# NLGT: Neighborhood-based and Label-enhanced Graph Transformer Framework for Node Classification

**Xiaolong Xu[1,2,3,4], Yibo Zhou[1], Haolong Xiang[1,3*], Xiaoyong Li[5*], Xuyun Zhang[6], Lianyong Qi[7], Wanchun Dou[8]**

[1]School of Software, Nanjing University of Information Science and Technology, China
[2]Yunnan Key Laboratory of Service Computing, Yunan University of Finance and Economics, China
[3]Jiangsu Province Engineering Research Center of Advanced Computing and Intelligent Services, China
[4]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, China
[5]College of Meteorology and Oceanography, National University of Defense Technology, China
[6]School of Computing, Macquarie University, Australia
[7]College of Computer Science and Technology, China University of Petroleum (East China), China
[8]State Key Laboratory for Novel Software Technology, Nanjing University, China
njuxlxu@gmail.com, {ybzhou, hlxiang}@nuist.edu.cn, xuyun.zhang@mq.edu.au, lianyongqi@gmail.com

## Abstract

Graph Neural Networks (GNNs) are widely applied on graph-level tasks, such as node classification, link prediction and graph generation. Existing GNNs mostly adopt a message-passing mechanism to aggregate node information with their neighbors, which often makes node information similar after rounds of aggregations and leads to oversmoothing. Although recent works have made improvements by combining different message aggregation methods or introducing semantic encodings as priors, these message-passing based GNNs still fail to combat oversmoothing after multiple iterations of node aggregation. Besides, the feature extraction ability of these methods is restricted because of the graph sparsity that hinders the aggregation of node information. To deal with the above two issues, we propose **N**eighborhood-based and **L**abel-enhanced **G**raph **T**ransformer (NLGT), a novel and effective framework for graph learning. Specifically, we present a label-enhanced feature fusion mechanism that integrate the shallow node features and label embeddings as enhanced features. Moreover, we design a neighborhood-based mask attention mechanism to alleviate the negative effects caused by the sparsity of the graph. In the predicting stage, we aggregate the prediction results from multiple sampled sub-graphs and apply voting mechanisms to enhance the accuracy and robustness of our framework. Finally, extensive experiments are conducted on four open benchmark datasets, which demonstrate the effectiveness and robustness of our proposed framework compared with existing state-of-the-art methods.

## Introduction

Graph structures play a crucial role in various real-life scenarios, such as citation networks, social networks, and biological gene networks (Liu et al. 2024). Graph Neural Networks (GNNs), as dominant techniques for modeling graph structures, have achieved remarkable success on graph-level tasks such as node classification, link prediction and graph generation (Wang et al. 2023). There are a wide range of GNNs, such as Graph Convolutional Networks (GCN) (Kipf and Welling 2017), Graph Sample and Aggregate (GraphSAGE) (Hamilton, Ying, and Leskovec 2017), and Graph Attention Networks (GAT) (Veličković et al. 2018), which can effectively capture information through diverse feature aggregation mechanisms. However, these traditional methods are based on message-passing mechanism, which aggregates the node information from its neighborhood nodes. This process will lead to oversmoothing, i.e., node information converges to a similar state, resulting in a reduction of the node diversity (Oono and Suzuki 2020). In real-world scenarios, graphs are often sparse, which affects the efficiency of information aggregation and increases the difficulty of convergence for these message-passing networks (Yun et al. 2020). Furthermore, message-passing based GNNs apply an iterative mechanism, which is time-consuming and poses challenges for parallel training of the model. Some methods have been proposed to tackle the above problems, but it is still challenging to provide an effective learning framework in the graph domain.

There have been many works that attempted to leverage transformers into the graph domain to mitigate the problems existing in traditional GNNs, which are called Graph Transformers (GTs). For instance, Graphormer (Ying et al. 2021) implements a dense attention mechanism supplemented by centrality and spatial encodings to capture the structural information in graphs. Exphormer (Shirzad et al. 2023) adopts a sparse attention mechanism to make graph transformers more scalable. Polynomial-Expressive Graph Transformer (Polynormer) (Deng, Yue, and Zhang 2024) presents a local-to-global attention mechanism to balance the trade-off between expressivity and scalability of models. These transformer-based graph classification methods utilize self-attention mechanisms to capture long-range dependencies among nodes, which enables a global understanding of the graph structures. Additionally, the parallelization feature of transformers allows for efficient processing and training on large-scale graphs. However, the performance of classification is restricted by only considering shallow node


*Corresponding authors.

features. Moreover, some GTs incorporate spatial or semantic information as priors, which results in high model complexity and poses challenges to training and deployment.

To solve the above problems, we propose a novel Graph Transformer framework called **N**eighborhood-based and **L**abel-enhanced **G**raph **T**ransformer (NLGT, which is an effective neighborhood-based learning framework for graph-level tasks. To make NLGT more powerful and reasonable, we analyse classical GNNs and summarize three basic paradigms for graph learning: 1) Global graph partitioning and sampling; 2) Topological structure learning; 3) Locality information integration. According to these paradigms, we firstly adopt hierarchical neighborhood sampling for each node to reduce the model complexity and serialize the sampled sub-graphs as unordered sequences. Then, we propose a label-enhanced features fusion mechanism, which regards label embeddings as additional feature information in both training and predicting stage. Especially, a learnable $[CLS]$ token is applied as a classification head to predict the label of the center node in the sub-graph. Furthermore, we design a neighborhood-based mask attention mechanism, which considers the correlations among all nodes to enhance the global understanding of the graph structures. In the predicting stage, we aggregate the prediction results from multiple sampled sub-graphs to ultimately determine the label of the center node. Finally, extensive experiments are conducted on four real-world datasets of node classification, which illustrate the outstanding performance of our proposed NLGT framework comparing with the state-of-the-art methods. Ablation studies are conducted to evaluate the effects of each component, which demonstrates the effectiveness and robustness of the proposed NLGT framework.

