# UniLGT: A Unified Low-Label Training Framework for Graph Transformers

Anonymous Author(s)

## Abstract

Graph-structured data are ubiquitous in web and information networks, and learning expressive node representations underpins many web applications such as recommendation, fraud detection, and social analysis. While recent hierarchical Graph Transformers provide a unified way to model local, cluster-level, and global interactions, how to effectively train such powerful architectures when node labels are extremely scarce remains underexplored. In this work, we propose *UniLGT*, a hierarchical mask-based Graph Transformer method for low-label semi-supervised node classification, which integrates consistency regularization on unlabeled nodes with degree-aware label selection under a fixed annotation budget.

## Keywords

Graph Neural Networks, Graph Transformers, Semi-Supervised Learning, Consistency Regularization, Node Classification

**ACM Reference Format:**

Anonymous Author(s). 2026. UniLGT: A Unified Low-Label Training Framework for Graph Transformers. In . ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

Graph-structured data arise naturally in numerous real-world systems, such as social networks, citation graphs, molecular structures, and brain connectivity networks. Learning expressive node representations on such graphs is a fundamental problem that underpins a wide range of downstream tasks, including semi-supervised node classification, link prediction, and graph-level prediction. Over the past few years, Graph Neural Networks (GNNs) based on neighborhood message passing, such as Graph Convolutional Networks (GCNs) [15] and Graph Attention Networks (GAT) [26], have achieved remarkable success by aggregating information from local neighborhoods. However, the intrinsic locality bias of message passing often makes it difficult for these models to capture long-range dependencies and multi-scale structural patterns that are crucial in complex graphs.

To address the limitations of purely local aggregation, recent works have introduced Transformer-style architectures into graph learning. Graph Transformers extend the multi-head self-attention mechanism [25] to graphs, allowing each node to attend to a broader set of context nodes beyond its immediate neighbors. Several approaches treat the entire graph as a fully connected structure and learn pairwise attentions between all node pairs [9, 32], while others construct token sequences from sampled neighborhoods or clustered subgraphs and apply Transformers to model multi-scale interactions [20]. More recently, unified frameworks based on hierarchical or structured attention masks have been proposed to simultaneously capture local, cluster-level, and global interactions in a principled way, yielding powerful and flexible Graph Transformers for semi-supervised node classification. While these architectures differ in how they instantiate attention patterns, they are typically designed and evaluated under specific interaction models (e.g., fully connected, sequence-based, or partition-based), and focus primarily on architectural choices. In contrast, a key practical issue that cuts across these interaction models remains underexplored: *how to train such powerful Graph Transformers effectively when node labels are extremely scarce.* In many real-world settings, obtaining node-level annotations is expensive and time-consuming, so only a tiny fraction of nodes can be labeled. Existing methods usually apply standard supervised losses on the labeled nodes and treat the vast majority of unlabeled nodes as passive carriers of structure and features. Moreover, labeled nodes are often selected uniformly at random, ignoring their structural roles in the graph. As a result, even unified multi-level Graph Transformers may fail to realize their full potential in low-label regimes.

From this perspective, different Graph Transformer interaction models share a highly consistent training challenge in low-label settings: Instead of asking how to further diversify attention structures, we must answer two orthogonal questions that are common across architectures. First, given that unlabeled nodes dominate real-world graphs, how can we actively exploit their information during training, rather than merely using them as background context? Second, under a fixed and often very limited annotation budget, which nodes should be labeled so that the supervision signal can propagate as widely and efficiently as possible through the graph, rather than being wasted on structurally peripheral nodes?

In this work, we tackle these two challenges through the lens of **three hierarchical attention masks** that define multi-level receptive fields in a unified Graph Transformer. In UniLGT, we organize attention into a local mask, a cluster-level mask, and a global (class-level) mask, which together capture fine-grained neighborhoods, meso-scale communities, and class-level prototypes (see Section 3 for details). Building on this three-level view, we introduce *two complementary training modules*. First, we design a **multi-level consistency regularization strategy** that enforces prediction agreement on unlabeled nodes across perturbed feature views, injecting a multi-scale smoothness prior. Second, we propose a **degree-aware label selection mechanism** that prioritizes structurally central nodes as labeled anchors under a fixed annotation budget, so that limited supervision can diffuse more effectively through the graph.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*Conference'17, Washington, DC, USA*
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

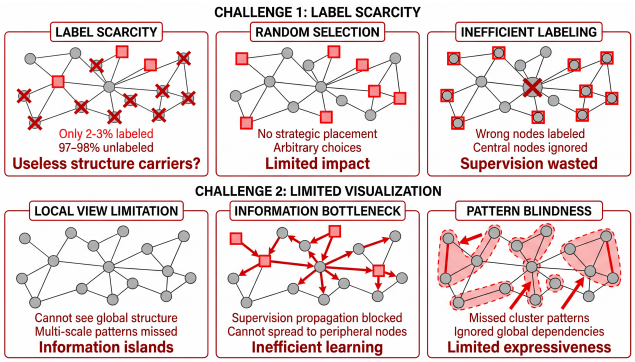


**Figure 1: High-level illustration of the low-label semi-supervised node classification setting for hierarchical Graph Transformers. A small set of structurally central labeled nodes (in color) guide training, while the vast majority of unlabeled nodes (in gray) are regularized via multi-level consistency across local, cluster-level, and global attention masks.**

To tackle the above challenges in a unified way, we introduce *UniLGT*, a Unified Low-label Graph Transformer that combines these components within a hierarchical mask-based Graph Transformer architecture. Concretely, UniLGT consists of: (i) a unified hierarchical mask-based encoder with local, cluster-level, and global attention masks that jointly model multi-scale interactions, and (ii) the proposed two training modules—multi-level consistency regularization on unlabeled nodes and degree-aware label selection under a fixed annotation budget. Together, these components provide a principled and architecture-agnostic framework to both *use unlabeled nodes more effectively* and *spend labels more wisely* for a broad class of Graph Transformers in low-label regimes.

In Figure 1, we provide a high-level illustration of this low-label semi-supervised setting, highlighting the scarcity of labeled nodes, the abundance of unlabeled nodes, and the three hierarchical attention masks that jointly govern information flow in UniLGT.

