Position-aware Structure Learning for Graph Topology-imbalance by Relieving Under-reaching and Over-squashing

Qingyun Sun  
Beihang University  
Beijing, China  
sunqy@buaa.edu.cn

Jianxin Li  
Beihang University  
Beijing, China  
ljx@buaa.edu.cn

Haonan Yuan  
Beihang University  
Beijing, China  
yuanhn@act.buaa.edu.cn

Xingcheng Fu  
Beihang University  
Beijing, China  
fuxc@act.buaa.edu.cn

Hao Peng  
Beihang University  
Beijing, China  
penghao@act.buaa.edu.cn

Cheng Ji  
Beihang University  
Beijing, China  
jicheng@act.buaa.edu.cn

Qian Li  
Beihang University  
Beijing, China  
lqian@act.buaa.edu.cn

Philip S. Yu  
University of Illinois at Chicago  
Chicago, USA  
psyu@uic.edu

ABSTRACT

Topology-imbalance is a graph-specific imbalance problem caused by the uneven topology positions of labeled nodes, which significantly damages the performance of GNNs. What topology-imbalance means and how to measure its impact on graph learning remain under-explored. In this paper, we provide a new understanding of topology-imbalance from a global view of the supervision information distribution in terms of under-reaching and over-squashing, which motivates two quantitative metrics as measurements. In light of our analysis, we propose a novel position-aware graph structure learning framework named PASTEL, which directly optimizes the information propagation path and solves the topology-imbalance issue in essence. Our key insight is to enhance the connectivity of nodes within the same class for more supervision information, thereby relieving the under-reaching and over-squashing phenomena. Specifically, we design an anchor-based position encoding mechanism, which better incorporates relative topology position and enhances the intra-class inductive bias by maximizing the label influence. We further propose a class-wise conflict measure as the edge weights, which benefits the separation of different node classes. Extensive experiments demonstrate the superior potential and adaptability of PASTEL in enhancing GNNs’ power in different data annotation scenarios.

CCS CONCEPTS

• Computing methodologies → Neural networks; Learning latent representations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Qingyun Sun, Jianxin Li, Haonan Yuan, Xingcheng Fu, Hao Peng, Cheng Ji, Qian Li, and Philip S. Yu. 2022. Position-aware Structure Learning for Graph Topology-imbalance by Relieving Under-reaching and Over-squashing. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22), October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557419

1 INTRODUCTION

Graph learning [13, 24, 52] has gained popularity over the past years due to its versatility and success in representing graph data across a wide range of domains [9, 14, 26, 40, 50]. Graph Neural Networks (GNNs) [39, 47] have been the “battle horse” of graph learning, which propagate the features on the graph by exchanging information between neighbors in a message-passing paradigm [15]. Due to the asymmetric and uneven topology, learning on graphs by GNNs suffers a specific imbalance problem, i.e., topology-imbalance. Topology-imbalance [7] is caused by the uneven position distribution of labeled nodes in the topology space, which is inevitable in real-world applications due to data availability and the labeling costs. For example, we may only have information for a small group of users within a local community in social networks, resulting in a serious imbalance of labeled node positions. The uneven position distribution of labeled nodes leads to uneven information propagation, resulting in the poor quality of learned representations.

Although the imbalance learning on graphs has attracted many research interests in recent years, most of them focus on the class-imbalance issue [30, 46], i.e., the imbalanced number of labeled nodes of each class. The topology-imbalance issue is proposed recently and is still under-explored. The only existing work, ReNode [7], provides an understanding of the topology-imbalance issue from the perspective of label propagation and proposes a sample re-weighting method. However, ReNode takes the node topological
boundaries as decision boundaries based on a homophily assumption, which does not work with real-world graphs. The strong assumption leads to poor generalization and unsatisfied performance of ReNode (see Section 5.2.1). There are two remaining questions:
(1) Why does topology-imbalance affect the performance of graph representation learning? and (2) What kind of graphs are susceptible to topology-imbalance? To answer the above two questions, how to measure the influence of labeled nodes is the key challenge in handling topology-imbalance due to the complex graph connections and the unknown class labels for most nodes in the graph.

**New understanding for topology-imbalance.** In this work, we provide a new understanding of the topology-imbalance issue from a global view of the supervision information distribution in terms of under-reaching and over-squashing: (1) **Under-reaching:** the influence of labeled nodes decays with the topology distance [3], resulting in the nodes far away from labeled nodes lack of supervision information. In Figure 1, the node \(v_a\) cannot reach the valuable labeled node \(v_c\) within the receptive field of the GNN model, resulting in the quality of information it received is limited. (2) **Over-squashing:** the supervision information of valuable labeled nodes is squashed when passing across the narrow path together with other useless information. In Figure 1, the valuable supervision information of \(v_b\) to \(v_c\) is compressed into a vector together with the information of many nodes belonging to other classes, resulting in the quality of supervision information that \(v_a\) received being poor. Then we introduce two metrics (reaching coefficient and squashing coefficient) to give a quantitative analysis of the relation between the learning performance, label positions, and graph structure properties. We further draw a conclusion that better reachability and lower squashing to labeled nodes lead to better classification performance for GNN models.

