{\mathrm{u}}}} + \sqrt{\frac{\log(4/\delta)}{n_{\mathrm{p}}}}\right) \tag{22}$$

and the upper bound of $e^{\hat{\boldsymbol{A}}}$ is

$$b^{\hat{\boldsymbol{A}}} = \frac{c}{\mathbb{Q}^{\hat{\boldsymbol{A}}}_{\mathrm{p}}(c^*)}\left(\sqrt{\frac{\log(4/\delta)}{n_{\mathrm{u}}}} + \sqrt{\frac{\log(4/\delta)}{n_{\mathrm{p}}}}\right). \tag{23}$$

The $\mathbb{Q}^{\boldsymbol{A}}_{\mathrm{p}}(c^*) := \sum_{\boldsymbol{x}_i \in \mathcal{P}} \mathbb{1}[f_w(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}) \geq c^*]/n$ and $\mathbb{Q}^{\hat{\boldsymbol{A}}}_{\mathrm{p}}(c^*) := \sum_{\boldsymbol{x}_i \in \mathcal{P}} \mathbb{1}[f_w(\boldsymbol{x}_i, \mathcal{G}^{\hat{\boldsymbol{A}}}_{\boldsymbol{x}_i}) \geq c^*]/n$, we know that the GNN model transforms each node $(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$ to a positive posterior probability $\boldsymbol{z} \in [0, 1]$, i.e., $\boldsymbol{z} = f_w(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$. Numerous studies (Yan et al., 2022; Zhu et al., 2020; 2021) have both theoretically and empirically demonstrated that heterophilic structures impede the confidence of GNN predictions. Therefore, for positive nodes, the positive posterior probability $\hat{\boldsymbol{z}} = f_w(\boldsymbol{x}, \mathcal{G}^{\hat{\boldsymbol{A}}}_{\boldsymbol{x}})$ is greater than $\boldsymbol{z} = f_w(\boldsymbol{x}, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}})$ due to the reduction of heterophilic structure.

Thus, we have

$$\sum_{\boldsymbol{x}_i \in \mathcal{P}} \mathbb{1}[f_w(\boldsymbol{x}_i, \mathcal{G}^{\hat{\boldsymbol{A}}}_{\boldsymbol{x}_i}) \geq c^*]/n \geq \sum_{\boldsymbol{x}_i \in \mathcal{P}} \mathbb{1}[f_w(\boldsymbol{x}_i, \mathcal{G}^{\boldsymbol{A}}_{\boldsymbol{x}_i}) \geq c^*]/n, \tag{24}$$

then

$$\mathbb{Q}^{\hat{\boldsymbol{A}}}_{\mathrm{p}}(c^*) \geq \mathbb{Q}^{\boldsymbol{A}}_{\mathrm{p}}(c^*). \tag{25}$$

According to Equation 22 and 23, we have following inequality:

$$b^{\hat{\boldsymbol{A}}} \leq b^{\boldsymbol{A}}. \tag{26}$$

E. Proof of Theorem 4.5

Firstly, we analyze the layer-wise propagation rules of GNN for the node $\boldsymbol{x}_i$:

$$\boldsymbol{x}_i^{(l+1)} = \sigma\left(\sum_{\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)} \tilde{\boldsymbol{A}}_{ij} \boldsymbol{x}_i^{(l)} w^{(l)}\right),$$

where $\tilde{\boldsymbol{A}}_{ij} = \boldsymbol{A}_{ij}/d_{ii}$ is the normalized weight of edge e_{ij} with the degree d_{ii}. This formula can be decomposed into the following two steps: (1) In *aggregation* step, we calculate the aggregated representation $\boldsymbol{h}_i^{(l)}$ of all neighborhoods $\mathcal{N}(\boldsymbol{x}_i)$:

$$\boldsymbol{h}_i^{(l)} = \sum_{\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)} \tilde{\boldsymbol{A}}_{ij} \boldsymbol{x}_j^{(l)}.$$

