Overcoming Class Imbalance: Unified GNN Learning with Structural and Semantic Connectivity Representations

Abdullah Alchihabi[1] **Hao Yan**[1] **Yuhong Guo**[1,2]
[1]Carleton University, Ottawa, Canada
[2]Canada CIFAR AI Chair, Amii, Canada
{abdullahalchihabi@cmail., haoyan6@cmail., yuhong.guo@}carleton.ca

Abstract

Class imbalance is pervasive in real-world graph datasets, where the majority of annotated nodes belong to a small set of classes (majority classes), leaving many other classes (minority classes) with only a handful of labeled nodes. Graph Neural Networks (GNNs) suffer from significant performance degradation in the presence of class imbalance, exhibiting bias towards majority classes and struggling to generalize effectively on minority classes. This limitation stems, in part, from the message passing process, leading GNNs to overfit to the limited neighborhood of annotated nodes from minority classes and impeding the propagation of discriminative information throughout the entire graph. In this paper, we introduce a novel Unified Graph Neural Network Learning (Uni-GNN) framework to tackle class-imbalanced node classification. The proposed framework seamlessly integrates both structural and semantic connectivity representations through semantic and structural node encoders. By combining these connectivity types, Uni-GNN extends the propagation of node embeddings beyond immediate neighbors, encompassing non-adjacent structural nodes and semantically similar nodes, enabling efficient diffusion of discriminative information throughout the graph. Moreover, to harness the potential of unlabeled nodes within the graph, we employ a balanced pseudo-label generation mechanism that augments the pool of available labeled nodes from minority classes in the training set. Experimental results underscore the superior performance of our proposed Uni-GNN framework compared to state-of-the-art class-imbalanced graph learning baselines across multiple benchmark datasets.

1 Introduction

Graph Neural Networks (GNNs) have exhibited significant success in addressing the node classification task [Kipf and Welling, 2017, Hamilton et al., 2017, Veličković et al., 2018] across diverse application domains from molecular biology [Hao et al., 2020] to fraud detection [Zhang et al., 2021]. The efficacy of GNNs has been particularly notable when applied to balanced annotated datasets, where all classes have a similar number of labeled training instances. The performance of GNNs experiences a notable degradation when confronted with an increasingly imbalanced class distribution in the available training instances [Yun et al., 2022]. This decline in performance materializes as a bias towards the majority classes, which possess a considerable number of labeled instances, resulting in a challenge to generalize effectively over minority classes that have fewer labeled instances [Park et al., 2021, Yan et al., 2023]. The root of this issue lies in GNNs' reliance on message passing to disseminate information across the graph. Specifically, when the number of labeled nodes for a

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Advancing Graph Machine Learning.

particular class is limited, GNNs struggle to propagate discriminative information related to that class throughout the entire graph. This tendency leads to GNNs' overfitting to the confined neighborhood of labeled nodes belonging to minority classes [Tang et al., 2020, Yun et al., 2022, Li et al., 2023]. This is commonly denoted as the 'under-reaching problem' [Sun et al., 2022] or 'neighborhood memorization' [Park et al., 2021].

In this study, we present a novel Unified Graph Neural Network Learning (Uni-GNN) framework designed to tackle the challenges posed by class-imbalanced node classification tasks. Our proposed framework leverages both structural and semantic connectivity representations, specifically addressing the under-reaching and neighborhood memorization issues. To achieve this, we construct a structural connectivity based on the input graph structure, complemented by a semantic connectivity derived from the similarity between node embeddings. The Uni-GNN framework's unique utilization of both structural and semantic connectivity empowers it to effectively extend the propagation of node embeddings beyond the standard neighborhood. Moreover, to harness the potential of unlabeled nodes in the graph, we introduce a balanced pseudo-label generation method. This method strategically samples unlabeled nodes with confident predictions in a class-balanced manner, effectively increasing the number of labeled instances for minority classes. Our experimental evaluations on multiple benchmark datasets underscore the superior performance of the proposed Uni-GNN framework compared to state-of-the-art Graph Neural Network methods designed to address class imbalance.