**Present work.** In light of the above analysis, we propose a Position-Aware STructurE Learning method named PASTEL, which directly optimizes the information propagation path and solves the problem of topology-imbalance issue in essence. The key insight of PASTEL is to enable nodes within the same class to connect more closely with each other for more supervision information. Specifically, we design a novel anchor-based position encoding mechanism to capture the relative position between nodes and incorporate the position information into structure learning.

Then we design a class-wise conflict measure based on the Group PageRank, which measures the influence from labeled nodes of each class and acts as a guide to increase the intra-class connectivity via adjusting edge weight. The main contributions are as follows:

- We provide a new understanding of the topology-imbalance issue from the perspective of supervision information distribution in terms of under-reaching and over-squashing and provide two new quantitative metrics for them.
- Equipped with the proposed position encodings and class-wise conflict measure, PASTEL can better model the relationships of node pairs and enhance the intra-class inductive bias by maximizing the label influence.
- Experimental results demonstrate that the proposed PASTEL enjoys superior effectiveness and indeed enhances the GNN model’s power for in-the-wild extrapolation.

2 RELATED WORK

2.1 Imbalance Learning

Imbalanced classification problems [18, 41] have attracted extensive research attention. Most existing works [16, 25] focus on the class-imbalance problem, where the model performance is dominated by the majority class. The class-imbalance learning methods can be roughly divided into two types: data-level re-sampling and algorithm-level re-weighting. **Re-sampling** methods re-sample [2, 5, 48] or augment data [30] to balance the number of data for each class during the data selection phase. **Re-weighting** methods [4, 11, 32] adjust different weights to different data samples according to the number of data during the training phase.

For the graph-specific topology-imbalance issue as mentioned in Section 1, directly applying these methods to the graph data fails to take the special topology properties into consideration. ReNode [7] is the first work for the graph topology-imbalance issue, which follows the paradigm of classical re-weighting methods. Specifically, ReNode defines an influence conflict detection based metric and re-weights the labeled nodes based on their relative positions to class boundaries. However, ReNode is limited by its homophily assumption and only has a slight performance improvement. **In this paper, PASTEL alleviates topology-imbalance by learning a new structure that maximizes the intra-class label influence, which can be seen as "label re-distribution" in the topology space.**

2.2 Graph Structure Learning

Graph structure learning [55] learns an optimized graph structure for representation learning and most of them aim to improve the robustness [20, 54] of GNN models. There are also some works [8, 10, 12, 38, 42] that utilize the structure learning to improve the graph representation quality. As for the over-squashing problem, [45] assigns different weights to edges connected to two nodes of the same class for better representations. However, [45] still fails with the issue of under-reaching. SDRF [42] rewires edges according to the Ricci curvatures to solve the over-squashing problem by only considering topology properties.

Multiple measurements in existing structure learning works are leveraged for modeling node relations, including node features [53], node degrees [20], node encodings [51] and edge attributes [54].
The node positions play an important role in generating discriminative representations [49] and are seldom considered in structure learning. In this work, we advance the structure learning strategy for the graph topology-imbalance issue and introduce a position-aware framework to better capture the nodes’ underlying relations.

3 UNDERSTANDING TOPOLOGY-IMBALANCE

In this section, we provide a new understanding of the topology-imbalance issue in terms of under-reaching and over-squashing. Then we perform a quantitative analysis of the relations between them to answer two questions:

Q1: Why does topology-imbalance affect the performance of graph representation learning?
Q2: What kind of graphs are susceptible to topology-imbalance?

3.1 Notations and Preliminaries

Consider a graph $G = (V, E)$, where $V$ is the set of $N$ nodes and $E$ is the edge set. Let $A \in \mathbb{R}^{N \times N}$ be the adjacency matrix and $X \in \mathbb{R}^{D \times N}$ be the node attribute matrix, where $D_0$ denotes the dimension of node attributes. The diagonal degree matrix is denoted as $D = \sum_{i=1}^{N} A_{ii}$. The graph diameter is denoted as $D_G$. Given the labeled node set $V_{L}$ and their labels $Y_{L}$, where each node $v_i$ is associated with a label $y_i$, semi-supervised node classification aims to train a node classifier $f_{\theta}: \psi \rightarrow \mathbb{R}^{C}$ to predict the labels $Y_{L}$ of remaining nodes $\{V_{U} = V \setminus V_{L}\}$, where $C$ denotes the number of classes. We separate the labeled node set $V_{L}$ into $\{V_{L}^{1}, V_{L}^{2}, \cdots, V_{L}^{C}\}$, where $V_{L}^{i}$ is the nodes of class $i$ in $V_{L}$.