(2) In *transformation* step, the aggregated representation $\boldsymbol{h}_i^{(l)}$ is mapped to a new space by a transformation matrix and nonlinear function σ:

$$\boldsymbol{x}_i^{(l+1)} = \sigma\big(\boldsymbol{h}_i^{(l)} w^{(l)}\big).$$

we assume that the dimension of node representations is one, but note that the conclusion can be easily generalized to the case of multi-dimensional representations since the function $D_{\mathrm{PN}}(\boldsymbol{x})$ can be decomposed into the sum of one-dimensional cases. This proof is according to (Wang & Leskovec, 2020). In the following of this proof, we still use bold notations $\boldsymbol{x}_i^{(l)}$ and $\boldsymbol{h}_i^{(l)}$ to denote node representations, but keep in mind that they are scalars rather than vectors.

We give two lemmas before proving Theorem 4.5. The first one is about the gradient of $D_{\mathrm{PN}}(\boldsymbol{x})$:

Lemma E.1. $\boldsymbol{h}_i^{(l)} = \boldsymbol{x}_i^{(l)} - \frac{\partial D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})}{\partial \boldsymbol{x}_i^{(l)}}$.

Proof.

$$\boldsymbol{x}_i^{(l)} - \frac{\partial D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})}{\partial \boldsymbol{x}_i^{(l)}} = \boldsymbol{x}_i^{(l)} - \sum_{\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)} \tilde{\boldsymbol{A}}_{ij}(\boldsymbol{x}_i^{(l)} - \boldsymbol{x}_j^{(l)}) = \sum_{\boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i)} \tilde{\boldsymbol{A}}_{ij} \boldsymbol{x}_j^{(l)} = \boldsymbol{h}_i^{(l)} \tag{27}$$

□

It is interesting to see from Lemma E.1 that the aggregation step in GCN is equivalent to running gradient descent for one step with a step size of one. However, this is not able to guarantee that $D_{\mathrm{PN}}(\boldsymbol{h}^{(l)}) \le D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})$ because the step size may be too large to reduce the value of D_{PN}.

The second lemma is about the Hessian of $D_{\mathrm{PN}}(\boldsymbol{x})$:

Lemma E.2. $\nabla^2 D_{\mathrm{PN}}(\boldsymbol{x}) \preceq 2I$*, or equivalently,* $2I - \nabla^2 D_{\mathrm{PN}}(\boldsymbol{x})$ *is a positive semidefinite matrix.*

Proof. We first calculate the Hessian of $D_{\mathrm{PN}}(\boldsymbol{x}) = \frac{1}{2}\sum_{\boldsymbol{x}_i \in \mathrm{P}, \boldsymbol{x}_j \in \mathrm{N}} \tilde{a}_{ij} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$:

$$\nabla^2 D_{\mathrm{PN}}(\boldsymbol{x}) = \begin{bmatrix} 1-\tilde{a}_{11} & -\tilde{a}_{12} & \cdots & -\tilde{a}_{1n} \\ -\tilde{a}_{21} & 1-\tilde{a}_{22} & \cdots & -\tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\tilde{a}_{n1} & -\tilde{a}_{n2} & \cdots & 1-\tilde{a}_{nn} \end{bmatrix} = I - D_{\mathrm{PN}}^{-1} A. \tag{28}$$

Therefore, $2I - \nabla^2 D_{\mathrm{PN}}(\boldsymbol{x}) = I + D_{\mathrm{PN}}^{-1}\boldsymbol{A}$. Since $D_{\mathrm{PN}}^{-1}\boldsymbol{A}$ is Markov matrix (i.e., each entry is non-negative and the sum of each row is one), its eigenvalues are within the range [-1, 1], so the eigenvalues of $I + D_{\mathrm{PN}}^{-1}\boldsymbol{A}$ are within the range [0, 2]. Therefore, $I + D_{\mathrm{PN}}^{-1}\boldsymbol{A}$ is a positive semidefinite matrix, and we have $\nabla^2 D_{\mathrm{PN}}(\boldsymbol{x}) \preceq 2I$. □