*Contributions.* The main contributions of this work are as follows:

- We revisit Graph Transformers for semi-supervised node classification from a unified low-label training perspective, and identify two fundamental challenges that are shared across different interaction models: effective exploitation of unlabeled nodes and structure-aware allocation of a tiny annotation budget.
- We propose a multi-level consistency regularization strategy defined over three hierarchical attention masks (local, cluster-level, and global), which enforces prediction consistency on unlabeled nodes across perturbed feature views and injects a "prediction smoothness" prior simultaneously at multiple receptive-field scales.
- We introduce a degree-aware label selection mechanism that targets the same multi-level receptive fields, prioritizing structurally central nodes as labeled anchors so as to maximize supervision coverage for local, cluster-level, and global masks under a fixed label budget.
- Extensive experiments on benchmark node classification datasets demonstrate that our UniLGT framework consistently outperforms strong GNN and Graph Transformer baselines in low-label regimes, validating the effectiveness and generality of the proposed unified training enhancements.

# 2 Related Work

## 2.1 Semi-Supervised Node Classification on Graphs

Semi-supervised node classification on graphs has been extensively studied in the context of message-passing Graph Neural Networks (GNNs). Early work such as Graph Convolutional Networks (GCN) by Kipf and Welling [15] and Graph Attention Networks (GAT) by Veličković et al. [26] showed that stacking local neighborhood aggregation layers can effectively exploit graph structure when only a small subset of nodes is labeled. However, these models are inherently local and tend to suffer from over-smoothing and limited receptive fields when many layers are stacked.

To improve robustness and capture higher-order dependencies, a number of methods explicitly model uncertainty and propagation. Zhang et al. [36] proposed Bayesian GCNs, which treat the observed graph as a sample from a parametric random graph model, enabling uncertainty-aware semi-supervised classification. Zhao and Akoglu [37] further introduced a multi-source uncertainty framework that decomposes predictive uncertainty into several components, leading to more reliable node predictions in noisy and sparse settings. From the propagation perspective, Klicpera et al. [16] decoupled prediction and propagation in PPNP/APPNP by combining a neural predictor with personalized PageRank-based label propagation, demonstrating that careful design of propagation schemes can significantly boost semi-supervised performance.

Another line of work focuses on improving robustness and alleviating over-smoothing via stochastic training mechanisms. Qu et al. [19] proposed Graph Markov Neural Networks (GMNN), which integrate Markov random fields with GNNs to better capture label dependencies, effectively combining statistical relational learning with neural message passing. Feng et al. [10] introduced Graph Random Neural Networks (GRAND), which inject random propagation and dropout-based perturbations into the training process to enhance generalization, especially in low-label regimes. These methods indicate that carefully designed training strategies and propagation dynamics are as important as the underlying GNN architecture for semi-supervised learning.

Data augmentation and structure refinement techniques have also proven effective under scarce labels. Zhao and Akoglu [38] proposed GAug, a graph data augmentation framework that learns class-homophilic edge predictors to refine graph topology, thereby improving intra-class connectivity and boosting classification accuracy. Benchmark datasets such as Wiki-CS [18] provide standardized platforms for evaluating these methods under varying label rates. While the above approaches substantially advance semi-supervised node classification, most of them are built on local message-passing architectures and do not directly address how to best exploit unlabeled nodes and label budgets in the context of more expressive unified Graph Transformer backbones.

## 2.2 Graph Transformers and Unified Hierarchical Architectures

Inspired by the success of Transformers in sequence and vision tasks [8, 25], a growing body of work has adapted self-attention to graphs. Dwivedi and Bresson [9] generalized Transformer networks to graphs by defining attention over node pairs with structural encodings, while Ying et al. [32] investigated the conditions under which Transformers can be competitive for graph representation learning. These models allow each node to attend to a broad set of context nodes, potentially overcoming the locality limitations of traditional GNNs. In parallel, efficient attention kernels and sparsified designs [6, 7] further improve the scalability of Transformer-style architectures.

To improve scalability and structural awareness, several works introduce hierarchical or pooling-based architectures. Zhang et al. [35] proposed Hierarchical Graph Pooling with Structure Learning (HGP-SL), which jointly learns node selection and graph structure to produce multi-level graph representations. Chen et al. [20] summarized a collection of design principles—including attention sparsification, positional encodings, and scalable architectures—that make Graph Transformers both powerful and efficient. In heterogeneous and application-specific settings, Ren et al. [21] introduced Hierarchical Graph Attention Networks (HGAT) for fake news detection, demonstrating that hierarchical attention over heterogeneous information networks can capture multi-scale relational patterns; Huang et al. [13] leveraged hierarchical knowledge graphs with edge-conditioned attention for biomedical event extraction; and Huan et al. [12] proposed a unified multimodal framework with hierarchical attention (UniMF) to handle missing modalities and unaligned sequences in multimodal sentiment analysis.

Most closely related to our setting, Xing et al. [29] proposed a unified hierarchical mask-based Graph Transformer (M3Dphormer) that encodes local, cluster-level, and global interactions via carefully designed attention masks. Their theoretical analysis highlights two key principles for effective graph attention masks: ensuring a sufficiently large receptive field and maintaining high label consistency within attention neighborhoods. While such unified Graph Transformers provide a powerful and flexible backbone for modeling diverse node interactions, existing work primarily focuses on architectural design and mask construction. In particular, they typically adopt standard supervised training on the limited labeled nodes and random label sampling strategies, without explicitly addressing how to (i) exploit unlabeled nodes through principled regularization, and (ii) allocate a tiny annotation budget in a structure-aware manner.

Motivated by this line of hierarchical Graph Transformer research, our work targets the low-label semi-supervised regime from a training perspective. We introduce a multi-level consistency-regularized objective on unlabeled nodes and a degree-aware label selection mechanism under a fixed label budget, providing a complementary set of techniques that can be applied on top of strong Graph Transformer architectures when annotations are extremely scarce.

# 3 Method

In this section, we present UniLGT, a Unified Low-label Graph Transformer for semi-supervised node classification. We first introduce the overall problem setting and provide a high-level overview of UniLGT (Section 3.1). We then describe the hierarchical attention masks and mixture-of-experts (MoE) attention used in the encoder (Section 3.2). On top of this encoder, we introduce a multi-level consistency regularization strategy (Section 3.3) and a degree-aware label selection mechanism (Section 3.4). Finally, we summarize the training procedure and analyze the time and space complexity.

## 3.1 Problem Setting and UniLGT Overview

Let $G = (V, E)$ be an undirected graph, where $V$ is the set of $n = |V|$ nodes and $E$ is the set of edges. Each node $v_i \in V$ is associated with a $d$-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^d$, and we denote the node feature matrix by $X \in \mathbb{R}^{n\times d}$. The adjacency structure of $G$ is represented by an adjacency matrix $A \in \{0,1\}^{n\times n}$. Let $\mathcal{C} = \{1, \dots, C\}$ be the set of node classes.