3.2 Understanding Topology-Imbalance via Under-reaching and Over-squashing

In GNNs, node representations are learned by aggregating information from valuable neighbors. The quantity and quality of the information received by the nodes decide the expressiveness of their representations. We perceive the imbalance of the labeled node positions affects the performance of GNNs for two reasons:

1) Under-reaching: The influence from labeled nodes decays with the topology distance [5], resulting in that the nodes far away from labeled nodes lack supervision information. When the node can’t reach enough valuable labeled nodes within the receptive field of the model, the quantity of information it received is limited.

2) Over-squashing: The receptive field of GNNs is exponentially-growing and all information is compressed into fixed-length vectors [1]. The supervision information of valuable labeled nodes is squashed when passing across the narrow path together with other useless information.

3.3 Quantitative Analysis

To provide quantitative analysis for topology-imbalance, we propose two metrics for reachability and squashing. First, we define a reaching coefficient based on the shortest path, which determines the minimum layers of GNNs to obtain supervision information:

**Definition 1 (Reaching Coefficient).** Given a graph $G$ and labeled node set $V_{L}$, the reaching coefficient $RC$ of $G$ is the mean length of the shortest path from unlabeled nodes to the labeled nodes of their corresponding classes:

$$RC = \frac{1}{|V_U|} \sum_{v_i \in V_U} \frac{1}{|V_{L}^{y_i}|} \sum_{v_j \in V_{L}^{y_i}} \left(1 - \frac{\log |P_{sp}(v_i, v_j)|}{\log D_G}\right),$$

(1)

where $V_{L}^{y_i}$ denotes the nodes in $V_{L}$ whose label is $y_i$, $P_{sp}(v_i, v_j)$ denotes the shortest path between $v_i$ and $v_j$, and $|P_{sp}(v_i, v_j)|$ denotes its length, and $D_G$ is the diameter of graph $G$. Specifically, for the unconnected $v_i$ and $v_j$, we set the length of their shortest path as $D_G$.

The reaching coefficient reflects how long the the distance when the GNNs passes the valuable information to the unlabeled nodes. Note that $RC \in [0, 1]$ and larger $RC$ means better reachability.

For the quantitative metric of over-squashing, we define a squashing coefficient using the Ricci curvature to formulate it from a geometric perspective. The Ricci curvature [28] reflects the change of topology properties of the two endpoints of an edge, where the negative $Ric(v_i, v_j)$ means that the edge behaves locally as a shortcut or bridge and positive $Ric(v_i, v_j)$ indicates that locally there are more triangles in the neighborhood of $v_i$ and $v_j$ [27, 42].

**Definition 2 (Squashing Coefficient).** Given a graph $G$, the squashing coefficient $SC$ of $G$ is the mean Ricci curvature of edges on the shortest path from unlabeled nodes to the labeled nodes of their corresponding classes:

$$SC = \frac{1}{|V_U|} \sum_{v_i \in V_U} \frac{1}{|N_{y_i}(v_i)|} \sum_{v_j \in N_{y_i}(v_i)} \frac{\sum_{t_k \in P_{sp}(v_i, v_j)} Ric(t_k, v_i)}{|P_{sp}(v_i, v_j)|},$$

(2)

where $N_{y_i}(v_i)$ denotes the labeled nodes of class $y_i$ that can reach $v_i$, $Ric(\cdot, \cdot)$ denotes the Ricci curvature, and $|P_{sp}(v_i, v_j)|$ denotes the length of shortest path between $v_i$ and $v_j$.

We leverage the Ollivier-Ricci curvature [28] as $Ric(\cdot, \cdot)$ here:

$$Ric(v_k, v_l) = \frac{\text{Wasserstein}(mass_k, mass_l)}{d_{geo}(v_k, v_l)},$$

(3)

where $\text{Wasserstein}(\cdot, \cdot)$ is the Wasserstein distance, $d_{geo}(\cdot, \cdot)$ is the geodesic distance function, and $mass_k$ is the mass distribution [28] of node $v_k$. Note that $SC$ can be either positive or negative and larger $SC$ means lower squashing because the ring structures are more friendly for information sharing.

In Figure 2 and Figure 3, we show the relation between the reaching coefficient $RC$, the squashing coefficient $SC$, and the classification accuracy. The higher the accuracy, the darker and larger the corresponding scatter. First, we analyze the performance of GCN when trained with the same graph structure but with different labeled nodes. In Figure 2, we generate a synthetic graph by the Stochastic Block Model (SBM) [19] with 4 classes and 3,000 nodes. We randomly sample some nodes as the labeled nodes 10 times and scatter the classification accuracy in Figure 2. We can observe that even for the same graph structure, the difference in positions of labeled nodes may bring up to 15.42% difference in accuracy. There is a significant positive correlation between the reaching coefficient, the squashing coefficient, and the model performance.