2 Method

In the context of semi-supervised node classification with class imbalance, we consider a graph $G = (V, E)$, where V represents the set of $N = |V|$ nodes, and E denotes the set of edges within the graph. E is commonly represented by an adjacency matrix A of size $N \times N$. This matrix is assumed to be symmetric (i.e. for undirected graphs), and it may contain either weighted or binary values. Each node in the graph is associated with a feature vector of size D, while the feature vectors for all the N nodes are organized into an input feature matrix $X \in \mathbb{R}^{N \times D}$. The set of nodes V is partitioned into two distinct subsets: V_l, comprising labeled nodes, and V_u, encompassing unlabeled nodes. The labeled nodes in V_l are paired with class labels, and this information is encapsulated in a label indicator matrix $Y^l \in \{0, 1\}^{N_l \times C}$. Here, C signifies the number of classes, and N_l is the number of labeled nodes. Further, V_l can be subdivided into C non-overlapping subsets, denoted as $\{V_l^1, \cdots, V_l^C\}$, where each subset V_l^i corresponds to the labeled nodes belonging to class i. It is noteworthy that V_l exhibits class imbalance, characterized by an imbalance ratio ρ. This ratio, defined as $\rho = \frac{\min_i(|V_l^i|)}{\max_i(|V_l^i|)}$, is considerably less than 1. Such class imbalance problem introduces challenges and necessitates specialized techniques in the development of effective node classification models.

2.1 Unified GNN Learning Framework

The Unified GNN Learning (Uni-GNN) framework comprises three crucial components: the structural node encoder, the semantic node encoder and the balanced node classifier. The overall framework of Uni-GNN is illustrated in Figure 1.

2.1.1 Structural Node Encoder

The objective of the structural encoder is to learn an embedding of the nodes based on structural connectivity. Instead of directly using the input adjacency matrix A, we construct a new structural connectivity-based graph adjacency matrix A_{struct} that extends connections beyond the immediate neighbors in the input graph. This matrix is determined by the distances between pairs of nodes, measured in terms of the number of edges along the shortest path connecting the respective nodes, such that

$$A_{\text{struct}}[i, j] = \begin{cases} \frac{1}{\text{SPD}(i,j)} & \text{SPD}(i, j) \leq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\text{SPD}(.,.)$ is the shortest path distance function that measures the distance between pairs of input nodes in terms of the number of edges along the shortest path in the input graph G. The hyper-parameter $\alpha \geq 1$ governs the maximum length of the shortest path distance to be considered. In A_{struct}, edges connecting node pairs are assigned weights that are inversely proportional to the length

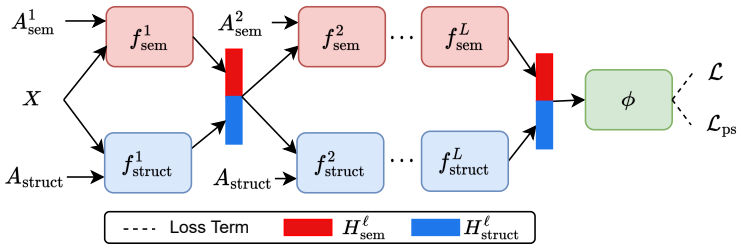

Figure 1: Overview of the proposed Unified GNN Learning framework. The structural (f_{struct}) and semantic (f_{sem}) node encoders leverage their respective connectivity matrices—structural (A_{struct}) and semantic ($\{A^\ell_{\text{sem}}\}^L_{\ell=1}$). The encoders share concatenated node embeddings—structural ($H^{\ell-1}_{\text{struct}}$) and semantic ($H^{\ell-1}_{\text{sem}}$)—at each message passing layer (ℓ). The balanced node classifier (ϕ) utilizes the final unified node embeddings ($H^L_{\text{struct}}||H^L_{\text{sem}}$) for both node classification and balanced pseudo-label generation.

of the shortest path between them. This design ensures that the propagated messages carry importance weights, scaling the messages based on the corresponding edge weights between connected nodes.

The structural node encoder, denoted as f_{struct}, consists of L message propagation layers. In each layer ℓ of the structural node encoder, denoted as f^ℓ_{struct}, the node input features comprise the learned node embeddings, $H^{\ell-1}_{\text{struct}}$ and $H^{\ell-1}_{\text{sem}}$, from the previous layer of both the structural encoder and the semantic encoder, respectively. As a consequence, the propagated messages encode both semantic and structural information facilitating the learning of more discriminative node embeddings. The constructed structural connectivity matrix A_{struct} is employed as the adjacency matrix for message propagation within each layer. We employ the conventional Graph Convolution Network (GCN) [Kipf and Welling, 2017] as our message-passing layer, given its simplicity, efficiency and ability to handle weighted graph structures, in the following manner:

$$\begin{aligned} H^\ell_{\text{struct}} &= f^\ell_{\text{struct}}(H^{\ell-1}_{\text{struct}}||H^{\ell-1}_{\text{sem}}, A_{\text{struct}}) \\ &= \sigma\left(\hat{D}^{-\frac{1}{2}}_{\text{struct}}\hat{A}_{\text{struct}}\hat{D}^{-\frac{1}{2}}_{\text{struct}}(H^{\ell-1}_{\text{struct}}||H^{\ell-1}_{\text{sem}})W^\ell_{\text{struct}}\right) \end{aligned} \tag{2}$$

Here, σ represents the non-linear activation function, "||" denotes the feature concatenation operation, W^ℓ_{struct} is the matrix of learnable parameters for f^ℓ_{struct}, $\hat{A}_{\text{struct}} = A_{\text{struct}} + I$ is the adjusted adjacency matrix with self-connections, and $\hat{D}_{\text{struct}}$ is the diagonal node degree matrix of $\hat{A}_{\text{struct}}$ such that $\hat{D}_{\text{struct}}[i,i] = \sum_j \hat{A}_{\text{struct}}[i,j]$. In the case of the first layer of f_{struct}, the node input features are solely represented by the input feature matrix X.

2.1.2 Semantic Node Encoder

The objective of the semantic node encoder is to learn node embeddings based on the semantic connectivity. The semantic node encoder, denoted as f_{sem}, comprises L message passing layers. In each layer ℓ of the semantic node encoder, represented by f^ℓ_{sem}, a semantic-based graph adjacency matrix A^ℓ_{sem} is constructed based on the similarity between the embeddings of nodes from the previous layer of the semantic and structural node encoders, measured in terms of clustering assignments. For each layer ℓ, the following fine-grained node clustering is performed:

$$S^\ell = g(H^{\ell-1}_{\text{struct}}||H^{\ell-1}_{\text{sem}}, K) \tag{3}$$

which clusters all the graph nodes into K ($K \gg C$) clusters. Here, $S^\ell \in \mathbb{R}^{N\times K}$ is the fine-grained clustering assignment matrix obtained from the clustering function g. The row $S^\ell[i]$ indicates the cluster to which node i is assigned. The fine-grained clustering function g takes as input the concatenation of the structural and semantic node embeddings from layer $\ell-1$, along with the number of clusters K, and outputs the cluster assignments S^ℓ. The clustering function g is realized by performing K-means clustering to minimize the following least squares clustering loss:

$$\mathcal{L}_{\text{clust}} = \sum_{i\in V}\sum_{k=1}^{K} S^\ell[i,k]\left\|(H^{\ell-1}_{\text{struct}}[i]||H^{\ell-1}_{\text{sem}}[i]) - \boldsymbol{\mu}_k\right\|^2 \tag{4}$$

where $\boldsymbol{\mu}_k$ represents the mean vector for the cluster k, and $S^\ell[i,k]$ has a binary value (0 or 1) that indicates whether node i is assigned to cluster k. Based on the fine-grained clustering assignment matrix S^ℓ, the construction of the semantic connectivity-based graph adjacency matrix A^ℓ_{sem} is detailed as follows:

$$A^\ell_{\text{sem}}[i,j] = \begin{cases} 1 & \text{if } S^\ell[i] = S^\ell[j] \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In the construction of A^ℓ_{sem}, nodes assigned to the same cluster are connected, establishing edges between them, while nodes assigned to different clusters are not connected, resulting in an adjacency matrix that encapsulates the semantic connectivity encoded within the fine-grained clusters. This process enables message propagation among nodes that share semantic similarities in the graph, irrespective of their structural separation. This is instrumental in addressing the issue of under-reaching of minority nodes.