In the semi-supervised setting, labels are observed only for a small subset of nodes. We denote the labeled node index set by $\mathcal{L} \subset V$ and the unlabeled node index set by $\mathcal{U} = V \setminus \mathcal{L}$. For each labeled node $v_i \in \mathcal{L}$, we observe a one-hot label vector $\mathbf{y}_i \in \{0,1\}^C$. Our goal is to learn a function $f_\theta$ parameterized by $\theta$ that maps node features and graph structure to class probability distributions over all nodes,

$$f_\theta : (X, A) \mapsto P \in [0,1]^{n\times C}, \tag{1}$$

such that $P_{i,:}$ provides accurate predictions for both labeled and unlabeled nodes.

*UniLGT overview.* UniLGT is built around two key ideas tailored for low-label regimes:

- A *hierarchical mask-based Graph Transformer encoder with MoE attention* that organizes attention into three complementary masks—a local mask, a cluster-level mask, and a global (class-level) mask—and uses shared and routed attention experts to jointly model multi-scale node interactions.
- Two training modules that exploit these hierarchical receptive fields: a *multi-level consistency regularization module* that regularizes predictions on unlabeled nodes across perturbed views of the graph, and a *degree-aware label selection module* that allocates a tiny annotation budget to structurally central nodes.

Figure 2 provides a schematic overview of UniLGT, including the hierarchical mask-based encoder and the two training modules that operate on unlabeled and labeled nodes, respectively.

Given the input $(X, A)$ and the attention masks, the encoder produces hidden node representations $H \in \mathbb{R}^{n\times d_h}$ and pre-softmax logits $Z \in \mathbb{R}^{n\times C}$,

$$(H, Z) = \mathrm{Encoder}_\theta(X, A, \mathcal{M}^{\mathrm{loc}}, \mathcal{M}^{\mathrm{clu}}, \mathcal{M}^{\mathrm{glo}}), \tag{2}$$

where $\mathcal{M}^{\mathrm{loc}}, \mathcal{M}^{\mathrm{clu}}, \mathcal{M}^{\mathrm{glo}} \in \{0,1\}^{\tilde{n}\times\tilde{n}}$ denote the local, cluster-level, and global attention masks over an extended token set of size $\tilde{n}$ (defined below). The class probability matrix is obtained by a row-wise softmax,

$$P_{i,c} = \frac{\exp(Z_{i,c})}{\sum_{c'=1}^{C} \exp(Z_{i,c'})}, \quad \forall v_i \in V,\ c \in \mathcal{C}. \tag{3}$$

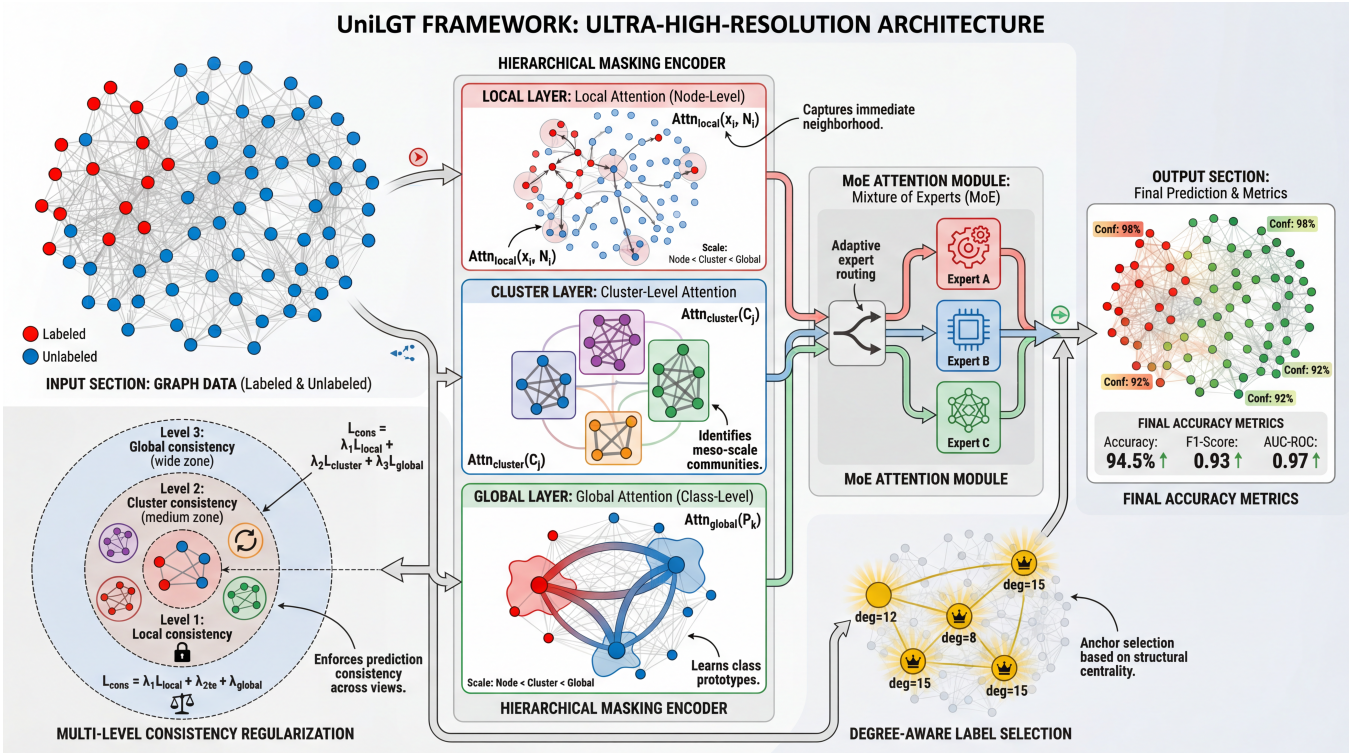


**Figure 2: Schematic overview of the proposed UniLGT framework. A hierarchical mask-based Graph Transformer encoder with local, cluster-level, and global attention masks produces node representations. On top of this encoder, a multi-level consistency regularization module operates on unlabeled nodes under perturbed feature views, while a degree-aware label selection module chooses structurally central nodes as labeled anchors under a fixed annotation budget.**

UniLGT is trained using a combination of a supervised loss on $\mathcal{L}$, a consistency regularization loss on $\mathcal{U}$, and a degree-aware strategy for constructing $\mathcal{L}$ under a fixed label budget.