Then we analyze the performance of GCN when trained with the same labeled nodes but on different graph structures. In Figure 3, we set the labeled nodes to be the same and generate different structures between them by controlling the edge probability between...
To form structure with better intra-class connectivity, we use an earning framework, to optimize the information propagation path.

### 4.1 Position-aware Structure Learning

In this section, we introduce PASTEL, a Position-Aware Structure Learning framework, to optimize the information propagation path directly and address the topology-imbalance issue in essence. In light of the analysis in Section 3.2, PASTEL aims to learn a better structure that increases the intra-class label influence for each class and thus relieves the under-reaching and over-squashing phenomena. The overall architecture of PASTEL is shown in Figure 4.

#### 4.1.1 Position-aware Structure Learning

To form structure with better intra-class connectivity, we use an anchor-based position encoding method to capture the topology distance between unlabeled nodes to labeled nodes. Then we incorporate both the merits of feature information as well as topology information to learn the refined structure.

**Anchor-based Position Encoding.** Inspired by the position in transformer [36, 43], we use an anchor-based position encoding method to capture the relative position of unlabeled nodes with respect to all the labeled nodes of the graph. Since we focus on maximizing the reachability between unlabeled nodes and labeled nodes within the same class, we directly separate the labeled node set \( \mathcal{V}_L \) into \( C \) anchor sets \( \{ \mathcal{V}_L^1, \mathcal{V}_L^2, \ldots, \mathcal{V}_L^C \} \), where each subset \( \mathcal{V}_L^C \) denotes the labeled nodes whose labels are \( c \). The class-wise anchor sets help distinguish the information from different classes rather than treating all the anchor nodes the same and ignoring the class difference as in [49]. Concretely, for any node \( v_i \), we consider a function \( \phi(v_i, \cdot) \) which measures the position relations between \( v_i \) and the anchor sets in graph \( G \). The function can be defined by the connectivity between the nodes in the graph.

\[
\text{p}_i = \left( \phi(v_i, \mathcal{V}_L^1), \phi(v_i, \mathcal{V}_L^2), \ldots, \phi(v_i, \mathcal{V}_L^C) \right),
\]

(4)

where \( \phi(v_i, \mathcal{V}_L^c) \) is the position encoding function defined by the connectivity between the node \( v_i \) and the anchor set \( \mathcal{V}_L^c \) in graph. Here we choose \( \phi(v_i, \mathcal{V}_L^c) \) to be the mean length of shortest path between \( v_i \) and nodes in \( \mathcal{V}_L^c \) if two nodes are connected:

\[
\phi(v_i, \mathcal{V}_L^c) = \frac{\sum_{j \in N_c(v_i)} |P_{sp}(v_i, v_j)|}{|N_c(v_i)|},
\]

(5)

where \( N_c(v_i) \) is the nodes connected with \( v_i \) in \( \mathcal{V}_L^c \) and \( |P_{sp}(v_i, v_j)| \) is the length of shortest path between \( v_i \) and \( v_j \). Then we transform the position encoding into the \( d_h \) dimensional space:

\[
h_i^h = W_\phi \cdot \text{p}_i,
\]

(6)

where \( W_\phi \) is a trainable vector. If two nodes have similar shortest paths to the anchor sets, their position encodings are similar.

**Position-aware Metric Learning.** After obtaining the position encoding, we use a metric function that accounts for both node feature information and the position-based similarities to measure the possibility of edge existence. PASTEL is agnostic to various similarity metric functions and we choose the widely used multi-head cosine similarity function here:

\[
a_{ij}^p = \frac{1}{m} \sum_{h=1}^{m} \cos \left( \text{W}_h \cdot (z_i || h^p_j) \right) \cdot \text{W}_h \cdot (z_j || h^p_i),
\]

(7)

where \( m \) is the number of heads, \( \text{W}_h \) is the weight matrix of the \( h \)-th head, \( z_i \) denotes the representation vector of node \( v_i \) and || denotes...
We aim to increase the intra-class connectivity among nodes, thereby increasing the supervision information they received and their influence on each other. Here we propose a class-wise conflict measure to guide what nodes should be more closely connected. According to the inherent relation of GNNs with Label Propagation [7, 45], we use a cosine annealing mechanism to calculate the edge weights by the relative position encoding. Then we propose a class-wise conflict measure to learn a graph with better intra-class connectivity.

4.2 Class-wise Conflict Measure

We aim to increase the intra-class connectivity among nodes, thereby increasing the supervision information they received and their influence on each other. Here we propose a class-wise conflict measure to measure their conflict when exchanging information. We use a cosine annealing mechanism to calculate the edge weights by the relative position encoding. Then we propose a class-wise conflict measure to learn a graph with better intra-class connectivity.

The GPR value contains not only the global topology information but also the annotation information. The supervision influence of labeled nodes of class $c$ is used as the connection strength of edge $e_{ij}$, with the corresponding element $a^p_{ij}$ in the adjacency matrix being:

$$a^p_{ij} = w_{ij} \cdot a^p_{ij}.$$  

(13)

The effectiveness of the class-wise conflict measure is evaluated in Section 5.3.2 and the change of GPR vectors is shown in Section 5.4.3.