The main contributions of this paper are summarized as follows:

- We investigate that message-passing based GNNs are susceptible to oversmoothing and always produce poor performance when the graph is sparse. Then, we analyse and summarize three basic paradigms for graph learning.
- We design a label-enhanced features fusion mechanism to enrich the representations of node features and mitigate the negative impact of oversmoothing. Furthermore, the ability of classification is enhanced by utilizing the label representation.
- We propose a novel neighbourhood-based attention mechanism that effectively captures the relationships between nodes, and integrates the topological information of neighboring regions. This approach mitigates the challenges posed by graph sparsity, maintaining a relatively low model complexity, and enables efficient parallelization for enhanced scalability.
- We conduct extensive experiments on four benchmark datasets, which show that our proposed framework consistently achieves outstanding performance on accuracy compared with classical GNNs and state-of-the-art GT models. Ablation studies on different variants further verify the effectiveness and robustness of our proposed framework. Our code of NLGT is available at https://github.com/LemonZyb/NLGT.git.

# Related Work

## Graph Neural Networks

GNNs are dominant techniques in the area of graph learning, which adopt a message-passing mechanism to learn the expressivity of a node by collecting information from its neighbors. Early GNNs include the development of a number of architectures, such as GCN (Kipf and Welling 2017), which leverages the concept of spectral convolution, applying convolutional layers to graph-structured data by utilizing the graph laplacian. GraphSAGE (Hamilton, Ying, and Leskovec 2017) is a scalable framework to address the challenge of large-scale graphs where it is impractical to train on the entire graph, which employs various aggregation functions to efficiently update node representations based on local neighborhood information. GAT (Veličković et al. 2018) introduces the concept of attention mechanisms to graph neural networks, allowing the model to focus selectively on different nodes in a neighborhood when aggregating information. GIN (Xu et al. 2019) confines the GNNs expressivity to the limits of the 1-Weisfeiler-Lehman (1-WL) isomorphism test. Recent works pay more attention to improving the expressiveness of GNNs by proposing many other mechanisms. For instance, Relational Pooling (Murphy et al. 2019) uses a one-hot encoding of the node as additional features that allow nodes to be distinguished. ASGN (Hao et al. 2020) adopts a teacher-student framework to fully utilize labeled and unlabeled graphs. DGN (Beaini et al. 2021) uses Laplacian eigenvectors to define directional flows for anisotropic message aggregation. PC-GNN (Liu et al. 2021) devises a label-balanced sampler to construct the sub-graphs for training. T2-GNN (Huo et al. 2023) proposes a general GNN framework based on teacher-student distillation to improve the performance of GNNs on incomplete graphs. The above methods have made great innovations to enhance their performance for graph-level tasks, however, oversmoothing and graph sparsity remain challenges for these message-passing based GNNs.

## Graph Transformers

Attention mechanisms have achieved remarkable success in sequence modeling since the foundational work on Transformer (Vaswani et al. 2017). In recent years, an increasing number of approaches have adopted attention mechanisms for graph modeling. GAT (Veličković et al. 2018) introduces a novel attention mechanism that enables nodes to effectively aggregate information from their neighbors. Building on this, Graphormer (Ying et al. 2021) implements a dense attention mechanism supplemented by structural features, such as centrality and spatial encodings. Furthermore, GraphGPS (Rampášek et al. 2022) presents a generalized framework that seamlessly integrates message-passing networks with attention mechanisms, which enables the combination of diverse positional and structural embeddings. NAGphormer (Chen et al. 2023) emphasizes the development of sampling-based scalable graph transformers. Recent studies have also focused on sparse graph attention mechanism and the efficiency of computation, such as Exphormer (Shirzad et al. 2023) adopts a sparse attention mechanism