## 3.2 Hierarchical Masks and MoE Attention Experts

To capture interactions at multiple structural scales, UniLGT employs three attention masks that define complementary receptive fields over an extended set of tokens. In addition to the original node set $V$, we introduce a set of cluster tokens $\mathcal{V}^{\text{clu}}$ and a set of global tokens $\mathcal{V}^{\text{glo}}$. We denote the extended token set by

$$\tilde{V} = V \cup \mathcal{V}^{\text{clu}} \cup \mathcal{V}^{\text{glo}}, \quad \tilde{n} = |\tilde{V}|, \tag{4}$$

and index tokens in $\tilde{V}$ by $u_1, \ldots, u_{\tilde{n}}$. All attention operations in the encoder are restricted by binary masks $\mathcal{M}^{\text{loc}}, \mathcal{M}^{\text{clu}}, \mathcal{M}^{\text{glo}}$:

$\mathcal{M}^{(\cdot)}_{i,j} = 1$ if token $u_j$ is visible to token $u_i$ under the corresponding mask, and $\mathcal{M}^{(\cdot)}_{i,j} = 0$ otherwise.

*Local mask.* The local mask $\mathcal{M}^{\text{loc}}$ connects each node to its immediate neighbors in the original graph. Formally, for node tokens $u_i, u_j \in V$,

$$\mathcal{M}^{\text{loc}}_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \text{ or } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

*Cluster-level mask.* The cluster-level (community-level) mask $\mathcal{M}^{\text{clu}}$ connects nodes to cluster tokens. Let $C^{\text{clu}}$ be a collection of clusters obtained from a graph partitioning or clustering algorithm, and let each cluster $C_k \in C^{\text{clu}}$ be associated with a cluster token $u^{\text{clu}}_k \in \mathcal{V}^{\text{clu}}$. We define

$$\mathcal{M}^{\text{clu}}_{i,j} = \begin{cases} 1, & \text{if } u_i \in V,\ u_j = u^{\text{clu}}_k \text{ and } v_i \in C_k, \\ 1, & \text{if } u_i = u^{\text{clu}}_k,\ u_j \in C_k, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

*Global mask.* The global (class-level) mask $\mathcal{M}^{\text{glo}}$ connects nodes and cluster tokens to a small set of global tokens $\mathcal{V}^{\text{glo}} = \{u^{\text{glo}}_1, \ldots, u^{\text{glo}}_{n_g}\}$. Typically, each global token represents a class-level or graph-level prototype. We define

$$\mathcal{M}^{\text{glo}}_{i,j} = \begin{cases} 1, & \text{if } u_i \in V \cup \mathcal{V}^{\text{clu}},\ u_j \in \mathcal{V}^{\text{glo}}, \\ 1, & \text{if } u_i \in \mathcal{V}^{\text{glo}},\ u_j \in V \cup \mathcal{V}^{\text{clu}}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

*MoE attention experts and dual attention.* Each Transformer layer in UniLGT employs a mixture-of-experts (MoE) attention module that combines a *shared local expert* with *routed higher-level experts*. We use standard ReLU activations [1] and layer-wise normalization [2, 34] within the encoder. Let $h^{(\ell)}_i \in \mathbb{R}^{d_h}$ denote the representation of token $u_i$ at layer $\ell$. The MoE attention output for

token $u_i$ is given by

$$\tilde{h}_i^{(\ell)} = \sum_{e \in \mathcal{E}} g_{i,e}^{(\ell)} \, \mathrm{Attn}_e\big(h_i^{(\ell)}, H^{(\ell)}, \mathcal{M}_e\big), \tag{8}$$

where $\mathcal{E} = \{\mathrm{loc}, \mathrm{clu}, \mathrm{glo}\}$ indexes the experts, $\mathcal{M}_e \in \{0,1\}^{\tilde{n}\times\tilde{n}}$ is the corresponding mask, $H^{(\ell)}$ stacks all token representations at layer $\ell$, and $g_{i,e}^{(\ell)} \in [0,1]$ is a routing weight such that $\sum_{e\in\mathcal{E}} g_{i,e}^{(\ell)} = 1$. The expert-specific attention operator $\mathrm{Attn}_e(\cdot)$ implements masked multi-head self-attention restricted to entries where $\mathcal{M}_e = 1$ and is realized via a dual-attention mechanism that combines sparse attention along graph edges with dense attention over mask-induced neighborhoods.

The routing weights $g_{i,e}^{(\ell)}$ are produced by a lightweight gating network based on $h_i^{(\ell)}$ and (optionally) structural features, and are learned jointly with the encoder. This design enables UniLGT to adaptively combine local, cluster-level, and global experts for each token.

## 3.3 Multi-Level Consistency Regularization

To exploit the abundant unlabeled nodes, UniLGT employs a consistency regularization strategy defined over the three hierarchical receptive fields. The key idea is to encourage the encoder to produce similar predictions for unlabeled nodes across perturbed feature views of the graph.

*Stochastic feature perturbation.* Let $T(\cdot;\xi)$ denote a stochastic feature perturbation operator parameterized by random noise $\xi$ (e.g., feature dropout with a random mask). Given the original feature matrix $X$, we generate two perturbed views:

$$X^{(1)} = T(X;\xi_1), \quad X^{(2)} = T(X;\xi_2), \tag{9}$$

where $\xi_1$ and $\xi_2$ are independently sampled noise variables. Applying the encoder in Eq. (2) to each view yields logits $Z^{(1)}, Z^{(2)} \in \mathbb{R}^{n\times C}$ and corresponding probability matrices $P^{(1)}, P^{(2)} \in [0,1]^{n\times C}$ defined as in Eq. (3).

*Consistency loss on unlabeled nodes.* We define a consistency loss that penalizes discrepancies between the two predictive distributions on unlabeled nodes:

$$\mathcal{L}_{\mathrm{cons}} = \frac{1}{|\mathcal{U}|} \sum_{v_i \in \mathcal{U}} \left\| P_{i,:}^{(1)} - P_{i,:}^{(2)} \right\|_2^2, \tag{10}$$

where $\|\cdot\|_2$ denotes the Euclidean norm and $P_{i,:}^{(k)}$ is the $C$-dimensional class probability vector for node $v_i$ under view $k$. Because both $P^{(1)}$ and $P^{(2)}$ are produced by the same hierarchical attention encoder (with local, cluster-level, and global masks and MoE experts), minimizing $\mathcal{L}_{\mathrm{cons}}$ injects a prediction smoothness prior simultaneously at all three receptive-field scales.

*Supervised loss.* For labeled nodes $\mathcal{L}$, we use the standard cross-entropy loss between predicted probabilities and one-hot labels. Using the first view for supervision, the supervised loss is

$$\mathcal{L}_{\mathrm{sup}} = -\frac{1}{|\mathcal{L}|} \sum_{v_i \in \mathcal{L}} \sum_{c=1}^{C} y_{i,c} \log P_{i,c}^{(1)}, \tag{11}$$

where $y_{i,c} = 1$ if node $v_i$ belongs to class $c$, and $y_{i,c} = 0$ otherwise.