4.3 Learning with the Optimized Structure

With the above structure learning strategy, we can obtain a position-aware adjacency $A_P$ with maximum intra-class connectivities:

$$A_P = \{a^p_{ij}, i, j \in \{1, 2, \cdots, N\}\}.$$  

(14)

The input graph structure determines the learning performance to a certain extent. Since the structure learned at the beginning is of poor quality, directly using it may lead to non-convergence or unstable training of the whole framework. We hence incorporate the original graph structure $A$ and a structure in a node feature view $A_N$ as supplementary to formulate an optimized graph structure $A^*$. Specifically, we also learn a graph structure $A_N = \{a^N_{ij}, i, j \in \{1, 2, \cdots, N\}\}$ in a node feature view with each element being:

$$a^N_{ij} = \frac{1}{m} \sum_{h=1}^{m} \cos \left( W_h \cdot \left( x_i \| h^p_h \right) \cdot W_h \cdot \left( x_j \| h^p_h \right) \right).$$  

(15)

where $x_i$ is the feature vector of node $v_i$ and $h^p_h$ is the position encoding with the original structure. Then we can formulate an optimized graph structure $A^*$ with respect to the downstream task:

$$A^* = \lambda_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} + (1 - \lambda_1) (\lambda_2 f(A_N) + (1 - \lambda_2) f(A_P)),$$  

(16)

where $f(\cdot)$ denotes the row-wise normalization function, and $\lambda_1$ and $\lambda_2$ are two constants that control the contributions of original structure.
The overall loss is defined as the combination of the node classification loss and graph regularization loss:

\[ L = L_{cls} + \beta_1 L_{smooth} + \beta_2 L_{con} + \beta_3 L_{spar}. \]

The overall process of PASTEL is shown in Algorithm 1.

Let's break down the algorithm into its components:

**Algorithm 1: The overall process of PASTEL**

**Input:** Graph \( G = (V, E) \) with node labels \( Y \); Number of heads \( m \); Number of training epochs \( E \); Structure fusion coefficients \( \lambda_1, \lambda_2 \); Loss coefficients \( \beta_1, \beta_2, \beta_3 \)

**Output:** Optimized graph \( G^* = (A^*, X) \), predicted label \( \hat{Y} \)

1. Parameter initialization;
2. for \( e = 1, 2, \cdots, E \) do
   1. Learn position-aware graph structure
   2. Learn position encodings \( h_{ij}^p \) \(
   \text{Eq. (6)} \)
   3. Learn edge possibility \( d_{ij}^e \) \(
   \text{Eq. (7)} \)
   4. Calculate the Group PageRank matrix \( p_{ij}^{PR} \) \(
   \text{Eq. (10)} \)
   5. Calculate the class-wise conflict measure \( w_{ij} \) \(
   \text{Eq. (12)} \)
   6. Obtain position-aware structure \( A_P \) \(
   \text{Eq. (14)} \)
   7. Learn node representations
   8. Obtain the optimized structure \( A^* \) \(
   \text{Eq. (16)} \)
   9. Calculate representations and labels \( Z, \hat{Y} \) \(
   \text{Eq. (20)} \)
   10. Optimize the losses \( L_{cls} \) \(
   \text{Eq. (21)} \), \( L_{smooth} \) \(
   \text{Eq. (17)} \), \( L_{con} \) \(
   \text{Eq. (18)} \), and \( L_{spar} \) \(
   \text{Eq. (19)} \)
   11. Update model parameters to minimize \( L \) \(
   \text{Eq. (22)} \)
3. end

5 EXPERIMENT

In this section, we first evaluate PASTEL\(^1\) on both real-world graphs and synthetic graphs. Then we analyze the main mechanisms of PASTEL and the learned structure. We mainly focus on the following research questions:

- **RQ1.** How does PASTEL perform in the node classification task? (Section 5.2)
- **RQ2.** How does the position encoding and the class-wise conflict measure influence the performance of PASTEL? (Section 5.3)
- **RQ3.** What graph structure PASTEL tend to learn? (Section 5.4)

5.1 Experimental Setups

5.1.1 Datasets. We conduct experiments on synthetic and real-world datasets to analyze the model’s capabilities in terms of both graph theory and real-world scenarios. The real-word datasets include various networks with different heterophily degrees to demonstrate the generalization of PASTEL. Cora and Citeseer [35] are citation networks. Photo [37] and and Actor [31] are co-occurrence network. Chameleon and Squirrel [34] are page-page networks in Wikipedia. Since we focus on the topology-imbalance issue in this work, we set the number of labeled nodes in each class to be 20.