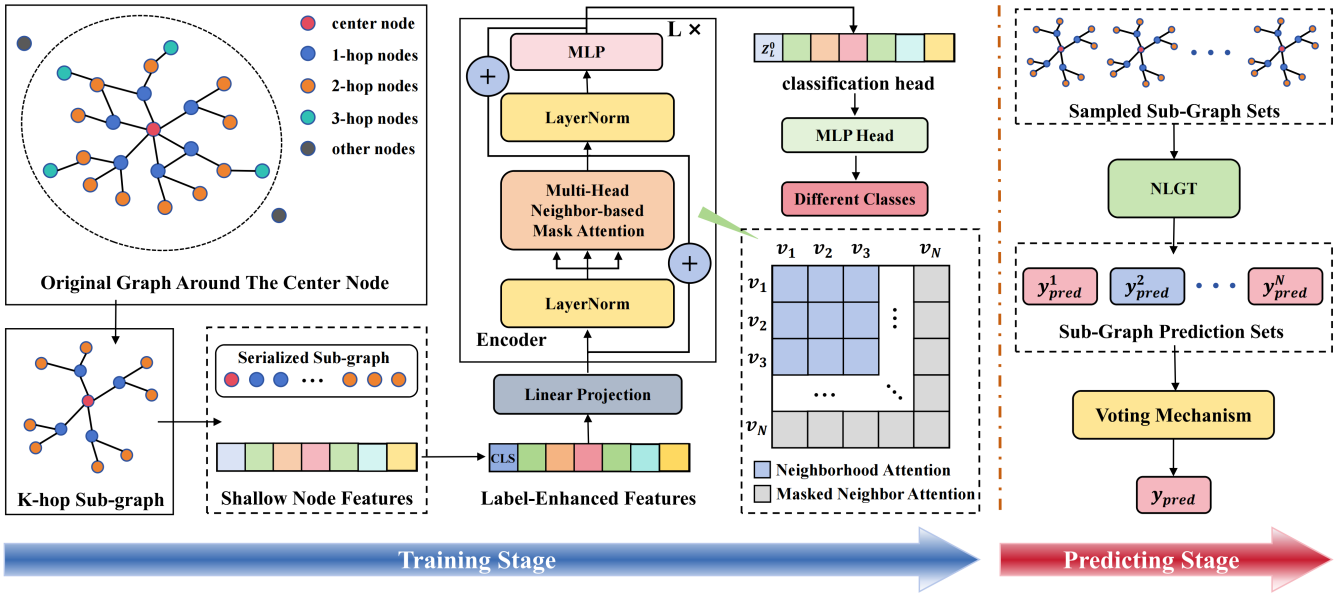



Figure 1: Overview of the NLGT framework.

and Polynormer (Deng, Yue, and Zhang 2024) computes local-to-global attention in linear time. These GTs methods enhance traditional GNNs by improving model expressiveness. However, most of these GTs methods require more intricate components to enhance their performance, which leads to increased model complexity. Furthermore, these methods only consider shallow node features, which restricts the representation of nodes, deteriorating the performance of classification.

Overall, previous message-passing based GNNs struggle with oversmoothing and graph sparsity. Concurrently, recent GT methods suffer from high complexity and limited expressivity. In this paper, we aim to propose a lightweight and effective framework to address these challenges.

## Problem Definition

Node classification problem is one of the most important tasks in graph domain, which has significant applications in real-world scenarios, such as paper citation prediction, social network analysis, and recommendation systems (Xu et al. 2024). The target of node classification problem is to utilize the set of labeled nodes to train a classifier model $C$, which is used to evaluate on those unlabeled nodes from the same graph.

Formally, Given a graph $G(V, E)$, where $V = \{v_1, v_2, \ldots, v_N\}$ represents the $N$ nodes in the graph, and $E = \{e_1, e_2, \ldots, e_M\}$ denotes the $M$ edges of the graph. We assume the problem provides $X = \{x_1, x_2, \ldots, x_N\}^T \in \mathbb{R}^{N \times d}$, which represents the feature matrix of the nodes, where $d$ denotes the dimension of features. Additionally, let $Y = \{y_1, y_2, \ldots, y_N\}^T \in \mathbb{R}^N$, which represents the labels of the nodes . For a given node $v_i$ in the graph, there exists a corresponding feature vector $x_i$ and a label $y_i$.

We denote the entire dataset as $D = \{v_i, x_i, y_i\}, i = 1, \ldots, N$, with the first $K$ nodes used for training as observable nodes and the left for testing. Consequently, we can partition the entire dataset into training set $D_{\text{train}} = \{v_j, x_j, y_j\}, j = 1, \ldots, K$ and test set $D_{\text{test}}$, where $D_{\text{train}} \cup D_{\text{test}} = D$. The task is training a classifier $C$ with parameters $\theta$ using observable nodes in $D_{\text{train}}$ to predict the class labels of the unobservable nodes in $D_{\text{test}}$. The optimization function of classifier $C$ can be formalized as:

$$\arg\max_{\theta} \log p_\theta(\hat{Y}|X) = \sum_{j=1}^{K} \log p_\theta(\hat{y}_j|X). \quad (1)$$

In essence, we aim to maximize the log-probability of the entire output $\hat{Y}$ by maximizing the sum of log-probabilities of individual outputs with the relevant input features $X$.

## NLGT Framework

### Overview

In this part, we present a novel and effective neighborhood-based node classification framework, called NLGT, which is shown in Figure 1. Firstly, we perform hierarchical neighborhood sampling for each node in the graph to control the model complexity. Then, the sampled sub-graphs are serialized as sequences, and we design a label-enhanced features fusion mechanism, which generates label embeddings as additional features to enhance the information expression of nodes. Next, these enhanced features are trained by our designed mask attention mechanism based on neighborhood

Algorithm 1: **NLGT**: **N**eighborhood-based and **L**abel-enhanced **G**raph **T**ransformer

**Input**: $G(V, E)$, observable nodes set $V_l$; node features $X$; node labels $Y$; recycling times $L$; objective function $J(\cdot)$; NLGT encoder $E(\cdot)$; projection function $F(\cdot)$.