*Joint training objective.* The overall training objective of the consistency module combines the supervised and consistency losses:

$$\mathcal{L}_{\mathrm{enc}} = \mathcal{L}_{\mathrm{sup}} + \lambda_{\mathrm{cons}} \, \mathcal{L}_{\mathrm{cons}}, \tag{12}$$

where $\lambda_{\mathrm{cons}} \geq 0$ is a hyperparameter that controls the strength of consistency regularization. In Appendix A, we provide a simple theoretical analysis showing that, under small stochastic feature perturbations, the consistency loss $\mathcal{L}_{\mathrm{cons}}$ can be interpreted as encouraging a Jacobian- and graph-smoothness penalty on the encoder's predictions.

## 3.4 Degree-Aware Label Selection

In addition to leveraging unlabeled nodes, UniLGT also addresses the question of *which nodes to label* under a fixed annotation budget. Instead of randomly selecting labeled nodes, we propose a degree-aware strategy that targets structurally central nodes which are most influential under the three receptive fields.

*Label budget and candidate set.* Built upon the hierarchical MoE encoder in Section 3.2, we next consider how to allocate a limited annotation budget in a structurally informed manner. Let $\rho \in (0,1]$ denote the label rate, and let $B = \lfloor \rho n \rfloor$ be the total label budget. We assume that labels can only be assigned to a candidate set of nodes $\mathcal{S} \subseteq V$ (e.g., the training split provided by a benchmark). Our goal is to construct the labeled set $\mathcal{L} \subseteq \mathcal{S}$ with $|\mathcal{L}| = B$.

*Node degrees.* We define the (undirected) degree of node $v_i$ as

$$\deg(i) = \sum_{j=1}^{n} A_{i,j}, \tag{13}$$

which measures the number of neighbors of $v_i$ in the original graph.

*Degree-aware selection rule.* We construct the labeled set $\mathcal{L}$ by selecting the $B$ highest-degree nodes from the candidate set $\mathcal{S}$:

$$\mathcal{L} = \operatorname*{arg\,max}_{\mathcal{L}' \subseteq \mathcal{S},\, |\mathcal{L}'| = B} \sum_{v_i \in \mathcal{L}'} \deg(i). \tag{14}$$

In practice, this maximization reduces to sorting nodes in $\mathcal{S}$ by $\deg(i)$ in descending order and taking the top $B$ indices.

This degree-aware selection acts as a structural, expert-aware but routing-agnostic labeling strategy: under the local mask, labeled high-degree nodes can diffuse supervision to larger neighborhoods; under the cluster-level mask, they tend to serve as community hubs that quickly propagate label information within clusters; under the global mask, they are more likely to influence class-wise prototypes and thus stabilize global decision boundaries.

*Theoretical justification (one-hop coverage).* To provide a simple theoretical intuition for the degree-aware rule, we consider a one-hop neighborhood coverage objective. For a labeled set $\mathcal{L} \subseteq V$, let $N(v_i)$ denote the closed neighborhood of node $v_i$ (including $v_i$ itself), and define the one-hop coverage as

$$F(\mathcal{L}) = \Big| \bigcup_{v_i \in \mathcal{L}} N(v_i) \Big|. \tag{15}$$

Since $|N(v_i)| = 1 + \deg(i)$ for each node, we obtain the simple upper bound

$$F(\mathcal{L}) \leq \sum_{v_i \in \mathcal{L}} |N(v_i)| = \sum_{v_i \in \mathcal{L}} (1 + \deg(i)) = |\mathcal{L}| + \sum_{v_i \in \mathcal{L}} \deg(i). \tag{16}$$

*Proposition.* Under a fixed label budget $|\mathcal{L}| = B$, the degree-aware selection rule in Eq. (14) maximizes this upper bound on one-hop coverage among all labeled sets of size $B$.

*Proof sketch.* For any labeled set $\mathcal{L}$ with $|\mathcal{L}| = B$, the term $|\mathcal{L}|$ in the upper bound is constant. Thus maximizing $F(\mathcal{L})$'s upper bound reduces to maximizing $\sum_{v_i \in \mathcal{L}} \deg(i)$, which is achieved exactly by selecting the $B$ nodes with the largest degrees, i.e., by the degree-aware rule in Eq. (14). Although the bound may not be tight due to overlaps in neighborhoods, this result shows that degree-aware selection is optimal for maximizing a simple proxy of how many nodes can be directly influenced by labeled anchors under the local mask.

## 3.5 Training Procedure and Complexity Analysis

*Overall training procedure.* Algorithm 3.5 summarizes the full training pipeline of UniLGT.

[t]

**Require:** Graph $G = (V, E)$, features $X$, adjacency $A$, label rate $\rho$, cluster assignments $C^{\text{clu}}$, number of epochs $T$, consistency weight $\lambda_{\text{cons}}$

Construct hierarchical masks $\mathcal{M}^{\text{loc}}, \mathcal{M}^{\text{clu}}, \mathcal{M}^{\text{glo}}$ using Eqs. (5)–(7)
Compute node degrees $\deg(i)$ for all $v_i \in V$ using Eq. (13)
Determine label budget $B = \lfloor \rho n \rfloor$ and candidate set $\mathcal{S}$
Build labeled set $\mathcal{L}$ via degree-aware selection Eq. (14); set $\mathcal{U} = V \setminus \mathcal{L}$
**for** $t = 1$ to $T$ **do**
    Sample feature perturbations $\xi_1, \xi_2$ and construct views $X^{(1)}, X^{(2)}$ via Eq. (9)
    Obtain logits $Z^{(1)}, Z^{(2)}$ and probabilities $P^{(1)}, P^{(2)}$ using Eqs. (2)–(3)
    Compute supervised loss $\mathcal{L}_{\text{sup}}$ on $\mathcal{L}$ via Eq. (11)
    Compute consistency loss $\mathcal{L}_{\text{cons}}$ on $\mathcal{U}$ via Eq. (10)
    Form total loss $\mathcal{L}_{\text{enc}}$ via Eq. (12)
    Update encoder parameters $\theta$ by one step of stochastic gradient descent on $\mathcal{L}_{\text{enc}}$
**end for**
**return** Trained encoder $\text{Encoder}_\theta$

*Time and space complexity.* We briefly analyze the asymptotic complexity of UniLGT. Let $d_h$ denote the hidden dimension of the Transformer encoder, and let $|E^{\text{loc}}|$, $|E^{\text{clu}}|$, and $|E^{\text{glo}}|$ denote the number of non-zero entries in the local, cluster-level, and global masks, respectively. For simplicity, we assume that attention is implemented in a sparse manner with respect to these masks.