5.1.2 Baselines. We choose representative GNNs as backbones including GCN [22], GAT [44], APPNP [23], and GraphSAGE [17]. The most important baseline is ReNode [7], which is the only existing work for the topology-imbalance issue. We also include some graph structure learning baselines to illustrate the specific effectiveness of PASTEL for the topology-imbalance issue. DropEdge [33] randomly removes edges at each epoch as structure augmentation. To evaluate the effect of increasing the reachability randomly, we use a adding edges method named AddEdge, whose adding strategy is similar to DropEdge. SDRF [42] rewires edges according to their curvatures for the over-squashing issue. NeuralSparse [54] removes potentially task-irrelevant edges for clearer class boundaries. IDGL [10] updates the node representations and structure based on these representations iteratively.

5.1.3 Parameter Settings. For the GNN backbones, we set their depth to be 2 layers and adopt the implementations from the PyTorch Geometric Library in all experiments. We set the representation dimension of all baselines and PASTEL to be 256. We re-implement the NeuralSparse [54] and SDRF [42] and the parameters of baseline methods are set as the suggested value in their papers or carefully tuned for fairness. For DropEdge and AddEdge, we set the edge dropping/adding probability to 10%. For PASTEL, we set the number of heads \( m = 4 \) and the random walk restart probability \( \alpha = 0.15 \). The structure fusing coefficients \( \lambda_1 \) and \( \lambda_2 \) and the loss coefficients \( \beta_1, \beta_2, \beta_3 \) are tuned for each dataset.

5.2 Evaluation (RQ1)

5.2.1 PASTEL for Real-world Graphs. We compare PASTEL with the baselines on several datasets on node classification. The overall Weighted-F1 (W-F1) scores and the class-balance Macro-F1 (M-F1) scores on different baselines are shown in Table 1. The best

---

\( ^1 \)The code of PASTEL is available at https://github.com/RingBDStack/PASTEL.
Table 1: Weighted-F1 score and Macro-F1 score (% ± standard deviation) of node classification on real-world graph datasets.

| Backbone | Model | Cora | CiteSeer | Photo | Actor | Chameleon | Squirrel |
|----------|-------|------|---------|-------|-------|-----------|----------|
| GAT      | original | 76.41±1.7 | 64.41±17 | 60.61±17 | 88.21±2.9 | 86.22±2.6 | 21.18±12 | 20.99±11 | 29.93±3.5 | 20.51±12 | 20.51±14 |
|          | ReNode  | 78.91±2.1 | 72.1±15 | 61.0±15 | 89.1±24 | 87.1±26 | 21.5±12 | 20.5±11 | 29.2±23 | 20.7±10 | 20.4±11 |
|          | AddEdge | 78.0±1.6 | 72.6±1±6 | 60.2±13 | 88.2±24 | 86.2±25 | 21.3±12 | 20.3±11 | 29.8±17 | 20.7±16 | 20.7±16 |
|          | SDRF    | 78.7±1.3 | 76.9±15 | 65.4±14 | 89.1±19 | 87.1±21 | 22.9±12 | 21.8±11 | 30.3±16 | 21.2±15 | 21.2±15 |
|          | NeuralSparse | 81.7±1.4 | 80.9±1.4 | 71.8±12 | 69.0±10 | 87.9±19 | 87.7±18 | 24.4±15 | 23.6±16 | 44.9±30 | 44.9±28 | 28.1±18 | 28.1±18 |
|          | IDGL    | 82.3±0.9 | 81.0±0.9 | 71.7±10 | 68.0±13 | 86.6±13 | 88.8±14 | 24.9±10 | 22.0±07 | 55.4±16 | 55.0±17 | 28.8±23 | 28.9±22 |
| PASTEL   | 82.5±0.3 | 81.2±0.3 | 72.9±0.8 | 69.3±0.9 | 91.4±2.7 | 93.3±2.2 | 26.4±10 | 24.4±12 | 57.8±2.4 | 57.3±2.4 | 37.5±0.6 | 37.5±0.7 |

Table 2: Weighted-F1 scores and improvements on graphs with different levels of topology-imbalance.

| Cora-L | Cora-M | Cora-H | RC | SC | RC | SC | RC | SC | RC | SC | RC | SC | RC | SC |
|--------|--------|--------|----|----|----|----|----|----|----|----|----|----|----|----|
| 0.4130 | 0.6383 | 0.4100 | 0.6024 | 0.4060 | 0.6302 |
| W-F1 (%) | F-W1 (%) | W-F1 (%) | F-W1 (%) | W-F1 (%) | F-W1 (%) |
| GCN    | 80.9±0.9 | 75.3±1.0 | 80.9±0.9 | 75.3±1.0 |
| ReNode | 81.3±0.7 | 76.4±0.8 | 70.4 | 75.3±0.9 | 70.5 | 78.3±1.1 | 70.8 |
| SDRF   | 81.0±0.7 | 70.1 | 78.9±0.8 | 70.1 | 77.9±0.1 | 70.4 |
| IDGL   | 82.5±1.0 | 71.6 | 80.4±1.0 | 71.6 | 81.6±1.1 | 71.4 |
| PASTEL | 82.7±0.9 | 71.8 | 81.0±0.9 | 72.2 | 81.9±1.1 | 71.4 |