1: Initialize $\theta, \phi, \psi$.
2: **for** $v \in V_l$ **do**
3: obtain sub-graph $g$ by neighborhood sampling.
4: $Y_i^{(0)} \leftarrow \begin{cases} Y_{cls}, & i = 0 \\ Y_i^{(0)}, & otherwise \end{cases}$
5: $X^{(0)} \leftarrow F_\phi(X^{(0)} \oplus Y^{(0)})$
6: **for** $l = 1$ to $L$ **do**
7: $\hat{X}^{(l)} \leftarrow E_\theta(\hat{X}^{(l-1)})$
8: **end for**
9: $\hat{Y} \leftarrow F_\psi(\hat{X}_0^{(L)})$
10: $J_\theta(\hat{Y}, Y).backward()$
11: Update $\theta, \phi, \psi$ via back propagation.
12: **end for**

information, which can guarantee the predicted node effectively capture the correlations with its neighbors. In the node prediction stage, we adopt a voting mechanism to enhance the classification accuracy and robustness of our proposed framework. The process of our framework is illustrated in Algorithm 1, detailed with the following subsections.

## Hierarchical Neighborhood Sampling

Previous GNNs use global topological information between nodes for feature representation learning, which is inefficient on large-scale graphs and consumes lots of computing resources. GraghSAGE and many other methods adopt neighborhood sampling and aggregation based on message-passing mechanism to capture the partial structure information in the graph. Based on these works, we adopt hierarchical neighborhood sampling for each node in the original graph, which is presented in line 3 of Algorithm 1. Specifically, assuming that the number of sampling layers is $K$, for each layer $k$, neighborhood sampling is performed on each node $v_i$ in graph $G(V, E)$, and the sampling number is $s_{ik}$, so for node $v_i$, the total number of sampled neighbors $S_i$ can be formalized as:

$$S_i = \sum_{k=1}^{K} s_{ik}, \tag{2}$$

where $S_i \ll |V|$. In this way, we can control the complexity of the model by setting the scale of sampling. Compared with the previous sampling methods based on message-passing mechanism, as we adopt the fully-connected attention method in following designs, which guarantee that we can focus on part of the global structural information and train the model in parallel.

## Graph Serialization

After applying neighborhood sampling for nodes, we serialize the sampled sub-graphs into sequences. Each node in the sequence is treated as a token, and the center nodes of sub-graphs are placed at the beginning of each sequence. The difference with general NLP problem is that the sequence of nodes is disordered, so we don't consider Positional Encoding (PE) commonly used in NLP problems. The node classification problem after graph serialization translates to predicting the label of the first node token in each sub-graph sequence.

## Label-Enhanced Features Fusion

Shallow features of nodes like skip-gram (Mikolov et al. 2013) are commonly used for modeling in node classification problems. (He et al. 2024) points out these shallow node features are limited in the complexity of the semantic features, which restricts the performance of classification. Some studies incorporate extra structural or semantic encodings as priors to enhance the expressivity of models, however, most of these methods increase the model complexity. Label Propagation Algorithm (LPA) (Garza and Schaeffer 2019) is a popular method in community detection (Fortunato 2010), which assumes that the nodes are more similar within the group, however, it lacks consideration of the characteristics of the nodes themselves. Based on the above works, we present a label-enhanced features fusion mechanism, which regards label embeddings as additional information to enhance the feature representation of nodes without increasing the model complexity, which is presented in lines 4 and 5 in Algorithm 1. Specifically, for a K-hop sampled and serialized sub-graph $G'(V', E')$, the node feature matrix is $X \in \mathbb{R}^{|V'| \times d}$ and the node category is $Y \in \mathbb{R}^{|V'|}$. At the beginning of the training stage, we will embed the node label into dimension $d_l$, then we obtain the node label embedding matrix $Y_{emb} \in \mathbb{R}^{|V'| \times d_l}$.

Since our goal is to predict the label of the first node in the sequence, the label of the center node should be unobservable. Similar to the $[class]$ token in BERT (Devlin et al. 2019) and ViT (Dosovitskiy et al. 2021), we apply a learnable $[CLS]$ token as the classification head to replace the first place of the $Y_{emb}$ matrix. So the $Y_{emb} = \left\{Y_{cls}^1, Y_{emb}^2, \ldots, Y_{emb}^{|V'|}\right\}$, then, we add it with the node's own original shallow feature $X$, and we get:

$$X_{enhanced} = X \oplus Y_{emb}. \tag{3}$$

So the $l_{th}$ output of the NLGT encoder $Z_l$ can be formalized as:

$$Z_l = NLGT_l\left(X_{enhanced}\right). \tag{4}$$

Finally, we extract the classification head which is attached to $Z_l^0$, and the classification head is implemented by a simple MLP layer, so the final prediction can be formalized as:

$$Y_{predict} = MLP\left(Z_l^0\right). \tag{5}$$

## Neighborhood-Based Mask Attention Mechanism

After serializing the sub-graph and obtaining the label-enhanced node features, we consider designing the encoder of our NLGT framework for further modeling. In general NLP tasks, we usually set a maximum length for sequences, which means when an input sequence exceeds this maximum length, it should be truncated; conversely, when the input length falls short, it should be padded using a