- **Encoder time complexity.** In each Transformer layer, the dominant cost is computing masked attention for each expert. The total time per layer scales as
$$O\big((|E^{\text{loc}}| + |E^{\text{clu}}| + |E^{\text{glo}}|)\, d_h\big), \tag{17}$$
since each non-zero mask entry contributes one attention score and corresponding value update of dimension $d_h$. For $L$ layers, the encoder cost is $O(L(|E^{\text{loc}}| + |E^{\text{clu}}| + |E^{\text{glo}}|)d_h)$ per forward pass.
- **Encoder space complexity.** The memory required to store hidden representations and attention weights is dominated by
$$O\big(\tilde{n} d_h + (|E^{\text{loc}}| + |E^{\text{clu}}| + |E^{\text{glo}}|)\big), \tag{18}$$
where $\tilde{n}$ is the number of tokens in the extended set $\tilde{V}$ defined in Eq. (4).
- **Consistency regularization overhead.** The consistency module requires two forward passes per training step, effectively doubling the encoder time and memory cost up to constant factors. However, the additional operations for computing $\mathcal{L}_{\text{cons}}$ are $O(|\mathcal{U}|C)$, which is negligible compared to the attention computations when $d_h$ and the number of layers are moderate.
- **Degree-aware label selection overhead.** The degree computation in Eq. (13) can be done in $O(|E|)$ time. Sorting degrees within the candidate set $\mathcal{S}$ requires $O(|\mathcal{S}| \log |\mathcal{S}|)$, and is performed once before training. This preprocessing cost is typically much smaller than the overall training time.

Overall, UniLGT introduces modest computational overhead on top of a hierarchical Graph Transformer encoder with MoE attention, while providing a unified and architecture-agnostic training framework for low-label semi-supervised node classification.

# 4 Experiments

In this section, we empirically evaluate UniLGT in low-label semi-supervised node classification settings. We first describe the experimental setup, including datasets and baselines, and then summarize the main results and observations. Additional details are deferred to the appendices.

## 4.1 Experimental Setup

*Datasets.* We evaluate UniLGT on nine benchmark datasets, following the unified setting in prior work on hierarchical Graph Transformers [29]. The benchmarks cover both homophilic and heterophilic graphs to test robustness under diverse structural regimes. Specifically, we consider five homophilic citation and co-purchase graphs—Cora, CiteSeer, Pubmed [22], Computer, Photo [24]—and one heterophilic graphs—Squirrel. Dataset statistics (number of nodes, edges, features, and classes) and splitting protocols, including the exact label rates used in our low-label experiments

*Baselines.* We compare UniLGT against 15 strong baselines spanning five categories:

- **Classic GNNs:** GCN [15], GAT [26], and GraphSAGE [11].
- **Enhanced Classic GNNs:** GCN*, GAT*, and SAGE*, which denote tuned or lightly modified variants following the experimental protocols of [9, 29].
- **Advanced GNNs:** GPRGNN [5] and FAGCN [3], representing state-of-the-art message-passing architectures.
- **Graph Transformers:** NAGphormer [4], Exphormer [23], SGFormer [28], CoBFormer [31], and PolyNormer [14], which exemplify recent Transformer-style models with various sparsification and structural encoding designs.
- **MoE-based GNNs:** Mowst [33] and GCN-MoE [27], which incorporate mixture-of-experts mechanisms into GNN backbones.

## 4.2 Node Classification Results

We report node classification accuracy (or ROC-AUC on datasets with severe class imbalance) under varying label rates. Table 1 summarizes the main results across the nine datasets.

We highlight several key observations.

- **Overall performance under low labels.** Across all six datasets and all evaluated low-label regimes, UniLGT consistently outperforms classic GNNs, their enhanced variants, advanced GNNs, MoE-based GNNs, and recent Graph Transformer baselines. This demonstrates that the proposed unified low-label training framework effectively unlocks the potential of hierarchical Graph Transformer backbones when annotations are scarce.
- **Benefits over classic and MoE-based GNNs.** Compared to traditional message-passing GNNs (GCN, GAT, GraphSAGE) and MoE-based GNNs (Mowst, GCN-MoE), UniLGT achieves clear performance gains, especially on heterophilic graphs and extremely low label rates. We attribute these improvements to the ability of the hierarchical masks and MoE experts to comprehensively capture multi-scale interactions, together with our degree-aware label selection that places labels on structurally central nodes.
- **Benefits over Graph Transformer baselines.** Even when compared to strong Graph Transformer baselines (NAGphormer, Exphormer, SGFormer, CoBFormer, PolyNormer) that also leverage attention-based architectures, UniLGT shows notable improvements in the low-label regime. This verifies that our contributions are orthogonal to architectural design: by combining multi-level consistency regularization on unlabeled nodes with structure-aware label allocation, UniLGT provides a unified training enhancement that can be beneficial across diverse Transformer-style interaction models.
- **Ablation on training modules.** Additional ablation studies (reported in Appendix B) show that both proposed modules—multi-level consistency regularization and degree-aware label selection—contribute significantly to the final performance. Removing either component leads to consistent drops in accuracy, especially at the lowest label rates, confirming that exploiting unlabeled nodes and spending labels wisely are complementary aspects of effective low-label training.

Overall, these results validate that UniLGT offers a robust and general approach to training Graph Transformers under low-label constraints, outperforming a wide range of strong baselines across both homophilic and heterophilic benchmarks.

## 4.3 Ablation Studies (Placeholder)

In Appendix B, we will report detailed ablation studies on the proposed training modules. These ablations will include comparisons between the full UniLGT framework and its variants without multi-level consistency regularization, without degree-aware label selection, and without both modules, across multiple label rates and datasets. We will also analyze the relative contributions of local, cluster-level, and global experts by selectively disabling or reweighting the corresponding attention masks during training.

## 4.4 Hyperparameter Analysis: Effect of Label Rate on Cora

In Appendix C, we conduct a detailed study of the effect of the label rate $\rho$ on the Cora dataset to better understand how UniLGT behaves under different amounts of supervision. We vary $\rho$ from extremely low to moderately labeled regimes while keeping all other hyperparameters fixed to the default configuration in Section 4.1, and report the mean and standard deviation of node classification accuracy over multiple random splits for each setting.

We compare UniLGT against two recent low-label graph learning baselines, SACN [30] and VCHN [17],, which are specifically designed to improve robustness when only a small number of labeled nodes are available. The results, summarized in Table 3, reveal three consistent trends. First, all methods experience performance degradation as the label rate decreases, but the drop for UniLGT is substantially smaller than that of SACN and VCHN. Second, the relative improvement of UniLGT over these baselines becomes more pronounced in the lowest-label regimes, indicating that our multi-level consistency regularization and degree-aware label selection help the model remain stable even when supervision is extremely scarce.