results are shown in bold and the runner-ups are underlined. PASTEL shows overwhelming superiority in improving the performance of backbones on all datasets. It demonstrates that PASTEL is capable of learning better structures with a more balanced label distribution that reinforces the GNN models. ReNode [7] achieves fewer improvements on datasets of poor connectivity (e.g., CiteSeer) and even damages the performance of backbones on heterogeneous datasets (e.g., Chameleon and Actor). We think it’s because ReNode [7] detects conflicts by Personalized PageRank and fails to reflect the node topological position well when the graph connectivity is poor. Besides, ReNode takes the topology decision boundary as the decision boundary, which is not applicable for heterogeneous graphs. AddEdge doesn’t work in most cases, demonstrating that randomly adding edge is not effective in boosting the reachability. The structure augmentation strategy should be carefully designed considering the node relations. SDRF [42] can improve the performance, supporting our intuition that relieving over-squashing helps graph learning. But SDRF is still less effective than PASTEL because it only considers the topological properties rather than the supervision information. Both NeuralSparse [54] and IDGL [10] show good performance among the baselines, showing the effectiveness of learning better structures for downstream tasks. However, they are still less effective than PASTEL which takes the supervision information distribution into consideration.

5.2.2 PASTEL under Different Levels of Topology-imbalance. To further analyze PASTEL’s ability in alleviating the topology-imbalance issue, we verify the PASTEL under different levels of topology-imbalance. We randomly sampled 1,000 training sets and calculate the reaching coefficient RC and squashing coefficient SC as introduced in Section 3.2. Then we choose 3 training sets with different levels of topology-imbalance according to the conclusion in Section 3.3 and we denote them as Cora-L, Cora-M, and Cora-H, according to the degree of topology imbalance. Note that larger RC means better reachability and larger SC means lower squashing. We evaluate PASTEL and several baselines with the GCN as the backbone and show the dataset information, the Weighted-F1 scores, and their improvements (Δ) over the backbones in Table 2. The performance of node representation learning generally gets
worse with the increase of the topology-imbalance degree of the dataset. Both the node re-weighting method (i.e., ReNode [7]) and the structure learning methods (i.e., IDGL [10], SDRF [42] and PASTEL) can achieve more improvement with the increase of dataset topology-imbalance. PASTEL performs best on all datasets with different degrees of topology-imbalance and it can achieve up to 4.4% improvement on the highly topology-imbalance dataset.

5.2.3 PASTEL for Synthetic Graphs. We generate 7 synthetic graph datasets with different community structures using the Stochastic Block Model (SBM) \( \mathcal{G}(N, C, p, q) \) [19], where the number of nodes \( N = 3000 \), the number of community \( C = 6 \), \( p \) denotes the edge probability within a community and \( q \) denotes the edge probability between communities. We show the classification Weighted-F1 scores and improvements are shown in Table 3. With a more clear community structure, the reaching coefficient \( RC \) increases and the squashing coefficient \( SC \) also increases, leading to the increase of GCN’s performance, which agrees with the conclusion obtained in Section 3.3. ReNode shows unsatisfied performance in boosting the node classification. PASTEL can increase the classification weighted-F1 score by 5.38%-21.35% on SBM graphs with different community structures, showing superior effectiveness.

5.3 Analysis of PASTEL (RQ2)
We conduct ablation studies for the two main mechanisms of PASTEL, position encoding and class-wise conflict measure.

5.3.1 Impact of the Position Encoding. We design an anchor-based position encoding mechanism in Section 4.1, which reflects the relative topological position to labeled nodes and further maximizes the label influence within a class. To evaluate the effectiveness of position encoding, we compare PASTEL with a variant PASTEL (w/o PE), which removes the position encoding and directly takes the node features for metric learning in Eq. (7). Here we use the GCN as the backbone. As shown in Figure 5, the structure learning strategy of PASTEL contributes the most, which can achieve at most 25.5% improvement in terms of Weighted-F1 score with only node features. Although PASTEL (w/o PE) effectively improves the performance of backbones to some extent, the position encoding still benefits learning better structure to relieve the topology-imbalance with 1.0%-1.8% improvements than PASTEL (w/o PE).

5.3.2 Impact of the Class-wise Conflict Measure. We designed a class-wise conflict measure in Section 4.2 as edge weights to guide learning structures with better intra-class connectivity. Here, we compare PASTEL with its two variants to analyze the impact of class-wise conflict measure: (1) PASTEL (w/o CCM), which removes the class-wise conflict measure and directly takes the learned edge possibilities in Eq. (7) as the edge weights. (2) PASTEL (Totoro), which takes the Totoro metric introduced in ReNode [7] as the conflict measure of nodes in Eq. (13). Here we use the GCN as the backbone. The comparison results are shown in Figure 6. On four datasets, PASTEL consistently outperforms the other two variants. Even without the conflict measure, PASTEL (w/o CCM) still shows better performance than PASTEL (Totoro), indicating the limitation of ReNode when capturing the relative topology positions without clear homophily structures.