Each layer ℓ of the semantic node encoder, f^ℓ_{sem}, takes the concatenation of node embeddings, $H^{\ell-1}_{\text{struct}} \| H^{\ell-1}_{\text{sem}}$, from the previous layer $\ell-1$ of both the structural encoder and the semantic encoder as input, aiming to gather richer information from both aspects, but propagates messages with the constructed semantic adjacency matrix A^ℓ_{sem}. For the first layer of f_{sem}, the input node features are simply the input features matrix X. We opt for the conventional Graph Convolution Network (GCN) [Kipf and Welling, 2017] as our message-passing layer again, which is employed in the following manner:

$$\begin{aligned} H^\ell_{\text{sem}} &= f^\ell_{\text{sem}}(H^{\ell-1}_{\text{struct}} || H^{\ell-1}_{\text{sem}}, A^\ell_{\text{sem}}) \\ &= \sigma\left(\hat{D}^{-\frac{1}{2}}_{\text{sem}} \hat{A}^\ell_{\text{sem}} \hat{D}^{-\frac{1}{2}}_{\text{sem}} (H^{\ell-1}_{\text{struct}} || H^{\ell-1}_{\text{sem}}) W^\ell_{\text{sem}}\right) \end{aligned} \tag{6}$$

Here, σ again represents the non-linear activation function; W^ℓ_{sem} is the matrix of learnable parameters for f^ℓ_{sem}; $\hat{A}^\ell_{\text{sem}} = A^\ell_{\text{sem}} + I$ is the adjusted adjacency matrix with self-connections; and the diagonal node degree matrix $\hat{D}_{\text{sem}}$ is computed as $\hat{D}_{\text{sem}}[i,i] = \sum_j \hat{A}^\ell_{\text{sem}}[i,j]$.

2.1.3 Balanced Node Classifier

We define a balanced node classification function ϕ, which classifies the nodes in the graph based on their structural and semantic embeddings learned by the Structural Encoder and Semantic Encoder respectively. In particular, the balanced node classification function takes as input the output of the L-th layers of the structural and semantic node encoders, denoted as H^L_{struct} and H^L_{sem}, respectively:

$$P = \phi(H^L_{\text{struct}} || H^L_{\text{sem}}) \tag{7}$$

where $P \in \mathbb{R}^{N \times C}$ is the predicted class probability matrix of all the nodes in the graph. Given the class imbalance in the set of labeled nodes V_l, the node classification function is trained to minimize the following weighted node classification loss on the labeled nodes:

$$\mathcal{L} = \sum\nolimits_{i \in V_l} \omega_{c_i} \ell_{\text{ce}}(P_i, Y^l_i). \tag{8}$$

Here, ℓ_{ce} denotes the standard cross-entropy loss function. For a given node i, P_i represents its predicted class probability vector, and Y^l_i is the true label indicator vector if i is a labeled node. The weight ω_{c_i} associated with each node i is introduced to balance the contribution of data from different classes in the supervised training loss. It gives different weights to nodes from different classes. In particular, the class balance weight ω_{c_i} is calculated as follows:

$$\omega_{c_i} = \frac{1}{|V^{c_i}_l|} \tag{9}$$

where c_i denotes the class index of node i, such that $Y^l[i,c_i] = 1$; and $|V^{c_i}_l|$ is the size of class c_i in the labeled nodes— i.e., the number of labeled nodes $V^{c_i}_l$ from class c_i. Since ω_{c_i} is inversely proportional to the corresponding class size, it enforces that larger weights are assigned to nodes from minority classes with fewer labeled instances in the supervised node classification loss, while smaller weights are assigned to nodes from majority classes with abundant labeled nodes. Specifically, through the incorporation of this class weighting mechanism, each class contributes equally to the supervised loss function, irrespective of the quantity of labeled nodes associated with it within the training set, thereby promoting balanced learning across different classes.

2.2 Balanced Pseudo-Label Generation

To leverage the unlabeled nodes in the graph, a balanced pseudo-label generation mechanism is proposed. The objective is to increase the number of available labeled nodes in the graph while considering the class imbalance in the set of labeled nodes. The goal is to generate more pseudo-labels from minority classes and fewer pseudo-labels from majority classes, thus balancing the class label distribution of the training data. In particular, for each class c, the number of nodes to pseudo-label, denoted as $\hat{N}_c$, is set as the difference between the largest labeled class size and the size of class c, aiming to balance the class label distribution over the union of labeled nodes and pseudo-labeled nodes:

$$\hat{N}_c = \max_{i \in \{1,\ldots,C\}}(|V_l^i|) - |V_l^c| \tag{10}$$

The set of unlabeled nodes that can be confidently pseudo-labeled to class c can be determined as:

$$\hat{V}_u^c = \{i \mid \max(P_i) > \epsilon, \text{ argmax}(P_i) = c, i \in V_u\} \tag{11}$$

where ϵ is a hyperparameter determining the confidence prediction threshold. Balanced sampling is then performed on each set $\hat{V}_u^c$ by selecting the top $\hat{N}_c$ nodes, denoted as $\text{Top}_{\hat{N}_c}(\hat{V}_u^c)$, with the most confident pseudo-labels based on the predicted probability $P_i[c]$. This results in a total set of pseudo-labeled nodes, denoted as $\hat{V}_u$, from all classes:

$$\hat{V}_u = \{\text{Top}_{\hat{N}_1}(\hat{V}_u^1), \cdots, \text{Top}_{\hat{N}_C}(\hat{V}_u^C)\}. \tag{12}$$