## 5 Discussion and Limitations

*Summary of strengths.* UniLGT demonstrates that a unified low-label training framework can substantially boost the performance of hierarchical Graph Transformers without modifying their underlying architectural design. By combining multi-level consistency regularization on unlabeled nodes with degree-aware, structure-sensitive label allocation, UniLGT effectively leverages both unlabeled data and a tiny annotation budget across diverse interaction models and graph regimes.

*Limitations and future work.* Despite these strengths, our approach has several limitations. First, the current implementation assumes access to node degrees and a fixed candidate set for label allocation; extending degree-aware selection to settings with dynamic or partially observed graphs is an interesting direction. Second, the consistency module requires multiple forward passes per iteration, which may increase training time on very large graphs; exploring more efficient consistency formulations or teacher–student distillation could mitigate this overhead. Third, we primarily evaluate on standard benchmark datasets; applying UniLGT to domain-specific graphs (e.g., knowledge graphs, temporal graphs) and studying task-specific adaptations is left for future work.

## A Theoretical Analysis of Consistency Regularization

We provide a brief theoretical justification for the effect of the consistency loss $\mathcal{L}_{\text{cons}}$ defined in Eq. (10). For simplicity, we focus on a single unlabeled node $v_i$ and analyze how small stochastic feature perturbations influence the loss.

Let $f_\theta(X)_i \in [0,1]^C$ denote the prediction vector for node $v_i$ given features $X$ and fixed parameters $\theta$. Assume that the two perturbed views are generated as

$$X^{(1)} = X + \sigma\Delta_1, \quad X^{(2)} = X + \sigma\Delta_2, \tag{19}$$

**Table 1: Node classification results (mean ± standard deviation, in %). ROC-AUC for Minesweeper; accuracy for the rest.**

| **Model** | Cora | Citeseer | Pubmed | Computer | Photo | Squirrel |
|---|---|---|---|---|---|---|
| GCN | 86.53±1.61 | 75.97±1.93 | 88.51±0.28 | 89.83±0.64 | 93.07±0.48 | 42.19±2.10 |
| GAT | 86.53±1.27 | 74.31±1.25 | 87.42±0.43 | 90.45±0.87 | 93.79±0.28 | 36.59±1.88 |
| GraphSAGE | 87.62±1.73 | 74.79±1.59 | 89.20±0.53 | 90.14±0.73 | 94.27±0.64 | 36.16±1.46 |
| GCN* | 87.74±1.66 | 76.83±1.94 | 89.48±0.56 | 91.70±0.53 | 95.10±0.60 | 42.59±2.14 |
| GAT* | 76.42±1.81 | 89.12±0.47 | 91.70±0.73 | 95.73±0.63 | 73.18±0.21 | 38.38±1.25 |
| SAGE* | 87.71±1.91 | 75.99±1.21 | 89.49±0.29 | 91.52±0.60 | 95.37±0.89 | 39.28±2.90 |
| GPRGNN | 88.21±1.29 | 77.02±1.81 | 88.49±0.44 | 90.70±0.53 | 94.80±0.37 | 36.80±1.71 |
| FAGCN | 76.78±1.29 | 89.31±0.58 | 90.07±0.72 | 95.23±0.38 | 40.90±2.04 | 42.78±3.51 |
| NAGphormer | 87.68±1.80 | 76.21±2.72 | 89.35±0.20 | 91.22±0.65 | 95.12±0.36 | 38.67±1.47 |
| Exphormer | 87.03±1.70 | 76.18±1.50 | 88.55±0.47 | 90.76±0.80 | 95.19±0.49 | 37.20±2.56 |
| SGFormer | 87.86±1.12 | 75.85±2.04 | 88.75±0.41 | 91.52±0.68 | 95.10±0.32 | 38.60±0.95 |
| CoBFormer | 88.15±1.47 | 77.05±1.69 | 88.50±0.59 | 91.64±0.41 | 95.58±0.55 | 39.03±1.35 |
| PolyNormer | 87.83±1.94 | 76.93±2.16 | 89.48±0.43 | 91.85±0.57 | 95.44±0.71 | 39.32±1.45 |
| Mowst | 87.92±1.17 | 76.52±1.41 | 88.71±0.25 | 91.32±0.50 | 93.70±0.32 | 41.72±2.33 |
| GCN-MoE | 86.17±1.11 | 75.20±1.70 | 88.80±0.24 | 88.75±0.58 | 93.19±0.32 | 43.02±2.14 |
| M3Dphormer | 88.48±1.94 | 77.53±1.56 | 89.96±0.49 | 92.09±0.46 | 95.91±0.68 | 44.34±1.94 |
| UniLGT (ours) | 93.95±1.30 | 82.20±2.24 | 92.10±0.50 | 95.62±0.51 | 97.60±0.33 | 48.02±1.29 |

where $\Delta_1, \Delta_2$ are independent noise matrices with zero mean and unit variance entries, and $\sigma > 0$ is a small noise scale.

*Proposition A.1 (Local smoothness under small perturbations).* For sufficiently small $\sigma$, the expected per-node contribution of $v_i$ to the consistency loss admits the approximation

$$\mathbb{E}_{\Delta_1,\Delta_2}\left[\left\|f_\theta(X^{(1)})_i - f_\theta(X^{(2)})_i\right\|_2^2\right] \approx 2\sigma^2\,\mathbb{E}_\Delta\left[\|J_i(X)\,\Delta\|_2^2\right], \quad (20)$$

where $J_i(X)$ is the Jacobian of $f_\theta(\cdot)_i$ with respect to the input features at $X$, and the expectation on the right-hand side is over a single noise matrix $\Delta$ with the same distribution as $\Delta_1$ and $\Delta_2$.

*Proof sketch.* Applying a first-order Taylor expansion of $f_\theta(X^{(k)})_i$ around $X$ for $k \in \{1, 2\}$ and using the independence and zero-mean properties of the noise, the leading term in the expectation of the squared difference is proportional to $\sigma^2$ times the squared Jacobian norm applied to the noise direction. Higher-order terms in $\sigma$ are negligible when $\sigma$ is small. □

Aggregating over unlabeled nodes and using the linearity of expectation, Proposition A.1 shows that, in the small-noise regime, minimizing $\mathcal{L}_{\text{cons}}$ approximately penalizes the sensitivity of predictions with respect to feature perturbations, as measured by the Jacobian of the encoder. In the context of UniLGT, this Jacobian depends on the hierarchical masks and MoE experts, so the induced smoothness prior is realized simultaneously along local, cluster-level, and global receptive fields.

## B Additional Ablation Results

Table 2 presents ablation results on the Cora dataset under varying label rates, showing the contribution of each proposed component. Table **??** provides a comprehensive comparison across all six datasets under the $\rho = 0.05$ label rate, illustrating the consistent benefits of both multi-level consistency regularization (CR) and degree-aware label selection (DAS).