$[PAD]$ token. We perform similar operations on the serialized sub-graph by setting a maximum number of nodes, which denoted as $N_{max}$. Assume a sub-graph sequence $S = \{v_1, v_2, \ldots, v_n\}$, where $n$ represents the length of the sequence and $n < N_{max}$. Then we apply a Padding operation to the sequence $S$, and we get the padded sequence $S_{pad} = \{v_1, v_2, \ldots, v_n, \ldots, v_{pad}, v_{pad}\}$. For the $[PAD]$ node, a masking method is typically employed to prevent attention calculated between irrelevant nodes by setting their attention coefficients to $-\infty$. For example, attention in GAT is masked based on the adjacency matrix of the graph. However, this mask method based only on local adjacency matrix, which limits the model's understanding of the global structure especially when the graph is sparse. Building upon this, we design a multi-head neighborhood-based mask attention mechanism which takes the interdependencies into account between each node in the subgraph and enhances the understanding of spatial global structural information. Besides, we mask the irrelevant information with those $[PAD]$ nodes comparing with fully-connected attention (Kreuzer et al. 2021). So for a padded sequence $S_{pad} \in \mathbb{R}^{N_{max}}$, which has a node feature matrix $X \in \mathbb{R}^{N_{max} \times d}$, where $d$ represents the embedding dimension of the features. For simplicity of illustration, we just consider the single-head attention computation, so we calculate $Q, K, V$ matrix as:

$$Q = X \cdot W_Q, K = X \cdot W_K, V = X \cdot W_V, \tag{6}$$

where $W_Q \in \mathbb{R}^{d \times d}$, $W_K \in \mathbb{R}^{d \times d}$, and $W_V \in \mathbb{R}^{d \times d}$ are projection matrix for $X$ corresponding to the representations of $Q, K, V$. Then, we further capture the correlations between $Q$ and $K$ as:

$$A = \frac{Q \cdot K^T}{\sqrt{d}}. \tag{7}$$

Since we only focus on the attention among the first $n_{th}$ non-PAD nodes, we apply a masking operation to the matrix $A$, which only retains the attention coefficients for the first $n \times n$ positions in $A$, and the remaining attention values are set to $-\infty$. Finally, we we get $A_{mask}$, and calculate the attention matrix as:

$$Attn(X) = Softmax(A_{mask}) \cdot V. \tag{8}$$

The extension to the multi-head attention is standard and straightforward. Our NLGT Encoder consists of alternating layers of multi-head neigborhood-based mask attention (MHNMSA) (Equ. (6)-(8)) and MLP block that containing two layers with a GELU non-linearity. LayerNorm (LN) is applied before every block, and residual connections after every block (Wang et al. 2019). Particularly, all bias terms are omitted for simplicity in our proposed framework. So for a $L$ layers NLGT encoder, the $l_{th}$ layer output can be formalized as:

$$H'(l) = MHNMSA(LN(H(l-1))) + H(l-1), \tag{9}$$

$$H(l) = MLP(LN(H'(l))) + H'(l). \tag{10}$$

In Algorithm 1, the procedure of NLGT encoder is presented in lines 6-9.

Through our proposed neighborhood-based mask attention mechanism, nodes can learn the correlations among each other, which further enhances the model's understanding of global structural information in the sub-graph even when the graph is sparse, ultimately improving the performance of node classification.

### Prediction Voting Mechanism

In the predicting stage, to overcome the effects of randomness caused by just considering the prediction result on a single sub-graph, we aggregate the results of multiple sampled sub-graphs of the same center node. In this way, we get the prediction set $Y_{pred} = \left\{y^0_{pred}, \ldots y^1_{pred}, \ldots y^T_{pred}\right\}$, where $y^i_{pred}$ is the prediction result of $i_{th}$ sampled sub-graph of the center node, and $T$ denotes to the total number of sampled sub-graphs. By analogizing the voting method for classification that commonly used in Ensemble Learning, we present a voting mechanism $Voting(\cdot)$ on the results of multiple sub-graphs and aggregate them to get the final prediction result $y_{pred}$ of the center node, which can be formalized as:

$$y_{pred} = Voting(Y_{pred}). \tag{11}$$

The most frequently occurring prediction result from multiple sub-graphs is selected as the final output. This process can effectively enhance the accuracy and robustness of our proposed NLGT.

## Experiments

### Experimental Setup

**Datasets.** We use four real-world commonly used benchmark datasets of different scales for model evaluation Three citation network datasets, Cora, Citeseer, and Pubmed (Sen et al. 2008) are chosen, we also select a relatively large-scale dataset Ogbn-Arxiv from OGB (Hu et al. 2021). Some statistics of these datasets are presented in Table 1. PyTorch Geometric (PYG) (Fey and Lenssen 2019) is mainly used for most of structured operations in the experiments.

| Datasets | #Nodes | #Edges | #Features | #Classes |
|---|---|---|---|---|
| Cora | 2,708 | 5,278 | 1,433 | 7 |
| Citeseer | 3,327 | 4,676 | 3,703 | 6 |
| Pubmed | 19,717 | 44,327 | 500 | 3 |
| Ogbn-Arxiv | 169,343 | 1,166,243 | 128 | 40 |

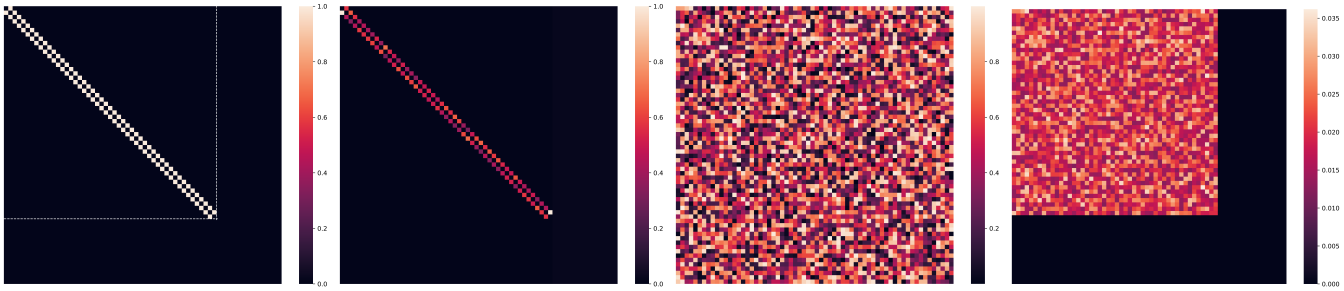
Table 1: Statistics of datasets.