The Unified GNN Learning framework is trained to minimize the following node classification loss over this set of pseudo-labeled nodes $\hat{V}_u$:

$$\mathcal{L}_{\text{ps}} = \frac{1}{|\hat{V}_u|} \sum\nolimits_{i \in \hat{V}_u} \ell_{\text{ce}}(P_i, \mathbf{1}_{\text{argmax}(P_i)}) \tag{13}$$

where ℓ_{ce} again is the standard cross-entropy loss function, P_i is the predicted class probability vector with classifier ϕ, and $\mathbf{1}_{\text{argmax}(P_i)}$ is a one-hot pseudo-label vector with a single 1 at the predicted class entry $\text{argmax}(P_i)$. This pseudo-labeling mechanism aims to augment the labeled node set, particularly focusing on addressing class imbalances by generating more pseudo-labels for minority classes.

Training Loss The unified GNN Learning framework is trained on the labeled set V_l and the selected pseudo-labeled set $\hat{V}_u$ in an end-to-end fashion to minimize the following integrated total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \lambda \mathcal{L}_{\text{ps}} \tag{14}$$

where λ is a hyper-parameter controlling the contribution of $\hat{V}_u$ to the overall loss $\mathcal{L}_{\text{total}}$.

3 Experiments

3.1 Experimental Setup

We conduct experiments on three datasets (Cora, CiteSeer and PubMed) [Sen et al., 2008]. To ensure a fair comparison, we adhere to the evaluation protocol used in previous studies [Zhao et al., 2021, Yun et al., 2022]. The datasets undergo manual pre-processing to achieve the desired imbalance ratio (ρ). We consider two imbalance ratios ($\rho = 10\%, 5\%$), and two different numbers of minority classes: 3 and 5 on Cora and CiteSeer datasets , and 1 and 2 on PubMed dataset. Additionally, for Cora and CiteSeer datasets, we also adopt a long-tail class label distribution setup similar to [Yun et al., 2022]. Graph Convolution Network (GCN) [Kipf and Welling, 2017] implements the message passing layers in our framework and all the comparison baselines. We compare our Uni-GNN with the GCN baseline, SMOTE [Chawla et al., 2002] and LTE4G [Yun et al., 2022].

3.2 Comparison Results

We evaluate the performance of Uni-GNN framework on the semi-supervised node classification task under class imbalance. Across the three datasets, we explore four distinct evaluation setups by manipulating the number of minority classes and the imbalance ratio (ρ). Additionally, we explore

Table 1: The overall performance on Cora, CiteSeer and PubMed datasets .