We can now prove Theorem 4.5:

Proof. Since D_{PN} is a quadratic function, we perform a second-order Taylor expansion of D_{PN} around $\boldsymbol{x}^{(l)}$ and obtain the following inequality:

$$\begin{aligned} D_{\mathrm{PN}}(\boldsymbol{h}^{(l)}) =& D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}) + \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})^\top(\boldsymbol{h}^{(l)} - \boldsymbol{x}^{(l)}) + \frac{1}{2}(\boldsymbol{h}^{(l)} - \boldsymbol{x}^{(l)})^\top \nabla^2 D_{\mathrm{PN}}(\boldsymbol{x})(\boldsymbol{h}^{(l)} - \boldsymbol{x}^{(l)}) \\ =& D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}) - \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})^\top \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}) + \frac{1}{2}\nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})^\top \nabla^2 D_{\mathrm{PN}}(\boldsymbol{x}) \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}) \\ \leq& D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}) - \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})^\top \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}) + \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)})^\top \nabla D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}) = D_{\mathrm{PN}}(\boldsymbol{x}^{(l)}). \end{aligned} \tag{29}$$

□

F. Experiment

F.1. Datasets

The statistical information of used datasets is shown in Table 4.

Table 4: Summary of datasets.

Name	Nodes	Edges	Features	Pos.	Neg.
Cora	2,708	5,278	1,433	818	1,890
CiteSeer	3,327	4,552	3,703	701	2,626
PubMed	19,717	44,324	500	7,875	11,842
WikiCS	11,701	215,603	300	2,679	9,022
Cornell	183	298	1,703	41	142
Chameleon	2,277	36,101	2,325	260	2,017
Squirrel	5,201	2,089	217,073	521	4,680
Actor	7,600	30,019	932	982	6,618
Wisconsin	251	515	1,703	59	192
Texas	183	325	1,703	50	133

F.2. Baseline Details

In more detail, we employ baselines:

(1) GCN: Utilizing the cross-entropy loss function to train a Graph Convolutional Network (GCN) and treating all unlabeled nodes as negative examples.

(2) MLP: Applying the cross-entropy loss during the training of a Multi-Layer Perceptron (MLP), where unlabeled nodes are treated as negative examples.

(3) GCN+TED: Combining the GCN model with the TED method (Garg et al., 2021), a state-of-the-art approach specifically tailored for PU learning, which estimates the unknown class prior.

(4) MLP+TED: Integrating the MLP model with the TED method for PU learning.

(5) GCN+NNPU: Adopting a GCN model with the Non-Negative PU (NNPU) method (Kiryo et al., 2017), which employs a non-negative risk estimator specially designed for PU learning to estimate the unknown negative risk.

(6) MLP+NNPU: Using an MLP model with the NNPU method.

(7) LSDAN (Ma & Zhang, 2017): Improving GCN on PU learning by the long-short distance attention that effectively combines the information of multi-hop neighbors.

(8) GRAB (Yoo et al., 2021): Estimating class priors using heuristics and learning from positively labeled nodes in the presence of unlabeled nodes, leveraging the homophily graph structure.

(9) PU-GNN (Yang et al., 2023): Introducing a distance-aware loss that leverages homophily within graphs to provide more precise supervision in the context of graph-based Positive-Unlabeled (PU) learning.

F.3. Performance with different GNN architectures

We evaluate our proposed GPL on different GNN architectures, i.e., GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), ARMA (Bianchi et al., 2021) and APPNP (Gasteiger et al., 2018). The experiments are conducted on various datasets, which are shown in Table 5. As can be seen, GPL performs similarly on different GNN architectures, showing consistent generalization on different architectures.