5.4 Analysis of the Learned Structure (RQ3)
We analyze the learned graph by PASTEL in terms of visualization and structural properties.

| SBM-1 | SBM-2 | SBM-3 | SBM-4 | SBM-5 | SBM-6 | SBM-7 |
|-------|-------|-------|-------|-------|-------|-------|
| \( p \) | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 |
| \( q \) | 0.0300 | 0.0100 | 0.0083 | 0.0071 | 0.0063 | 0.0056 |
| \( RC \) | 0.4979 | 0.4984 | 0.4990 | 0.4994 | 0.5002 | 0.5004 |
| \( SC \) | 0.0998 | 0.0999 | 0.1000 | 0.1000 | 0.1000 | 0.1017 |

Table 3: Weighted-F1 scores (%) and improvements (\( \Delta \)) on synthetic SBM graphs with different community structures.
Figure 7: Structure visualization. (a) Original graph of Cora and learned graphs by (b) ReNode, (c) SDRF, (d) IDGL and (e) PASTEL.

Table 4: Properties and performance of the original graph and learned graphs of Cora.

|                | Original Graph | ReNode | SDRF | IDGL | PASTEL |
|----------------|----------------|--------|------|------|--------|
| RC             | 0.4022         | 0.4022 | 0.4686| 0.3028| 0.3475 |
| SC             | -0.6299        | -0.6299| -0.4942| -0.4069| -0.3389|
| W-F1 (%)       | 79.44          | 80.34  | 82.01 | 82.38 | 82.86  |

5.4.1 Structure Visualization. In Figure 7, we visualize the original graph of Cora and the graphs learned by ReNode [7], SDRF [42], IDGL [10] and PASTEL using networkx. For clarity, the edges are not shown. The solid points denote the labeled nodes, the hollow points denote the unlabeled nodes, and the layout of nodes denotes their connectivities. The node size in Figure 7(b) denotes the learned node weight in ReNode, and the solid lines and dashed lines in Figure 7(c) denote the added and deleted edges by SDRF, respectively. As we can observe, ReNode gives more weights to nodes in the topology center of each class and SDRF tends to build connections between distant or isolated nodes. Even though the structure learned by IDGL can make the nodes of a class close, there are still some overlapping and entangled areas between classes. Benefiting from the position encoding and class-wise conflict measure, PASTEL can obtain graph structure with clearer class boundaries.

5.4.2 Change of RC and SC. We also show the reaching coefficient RC and the squashing coefficient SC of the above graphs in Figure 7 and the Weighted-F1 score learned on them in Table 4. Here we choose the GCN as the model backbone. All of the structure learning methods (SDRF [42], IDGL [10] and PASTEL) learn structures with larger reaching coefficient and larger squashing coefficient, leading to the performance improvement of node classification. This phenomenon supports our propositions in Section 3.3 again.

5.4.3 Change of GPR Vector. The class-wise conflict measure is calculated by the Group PageRank (GPR), which reflects the label influence of each class. We randomly choose 10 nodes for each class in Cora and show their GPR vectors $\mathbf{P}_{i}^{gpr}$ in the original graph in Figure 8(a) and the learned graph in Figure 8(b), respectively, where the color shade denotes the magnitude, $V_i$ denotes 10 nodes of class $i$ and $C_j$ denotes the $j$-th class. In Figure 8(a), the off-diagonal color blocks are also dark, indicating that the label influence of each class that nodes obtained from the original graph is still entangled to some extent, which could bring difficulties to the GNN optimization. After the structure learning guided by the proposed class-wise conflict measure, Figure 8(b) exhibits 7 clear diagonal blocks and the gaps between the diagonal and off-diagonal block are widened, indicating that nodes can receive more supervision information of its ground-truth class. We can further make a conclusion that the class-wise conflict measure plays an important role on giving guidance for more class connectivity orthogonality.

6 CONCLUSION

We proposed a novel framework named PASTEL for the graph topology-imbalance issue. We provide a new understanding and two quantitative analysis metrics of topology-imbalance in the perspective of under-reaching and over-squashing, answering the questions that how topology-imbalance affects GNN's performance as well as what graphs are susceptible to it. PASTEL designs an anchor-based position encoding mechanism and a class-wise conflict measure to obtain structures with better in-class connectivity. Comprehensive experiments demonstrate the potential and adaptability of PASTEL. An interesting future direction is to incorporate the proposed two quantitative metrics into the learning process to address topology-imbalance more directly.

ACKNOWLEDGMENTS

The corresponding author is Jianxin Li. The authors of this paper were supported by the NSFC through grant 6187202 and the Academic Excellence Foundation of BUAA for PhD Students. Philip S. Yu was supported by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

1856