**Baselines.**

- GCN (Kipf and Welling 2017): GCN leverages the concept of spectral convolution, and applies convolutional layers by utilizing the graph laplacian.
- GraphSAGE (Hamilton, Ying, and Leskovec 2017): GraphSAGE employs various aggregation functions to efficiently update node representations based on local neighborhood information.
- GAT (Veličković et al. 2018): GAT introduces the concept of attention mechanisms to graph neural networks,

| Datasets / Methods | Cora | | Citeseer | | Pubmed | | Ogbn-Arxiv | |
|---|---|---|---|---|---|---|---|---|
| **Metric** | Test.Acc | Valid.Acc | Test.Acc | Valid.Acc | Test.Acc | Valid.Acc | Test.Acc | Valid.Acc |
| GCN (2017) | 82.16 ± 0.64 | 82.50 ± 0.51 | 69.74 ± 0.85 | 70.25 ± 0.67 | 79.67 ± 0.68 | 80.46 ± 0.62 | 71.54 ± 0.46 | 71.78 ± 0.42 |
| GraphSAGE (2017) | 82.95 ± 0.27 | 83.14 ± 0.30 | 70.98 ± 0.26 | 71.23 ± 0.24 | 84.16 ± 0.25 | 84.08 ± 0.29 | 72.85 ± 0.16 | 72.96 ± 0.21 |
| GAT (2018) | 83.02 ± 0.35 | 83.11 ± 0.41 | 70.15 ± 0.63 | 71.09 ± 0.55 | 82.49 ± 0.52 | 82.70 ± 0.45 | 72.46 ± 0.31 | 72.83 ± 0.36 |
| GraphGPS (2022) | 83.06 ± 0.61 | 83.27 ± 0.55 | 72.21 ± 0.47 | 72.32 ± 0.39 | 82.76 ± 0.28 | 83.15 ± 0.31 | 70.97 ± 0.41 | 71.13 ± 0.36 |
| Exphormer (2023) | 83.34 ± 0.22 | 83.77 ± 0.25 | 72.61 ± 0.12 | 73.26 ± 0.17 | 83.36 ± 0.17 | 83.87 ± 0.15 | 72.44 ± 0.28 | 73.31 ± 0.26 |
| HEAL (2024) | 83.47 ± 0.17 | 83.72 ± 0.23 | 72.76 ± 0.20 | 73.14 ± 0.15 | 83.57 ± 0.15 | 84.26 ± 0.19 | 73.35 ± 0.13 | 73.68 ± 0.12 |
| Polynormer (2024) | 83.29 ± 0.24 | 83.50 ± 0.28 | 72.57 ± 0.28 | 72.76 ± 0.16 | 83.63 ± 0.09 | 83.91 ± 0.13 | 73.36 ± 0.11 | 73.65 ± 0.14 |
| **NLGT(Ours)** | **84.03 ± 0.15** | **84.45 ± 0.16** | **73.15 ± 0.18** | **73.67 ± 0.15** | **84.59 ± 0.12** | **85.05 ± 0.10** | **73.96 ± 0.08** | **74.21 ± 0.11** |

Table 2: Accuracy(%) comparison on different benchmark datasets. The best results are in bold and the second best is underlined "-". Our proposed NLGT outperforms other methods on all datasets.



(a) Adjacency Matrix of Sparse Sub-graph

(b) Adjacency Matrix-based Attention

(c) Fully-connected Attention Without Mask

(d) Neighborhood-based Masked Attention in NLGT

Figure 2: Visualization of Different Attention Mechanisms for Sparse Graphs

allowing the model to focus selectively on different nodes in a neighborhood when aggregating information.

- GraphGPS (Rampášek et al. 2022): GraphGPS presents a generalized framework that integrates message-passing networks with attention mechanisms.
- Exphormer (Shirzad et al. 2023): Exphormer adopts a sparse attention mechanism based on virtual global nodes and expander graphs.
- HEAL (Ju et al. 2024): HEAL explores the higher-order relationships among nodes to learn complex node dependencies beyond pair-wise relations.
- Polynormer (Deng, Yue, and Zhang 2024): Polynormer adopts a linear local-to-global attention scheme to learn high-degree equivariant polynomials whose coefficients are controlled by attention scores.

**Implementation Details.** The sampling specification is set as $[10, 5, 2]$, indicating the number of 1-hop, 2-hop, and 3-hop neighbors of each node, and the maximum of nodes $N_{max}$ is 168. Both the label embedding dimension and the $[CLS]$ embedding dimension are 128, with the same dimension applied to the $[PAD]$ embeddings and linear projection. NLGT consists of 4 layers and 2 attention heads, which enhances the ability of feature extraction. The inner-layer dimension of the feed-forward network is 512. The dropout ratio is 0.1, which is employed to mitigate over-fitting. SGD momentum is configured at 0.9 to facilitate better convergence, and weight decay is set to $5 \times 10^{-5}$ as a regularization parameter. The batch size is defined as 64 and the learning rate is $1 \times 10^{-3}$, which control the step size during optimization. Finally, NLGT utilizes prediction results from 10 subgraphs for the voting mechanism and allows for a maximum of 10 training epochs. All experiments are implemented by PyTorch, and are trained on two 24GB RTX 3090 GPU.