## C Effect of Label Rate on Cora

We further provide an illustrative table summarizing how node classification accuracy on Cora varies with the label rate $\rho$ for UniLGT and two representative low-label baselines, SACN and VCHN.

## References


[1] Abien Fred Agarap. 2018. Deep Learning using Rectified Linear Units (ReLU). *arXiv preprint arXiv:1803.08375* (2018).

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450* (2016).

[3] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In *AAAI Conference on Artificial*

**Table 2: Ablation results on Cora dataset under different label rates (accuracy in %).**

| Multi-level Cons. | Degree-aware Sel. | $\rho$=0.05 | $\rho$=0.10 | $\rho$=0.20 | $\rho$=0.50 | $\rho$=1.00 |
|---|---|---|---|---|---|---|
| ✓ | ✓ | 85.27 | 87.02 | 89.51 | 90.59 | 93.95 |
| | ✓ | 84.49 | 85.97 | 88.04 | 89.66 | 90.69 |
| ✓ | | 87.15 | 86.41 | 84.79 | 87.44 | 90.99 |
| | | 83.35 | 85.27 | 84.09 | 85.72 | 88.48 |

**Note: "✓" indicates the component is enabled.**

**Table 3: Illustrative accuracy (in %) on Cora under different label rates $\rho$. Results are reported for UniLGT, SACN, and VCHN.**

| Method | $\rho$=0.005 | $\rho$=0.01 | $\rho$=0.03 |
|---|---|---|---|
| UniLGT | 78.1 | 79.8 | 84.1 |
| SACN | 76.2 | 79.2 | 83.6 |
| VCHN | 74.9 | 77.8 | 83.1 |

*Intelligence*.

[4] Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. 2022. NAGphormer: A Tokenized Graph Transformer for Node Classification in Large Graphs. In *International Conference on Learning Representations*.

[5] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*.

[6] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, et al. 2020. Rethinking Attention with Performers. *arXiv preprint arXiv:2009.14794* (2020).

[7] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems*.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[9] Vijay Prakash Dwivedi and Xavier Bresson. 2021. A Generalization of Transformer Networks to Graphs. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[10] Wu Feng, Yonggang Zhang, Yuxiao Dong, and Jiawei Han. 2020. Graph Random Neural Networks for Semi-supervised Learning on Graphs. In *Advances in Neural Information Processing Systems*.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*.

[12] Xiaoyang Huan et al. 2023. UniMF: A Unified Multimodal Framework with Hierarchical Attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[13] Siyuan Huang, Yunchong Song, Jiayue Zhou, and Zhouhan Lin. 2020. Biomedical Event Extraction with Hierarchical Knowledge Graphs. In *Advances in Neural Information Processing Systems*.

[14] Chenhui Hussain, Zichao Yue, and Zhiru Zhang. 2023. PolyNormer: Polynomial-expressive Graph Transformer in Linear Time. In *International Conference on Learning Representations*.

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[16] Johannes Klicpera, Aleksandar Bojchevski, and Stephan G"unnemann. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*.

[17] Zhuolin Liao, Xiaolin Zhang, Wei Su, and Kun Zhan. 2022. View-consistent heterogeneous network on graphs with few labeled nodes. *IEEE Transactions on Cybernetics* 53, 9 (2022), 5523–5532.

[18] Peter Mernyei and Ct1blina Cangea. 2020. Wiki-CS: A Wikipedia-based Benchmark for Graph Neural Networks. *arXiv preprint arXiv:2007.02901* (2020).

[19] Meng Qu, Yoshua Bengio, and Jian Tang. 2019. GMNN: Graph Markov Neural Networks. In *International Conference on Machine Learning*.

[20] Luka Rampšek, Mikita Galkin, Julian Lange, James Rowbottom, Pietro Lio, Michael Bronstein, and Xavier Bresson. 2022. Recipe for a General, Powerful, Scalable Graph Transformer. *arXiv preprint arXiv:2205.12454* (2022).

[21] Yuxiang Ren, Hongzhi Guo, Shaojing Li, et al. 2020. HGAT: Hierarchical Graph Attention Network for Fake News Detection. In *Conference on Information and Knowledge Management*.

[22] Prithviraj Sen, Gal Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective Classification in Network Data. In *AAAI Spring Symposium on Social Information Processing*.

[23] Hamed Shah, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. 2022. Exphormer: Sparse Transformers for Graphs. In *International Conference on Machine Learning*.

[24] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan G"unnemann. 2018. Pitfalls of Graph Neural Network Evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lus Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[26] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*.

[27] Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Kompella, and Zhangyang Wang. 2023. Graph Mixture of Experts: Learning on Large-scale Graphs with Explicit Diversity Modeling. *Advances in Neural Information Processing Systems* 36 (2023), 50825–50837.

[28] Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. 2023. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems* 36 (2023), 64753–64773.

[29] Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. 2025. M3Dphormer: A Unified Hierarchical Mask-based Graph Transformer. In *International Conference on Machine Learning*.

[30] Shuaike Xu, Xiaolin Zhang, Peng Zhang, and Kun Zhan. 2024. Structure-Aware Consensus Network on Graphs with Few Labeled Nodes. *arXiv preprint arXiv:2407.02188* (2024).

[31] Zhiqiang Xu et al. 2023. CoBFormer: Mitigating Over-smoothing in Graph Transformers. *arXiv preprint arXiv:2405.01102* (2023).

[32] Zhitao Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Badly for Graph Representation?. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[33] Hanqing Zeng, Hanjia Lyu, Diyi Hu, Yinglong Xia, and Jiebo Luo. 2024. Mixture of Weak and Strong Experts on Graphs. In *International Conference on Learning Representations*.

[34] Biao Zhang and Rico Sennrich. 2019. Root Mean Square Layer Normalization. *Advances in Neural Information Processing Systems* 32 (2019).

[35] Hongyang Zhang, Yi Gao, Jie Wang, Zhiyuan Xu, and Jian Pei. 2019. Hierarchical Graph Pooling with Structure Learning. In *Advances in Neural Information Processing Systems*.

[36] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2019. Bayesian Graph Convolutional Neural Networks for Semi-supervised Classification. In *AAAI Conference on Artificial Intelligence*.

[37] Lingxiao Zhao and Leman Akoglu. 2020. Uncertainty-aware Semi-supervised Learning on Graph Data. In *SIAM International Conference on Data Mining*.

[38] Lingxiao Zhao and Leman Akoglu. 2021. Data Augmentation for Graph Neural Networks. In *AAAI Conference on Artificial Intelligence*.