	# Min. Class	3						5					
	ρ	10%			5%			10%			5%		
		bAcc.	Macro-F1	G-Means	bAcc.	Macro-F1	G-Means	bAcc.	Macro-F1	G-Means	bAcc.	Macro-F1	G-Means
Cora	GCN	$68.8_{(4.0)}$	$68.8_{(4.0)}$	$80.8_{(2.6)}$	$60.0_{(0.4)}$	$56.6_{(0.7)}$	$74.8_{(0.3)}$	$64.9_{(6.2)}$	$64.7_{(5.7)}$	$78.1_{(4.2)}$	$55.1_{(2.6)}$	$51.4_{(2.4)}$	$71.4_{(1.8)}$
	SMOTE	$65.1_{(4.0)}$	$62.3_{(5.1)}$	$78.3_{(2.7)}$	$59.0_{(2.2)}$	$53.9_{(2.6)}$	$74.2_{(1.5)}$	$60.3_{(7.6)}$	$58.7_{(8.9)}$	$74.9_{(5.3)}$	$49.1_{(4.1)}$	$45.6_{(5.4)}$	$67.0_{(3.1)}$
	LTE4G	$73.2_{(5.4)}$	$72.1_{(6.1)}$	$83.6_{(3.5)}$	$70.9_{(2.5)}$	$69.6_{(2.8)}$	$82.1_{(1.6)}$	$75.4_{(5.6)}$	$75.4_{(5.4)}$	$85.0_{(3.6)}$	$70.2_{(4.5)}$	$\mathbf{68.8}_{(4.7)}$	$81.7_{(3.0)}$
	Uni-GNN	$\mathbf{76.5}_{(0.5)}$	$\mathbf{76.4}_{(0.7)}$	$\mathbf{85.8}_{(0.3)}$	$\mathbf{71.5}_{(1.2)}$	$\mathbf{70.7}_{(1.5)}$	$\mathbf{82.5}_{(0.8)}$	$\mathbf{75.4}_{(3.7)}$	$\mathbf{75.4}_{(3.7)}$	$\mathbf{85.0}_{(2.4)}$	$\mathbf{70.5}_{(3.7)}$	$68.7_{(2.6)}$	$\mathbf{81.8}_{(2.5)}$
CiteSeer	GCN	$49.5_{(2.1)}$	$43.1_{(2.3)}$	$66.7_{(1.5)}$	$48.2_{(0.9)}$	$39.3_{(0.4)}$	$65.7_{(0.7)}$	$48.9_{(1.4)}$	$45.3_{(1.3)}$	$66.2_{(1.1)}$	$42.4_{(6.5)}$	$39.1_{(7.3)}$	$61.1_{(5.1)}$
	SMOTE	$48.7_{(2.5)}$	$40.1_{(1.8)}$	$66.1_{(1.9)}$	$47.8_{(0.8)}$	$38.9_{(1.9)}$	$65.4_{(0.6)}$	$44.9_{(4.4)}$	$41.9_{(4.1)}$	$63.2_{(3.4)}$	$40.1_{(2.0)}$	$34.2_{(1.5)}$	$59.4_{(1.6)}$
	LTE4G	$54.2_{(4.5)}$	$51.8_{(4.1)}$	$70.2_{(3.3)}$	$52.7_{(2.1)}$	$48.3_{(3.7)}$	$69.1_{(1.5)}$	$52.1_{(3.7)}$	$47.2_{(3.6)}$	$68.6_{(2.7)}$	$47.3_{(1.1)}$	$41.2_{(2.1)}$	$65.0_{(0.9)}$
	Uni-GNN	$\mathbf{59.1}_{(3.6)}$	$\mathbf{54.6}_{(3.3)}$	$\mathbf{73.6}_{(2.5)}$	$\mathbf{54.1}_{(3.1)}$	$\mathbf{48.5}_{(4.1)}$	$\mathbf{70.1}_{(2.3)}$	$\mathbf{58.3}_{(2.5)}$	$\mathbf{55.0}_{(1.3)}$	$\mathbf{73.1}_{(1.8)}$	$\mathbf{54.0}_{(2.2)}$	$\mathbf{51.4}_{(2.2)}$	$\mathbf{70.0}_{(1.6)}$
	# Min. Class	1						2					
	ρ	10%			5%			10%			5%		
PubMed	GCN	$60.4_{(6.5)}$	$55.9_{(9.5)}$	$69.6_{(5.2)}$	$58.6_{(3.0)}$	$51.9_{(6.2)}$	$68.1_{(2.4)}$	$49.1_{(10.9)}$	$44.0_{(14.5)}$	$60.3_{(8.9)}$	$41.0_{(5.6)}$	$32.2_{(8.4)}$	$53.7_{(4.7)}$
	SMOTE	$55.8_{(3.0)}$	$48.2_{(3.3)}$	$65.9_{(2.4)}$	$55.2_{(2.8)}$	$46.8_{(2.2)}$	$65.4_{(2.2)}$	$41.8_{(4.0)}$	$32.6_{(6.6)}$	$54.4_{(3.3)}$	$36.6_{(2.1)}$	$23.8_{(4.8)}$	$50.0_{(1.8)}$
	LTE4G	$62.6_{(3.0)}$	$59.2_{(6.7)}$	$71.4_{(2.4)}$	$60.0_{(5.4)}$	$55.3_{(8.2)}$	$69.3_{(4.3)}$	$52.1_{(7.0)}$	$50.2_{(8.3)}$	$62.9_{(5.7)}$	$48.5_{(9.9)}$	$44.3_{(12.2)}$	$48.5_{(9.9)}$
	Uni-GNN	$\mathbf{73.9}_{(2.3)}$	$\mathbf{73.8}_{(2.5)}$	$\mathbf{80.2}_{(1.8)}$	$\mathbf{67.1}_{(4.2)}$	$\mathbf{65.2}_{(5.3)}$	$\mathbf{74.8}_{(3.3)}$	$\mathbf{66.9}_{(4.3)}$	$\mathbf{66.0}_{(4.1)}$	$\mathbf{74.7}_{(3.3)}$	$\mathbf{63.4}_{(6.4)}$	$\mathbf{62.3}_{(8.6)}$	$\mathbf{72.0}_{(5.0)}$