**Metric.** We adopt the widely used Test.Accuracy and Valid.Accuracy to measure the performance on node classification of different methods (Ju et al. 2024; Deng, Yue, and Zhang 2024). A higher value indicates better accuracy performance. All experiments are run 10 times and averaged results are reported.

## Results Analysis

**Node Classification.** Table 2 summarizes performance of NLGT comparing with other Graph Networks on Cora, Citeseer, Pubmed and Ogbn-Arxiv datasets. Especially, GraphGPS, Exphormer, HEAL, and Polynormer are recently proposed transformer-based models in graph representation learning. Generally, NLGT outperforms previous state-of-the-art Graph Networks on all datasets, and the ad-

vanced conclusions are summarized as follows.

Firstly, comparing with previous message-passing mechanism based GNNs, our proposed NLGT outperforms on all tasks, especially on large-scale graphs by a larger margin, which indicates the effectiveness of our framework and its extensibility on large-scale graphs. Secondly, comparing with recent GT based models, NLGT outperforms on all tasks as well. Especially, NLGT only adopts label-enhanced features fusion for expressive representation without introducing additional spatial or semantic encodings as prior information like other methods, which demonstrates the simplicity and scalability of our NLGT framework. Lastly, NLGT has a relatively smaller disturbance of accuracy by adopting voting mechanism to further enhance the accuracy and the robustness.

**Attention Mechanisms.** To further investigate whether the neighborhood-based attention mechanism in NLGT can alleviate the learning difficulties caused by graph sparsity, we conduct extra experiments using three different attention mechanisms, as visualized in Figure 2. Three NLGT variant models based on different attention mechanisms are denoted as: 1) NLGT-A, representing attention based on adjacency relations; 2) NLGT-F, representing full attention that does not mask the $[PAD]$ nodes; 3) NLGT-M, representing the neighborhood-based masked attention used in this paper.

The experimental results shown in Table 3 demonstrate that the proposed neighborhood-based masked attention mechanism outperforms the others in node classification on sparse graphs. Specifically, for a sparse sampled sub-graph, using an adjacency-matrix based mask attention mechanism that only focuses on the relationships between adjacent nodes, neglecting the global spatial and semantic information in the graph, which limits the expressivity of model. Additionally, as the sub-graph becomes sparser, the amount of irrelevant information increases more. Applying a fully-connected attention mechanism without masking the the irrelevant information with those $[PAD]$ tokens will introduces lots of semantic noise into the updates of node features and lead to the decline of model performance.

Finally, our proposed neighborhood-based mask attention mechanism combines the advantages and disadvantages of the first two mechanisms, which is based on a fully-connected attention mechanism, and it ignores the correlations with the $[PAD]$ nodes. For sparse sub-graphs, this approach captures the long-range dependencies in sub-graph sequences more effectively, which enhances the global understanding of graphs and ignores extra irrelevant information with the $[PAD]$ nodes.

| Methods \ Datasets | Cora | Citeseer | Pubmed | Arxiv |
|---|---|---|---|---|
| NLGT-A | 83.12 ± 0.24 | 71.13 ± 0.35 | 83.41 ± 0.25 | 72.67 ± 0.26 |
| NLGT-F | 83.41 ± 0.38 | 72.45 ± 0.33 | 83.59 ± 0.22 | 73.41 ± 0.21 |
| NLGT-M | **84.03 ± 0.15** | **73.15 ± 0.18** | **84.59 ± 0.12** | **73.96 ± 0.08** |

Table 3: Accuracy (%) comparison of different attention mechanisms on different datasets. NLGT-M with neighborhood-based masked attention achieves the best performance.

| Neighbor-Based Attention | Label-Enhanced Features | Voting Mechanism | Test Accuracy(%) |
|---|---|---|---|
| - | - | - | 71.63 ± 0.26 |
| - | ✓ | - | 72.17 ± 0.23 |
| - | - | ✓ | 71.93 ± 0.21 |
| ✓ | - | - | 72.68 ± 0.18 |
| ✓ | ✓ | - | 73.28 ± 0.09 |
| ✓ | - | ✓ | 72.97 ± 0.16 |
| ✓ | ✓ | ✓ | **73.96 ± 0.08** |

Table 4: Ablation study results on Ogbn-Arxiv dataset with different variants of NLGT. Performance will increase as different components are incorporated.

## Ablation Study

We compare NLGT with several variants on Ogbn-Arxiv dataset to verify the effectiveness of its framework designs. The ablation results are included in Table 4. For fair comparison, we set the same model hyper-parameters in section. Implementation Details for all ablation experiments.

From the experimental results in Table 4, we can draw the following conclusions: (1) NLGT outperforms other variants, indicating the effectiveness of our proposed NLGT framework. (2) Adopting neighborhood-based attention mask yields a large margin performance boost in comparison to those variants without using mask mechanism, which indicates that when calculating the attention matrix, we need to filter out information between irrelevant nodes, and enhance the inter-correlations among relevant nodes. (3) In the case of adopting neighborhood-based attention mask mechanism, introducing label-enhanced features as additional information achieves better performance, which indicates that the label embeddings of neighborhood nodes hold significant reference for the label prediction of the center node, especially in cases where we only have shallow features of the nodes. (4) Applying voting mechanism in the predicting stage also improves the performance, which indicates that the integration of results from multiple sub-graphs is more robust and accurate than only relying on a single sub-graph.