Table 5: Mean F1 score ± stdev over different datasets. The experimental results are reported over five trials. Bold numbers are superior results.

Backbone	Cora	Pubmed	Citeseer	Wiki-CS	Cornell	Chameleon	Squirrel	Actor	Wisconsin	Texas
GPL+GCN	81.9±0.5	79.1±0.4	74.1±0.9	**82.3±0.8**	37.9±1.3	36.2±1.4	37.7±5.8	48.0±4.2	**45.8±2.7**	46.3±3.2
GPL+GAT	**82.1±0.5**	**79.8±0.7**	74.7±0.2	81.9±0.3	36.3±1.1	35.7±0.9	**37.9±4.6**	48.8±3.5	44.8±1.8	**46.6±2.1**
GPL+ARMA	81.2±1.1	78.9±0.9	74.1±0.5	81.7±0.7	**38.0±0.8**	35.9±1.2	36.9±4.5	47.0±2.2	44.9±2.3	46.5±2.9
GPL+APPNP	81.5±0.6	79.3±0.5	**74.9±0.7**	82.1±0.6	37.7±1.8	**36.8±1.1**	37.1±4.9	**49.5±3.8**	45.1±2.9	45.9±2.6

F.4. Hyperparameter sensitivity

In GPL, the hyperparameter K affects the performance by controlling the iteration times in the label propagation loss. To assess the sensitivity of GPL to K, we analyzed its effect as depicted in Figure 5. Our findings reveal an initial increase in the F1 score with a rising K, indicative of effective label propagation loss (LPL) functioning. However, a significant drop in the F1 score is observed when K increases excessively, suggesting that overly high iteration counts hinder the convergence of LPL and performance.

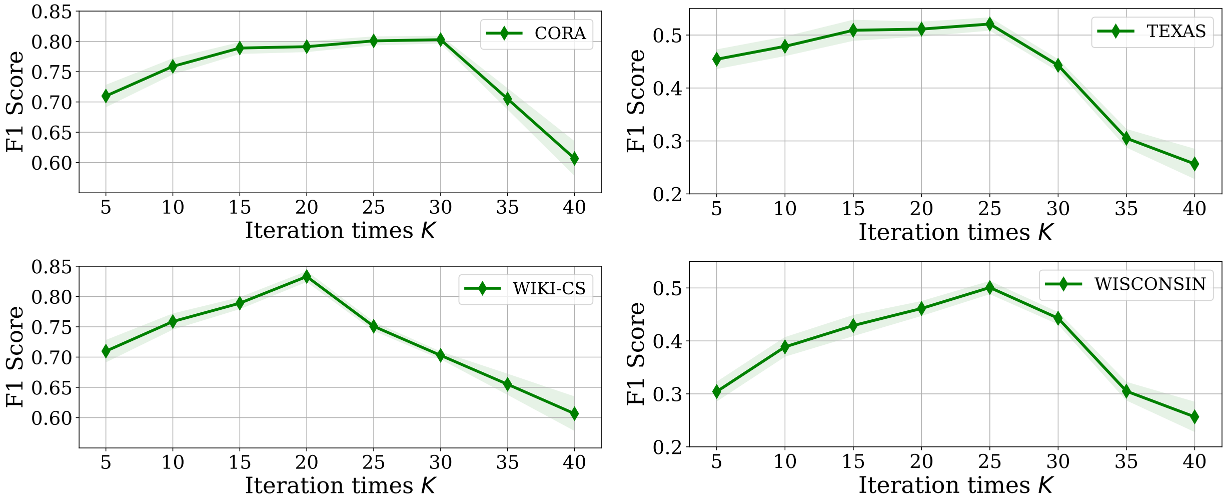

Figure 5: The F1 score of GPL with increasing K on *Cora*, *WIKI-CS*, *Texas* and *Wisconsin*.