Table 2: The overall performance on Cora-LT and CiteSeer-LT datasets .

	Cora-LT			CiteSeer-LT		
	bAcc.	Macro-F1	G-Means	bAcc.	Macro-F1	G-Means
GCN	$66.8_{(1.1)}$	$65.0_{(1.0)}$	$79.5_{(0.7)}$	$50.4_{(1.4)}$	$45.7_{(0.8)}$	$67.4_{(1.1)}$
SMOTE	$66.8_{(0.4)}$	$66.4_{(0.6)}$	$79.4_{(0.2)}$	$51.2_{(1.9)}$	$46.6_{(1.8)}$	$68.0_{(1.4)}$
LTE4G	$72.2_{(3.1)}$	$72.0_{(2.9)}$	$83.0_{(2.0)}$	$56.4_{(2.1)}$	$52.5_{(2.2)}$	$71.7_{(1.5)}$
Uni-GNN	$\mathbf{75.2}_{(1.3)}$	$74.8_{(1.3)}$	$\mathbf{84.9}_{(0.8)}$	$\mathbf{63.3}_{(1.7)}$	$\mathbf{61.6}_{(2.5)}$	$\mathbf{76.6}_{(1.2)}$

the long-tail class label distribution setting for Cora and CiteSeer with an imbalance ratio of $\rho = 1\%$. We assess the performance of Uni-GNN using balanced Accuracy (bAcc), Macro-F1, and Geometric Means (G-Means), reporting the mean and standard deviation of each metric over 3 runs. Table 1 summarizes the results for different numbers of minority classes and imbalance ratios on all three datasets, while Table 2 showcases the results under long-tail class label distribution on Cora and CiteSeer.

Table 1 illustrates that the performance of all methods diminishes with decreasing imbalance ratio (ρ) and increasing numbers of minority classes. Our proposed framework consistently outperforms the underlying GCN baseline and all other methods across all three datasets and various evaluation setups. The performance gains over the GCN baseline are substantial, exceeding 10% in most cases for Cora and CiteSeer datasets and 13% for most instances of the PubMed dataset. Moreover, Uni-GNN consistently demonstrates superior performance compared to all other comparison methods, achieving notable improvements over the second-best method (LTE4G) by around 3%, 5%, and 11% on Cora, CiteSeer with 3 minority classes and $\rho = 10\%$, and PubMed with 1 minority class and $\rho = 10\%$, respectively. Similarly, Table 2 highlights that Uni-GNN consistently enhances the performance of the underlying GCN baseline, achieving performance gains exceeding 8% and 12% on Cora-LT and CiteSeer-LT datasets, respectively. Furthermore, Uni-GNN demonstrates remarkable performance gains over all other class imbalance methods, surpassing 4% on CiteSeer-LT. These results underscore the superior performance of our framework over existing state-of-the-art class-imbalanced GNN methods across numerous challenging class imbalance scenarios.

4 Conclusion

In this paper, we introduced a novel Uni-GNN framework for class-imbalanced node classification. The proposed framework harnesses the combined strength of structural and semantic connectivity through dedicated structural and semantic node encoders, enabling the learning of a unified node representation. By utilizing these encoders, the structural and semantic connectivity ensures effective propagation of messages beyond the structural immediate neighbors of nodes, thereby addressing the under-reaching and neighborhood memorization problems. Moreover, we proposed a balanced pseudo-label generation mechanism to incorporate confident pseudo-label predictions from minority unlabeled nodes into the training set. Our experimental evaluations on three benchmark datasets for node classification affirm the efficacy of our proposed framework. The results demonstrate that Uni-GNN adeptly mitigates class imbalance bias, surpassing existing state-of-the-art methods in class-imbalanced graph learning.