## Conclusion

In this paper, we investigate that previous message-passing based GNNs fail to handle with oversmoothing and graph sparsity, while recent GT methods have restrictions in the model complexity and expressivity. To deal with the existing problems, we propose a novel framework called NLGT. Extensive experiments illustrate that our proposed framework achieves outstanding performance, compared with existing state-of-the-art methods on popular benchmark datasets.

In the future, we aim to explore the following problems on how to apply NLGT to a broader range of heterogeneous graph learning; how to define a more reasonable hierarchical sampling scale considering the diverse characteristics of different graph data.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 42050102, Grant No. 92267104 and Grant No. 62372242, and Jiangsu Provincial Major Project on Basic Research of Cutting-edge and Leading Technologies, under grant No. BK20232032. This research was also supported by the Dou Wanchun Expert Workstation of Yunnan Province No.202105AF150013. The authors wish to acknowledge Dr. Fei Dai, Professor of Southwest Forestry University, for his help in interpreting the significance of the results of this study

## References

Beaini, D.; Passaro, S.; Létourneau, V.; Hamilton, W.; Corso, G.; and Lió, P. 2021. Directional Graph Networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 748–758.

Chen, J.; Gao, K.; Li, G.; and He, K. 2023. NAGphormer: A Tokenized Graph Transformer for Node Classification in Large Graphs. In *11th International Conference on Learning Representations*.

Deng, C.; Yue, Z.; and Zhang, Z. 2024. Polynormer: Polynomial-Expressive Graph Transformer in Linear Time. In *12th International Conference on Learning Representations*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Fortunato, S. 2010. Community detection in graphs. *Physics Reports*, 486(3): 75–174.

Garza, S. E.; and Schaeffer, S. E. 2019. Community detection with the Label Propagation Algorithm: A survey. *Physica A: Statistical Mechanics and its Applications*, 534: 122058.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025–1035.

Hao, Z.; Lu, C.; Huang, Z.; Wang, H.; Hu, Z.; Liu, Q.; Chen, E.; and Lee, C. 2020. ASGN: An Active Semi-supervised Graph Neural Network for Molecular Property Prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 731–752.

He, X.; Bresson, X.; Laurent, T.; Perold, A.; LeCun, Y.; and Hooi, B. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. In *12th International Conference on Learning Representations*.

Hu, W.; Fey, M.; Ren, H.; Nakata, M.; Dong, Y.; and Leskovec, J. 2021. OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Huo, C.; Jin, D.; Li, Y.; He, D.; Yang, Y.; and Wu, L. 2023. T2-GNN: Graph Neural Networks for Graphs with Incomplete Features and Structure via Teacher-Student Distillation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, 4339–4346.

Ju, W.; Mao, Z.; Yi, S.; Qin, Y.; Gu, Y.; Xiao, Z.; Wang, Y.; Luo, X.; and Zhang, M. 2024. Hypergraph-enhanced Dual Semi-supervised Graph Classification. In *Forty-first International Conference on Machine Learning*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Kreuzer, D.; Beaini, D.; Hamilton, W.; Létourneau, V.; and Tossou, P. 2021. Rethinking Graph Transformers with Spectral Attention. In *Advances in Neural Information Processing Systems*, volume 34, 21618–21629.

Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2021. Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the Web Conference 2021*, 3168–3177.

Liu, Z.; Wang, C.; Feng, H.; and Chen, Z. 2024. Efficient Unsupervised Graph Embedding with Attributed Graph Reduction and Dual-level Loss. *IEEE Transactions on Knowledge and Data Engineering*, 1–14.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 3111–3119.

Murphy, R.; Srinivasan, B.; Rao, V.; and Ribeiro, B. 2019. Relational Pooling for Graph Representations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 4663–4673.

Oono, K.; and Suzuki, T. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *8th International Conference on Learning Representations*.

Rampášek, L.; Galkin, M.; Dwivedi, V. P.; Luu, A. T.; Wolf, G.; and Beaini, D. 2022. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35: 14501–14515.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Mag.*, 29(3): 93–106.

Shirzad, H.; Velingker, A.; Venkatachalam, B.; Sutherland, D. J.; and Sinop, A. K. 2023. Exphormer: Sparse Transformers for Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, 31613–31632.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, Q.; Li, B.; Xiao, T.; Zhu, J.; Li, C.; Wong, D. F.; and Chao, L. S. 2019. Learning Deep Transformer Models for Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1810–1822.

Wang, X.; Dong, Y.; Jin, D.; Li, Y.; Wang, L.; and Dang, J. 2023. Augmenting Affective Dependency Graph via Iterative Incongruity Graph Learning for Sarcasm Detection. In *AAAI Conference on Artificial Intelligence*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Xu, X.; Li, C.; Xiang, H.; Qi, L.; Zhang, X.; and Dou, W. 2024. Attention based document-level relation extraction with none class ranking loss. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*.

Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; and Liu, T.-Y. 2021. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34: 28877–28888.

Yun, C.; Chang, Y.-W.; Bhojanapalli, S.; Rawat, A. S.; Reddi, S.; and Kumar, S. 2020. O (n) connections are expressive enough: Universal approximability of sparse transformers. *Advances in Neural Information Processing Systems*, 33: 13783–13794.