References

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. In *Journal of Artificial Intelligence Research*, 2002.

D Manning Christopher, Raghavan Prabhakar, and Schutze Hinrich. Introduction to information retrieval, 2008.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Zhongkai Hao, Chengqiang Lu, Zhenya Huang, Hao Wang, Zheyuan Hu, Qi Liu, Enhong Chen, and Cheekong Lee. Asgn: An active semi-supervised graph neural network for molecular property prediction. In *SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Inter. Conference on Learning Representations (ICLR)*, 2017.

Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. Graphsha: Synthesizing harder samples for class-imbalanced node classification. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2023.

Joonhyung Park, Jaeyun Song, and Eunho Yang. Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International Conference on Learning Representations (ICLR)*, 2021.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. In *AI magazine*, 2008.

Qingyun Sun, Jianxin Li, Haonan Yuan, Xingcheng Fu, Hao Peng, Cheng Ji, Qian Li, and Philip S Yu. Position-aware structure learning for graph topology-imbalance by relieving under-reaching and over-squashing. In *International Conference on Information & Knowledge Management (CIKM)*, 2022.

Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Investigating and mitigating degree-related biases in graph convoltuional networks. In *International Conference on Information & Knowledge Management (CIKM)*, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Liang Yan, Shengzhong Zhang, Bisheng Li, Min Zhou, and Zengfeng Huang. Unreal: Unlabeled nodes retrieval and labeling for heavily-imbalanced node classification. In *arXiv preprint arXiv:2303.10371*, 2023.

Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. L2-gcn: Layer-wise and learned efficient training of graph convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Sukwon Yun, Kibum Kim, Kanghoon Yoon, and Chanyoung Park. Lte4g: long-tail experts for graph neural networks. In *International Conference on Information & Knowledge Management (CIKM)*, 2022.

Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng. Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance. In *International Conference on Data Mining (ICDM)*, 2021.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *International conference on web search and data mining (WSDM)*, 2021.

A Implementation Details

The semantic and structural encoders consist of 2 message passing layers each, followed by a Rectified Linear Unit (ReLU) activation function. The node classifier is composed of a single GCN layer, followed by ReLU activation, and then a single fully connected linear layer. Uni-GNN undergoes training using an Adam optimizer with a learning rate of $1e^{-2}$ and weight decay of $5e^{-4}$ over 10,000 epochs. The hyperparameter λ is assigned the value 1. For the hyperparameters K, α, ϵ, and β, we explore the following ranges: $\{3C, 4C, 10C, 20C, 30C\}$, $\{1, 2\}$, $\{0.5, 0.7\}$, and $\{50, 100\}$, respectively. Each experiment is repeated three times, and the reported performance metrics represent the mean and standard deviation across all three runs.

B Computational Complexity Analysis

The computational complexity of the standard GCN message passing layer is $\mathcal{O}(N \cdot D^2 + |E| \cdot D)$ [You et al., 2020]. Our proposed Uni-GNN framework incorporates two GCN-based node encoders: a semantic node encoder and a structural node encoder. For each encoder, we construct dedicated adjacency matrices. The construction of the structural connectivity-based adjacency matrix A_{struct} has a general computational complexity of $\mathcal{O}(N \cdot d_{\text{avg}}^{\alpha})$, where d_{avg} represents the average node degree in G. As for the construction of the semantic connectivity-based adjacency matrices, it requires performing K-means clustering that has a computational complexity of $\mathcal{O}(T \cdot N \cdot K \cdot D)$, where T denotes the number of iterations in K-means [Christopher et al., 2008].

C Compute Resources

The experiments are conducted on GPU servers, where each instance includes 8-Cores Intel CPUs, 32GB of RAM, and a RTX 3060 GPU with 12GB of VRAM.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately reflect the paper's contributions and scope. Experimental results provided in the paper support the claims made in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The main limitation lies in the extra computational complexity, which has been addressed in Section B.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The implementation details are shown in Section A.

   Guidelines:

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

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

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

Answer: [No]

Justification: The datasets we evaluate our method on are publicly available, while we provide the implementation details required to reproduce our experiments in Section A.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The detailed experimental settings are included in Section 3.1.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: The main results are reported with standard deviations.

Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute Resources are described in Section C.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper advances the research of GNN learning, with positive application potentials.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: We do not release new assets, therefore the paper poses no risks of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All the used datasets are properly cited.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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