Stage-Aware Graph Contrastive Learning with Node-oriented Mixture of Experts

Xiangkai Zhu[1], Yeyu Yan[2], Saiqin Long[1]*, Chao Li[3], Guanwen Chen[1], Longsheng Su[1]

[1]Jinan University
[2]Beijing Jiaotong University
[3]Shandong University of Science and Technology
18063597830@163.com, yanyeyu-work@foxmail.com, saiqinlong@jnu.edu.cn, lichao@sdust.edu.cn,
20200310507@stu.fosu.edu.cn, 20202005280@m.scnu.edu.cn,

Abstract

Text-attributed graphs (TAGs), which associate rich textual descriptions with each node, are widely employed to represent complex relationships among real-world textual entities. Currently, representation learning for TAGs leverages large language models (LLMs) to transform node-matched textual descriptions into node features or labels, followed by the message passing in graph neural networks (GNNs) that further improves the expressiveness of graph representation learning. Nevertheless, a simple experiment we conducted demonstrates that not all LLMs are readily compatible with GNNs. A salient finding indicates that architectural heterogeneity among LLMs manifests as substantial performance gap across diverse TAGs representation learning. Moreover, the node semantics encoded by LLMs are often misaligned with the message passing in GNNs, causing performance collapse. Motivated by this observation, we propose a novel self-supervised graph learning framework called Stage-Aware Graph Contrastive Learning (SAGCL). In particular, we propose the node-oriented mixture of experts (NodeMoE) to assign suitable candidate experts for each node. It flexibly balances the strengths of different language experts by low-rank decomposition and reparameterization strategies. Subsequently, to align the inductive biases of graph structures with the semantic perception capabilities of LLMs, the message passing in GNNs is decoupled into the feature transformation stage and the feature propagation stage. Given the two stage views, stage-aware graph contrastive learning is proposed to match the node semantics encoded by the LLM with the locally aware topological patterns within the GNN via self-supervised contrastive learning. Experiments on eight datasets and three downstream tasks demonstrate the effectiveness of SAGCL.

Code — https://github.com/boshizhu/SAGCL.git

Introduction

Text-attributed graphs (TAGs) are structured data that combine graph topology with rich textual semantics, typically by associating long-form text with each node. Due to their rich semantic information and explicit relational patterns, TAGs are receiving growing attention in domains such as citation networks (Zheng et al. 2023a), recommendation scenarios (Chen et al. 2025), and biological networks (Liu et al. 2024).

To extract potential information, prevailing methods (Tang et al. 2024) (Kong et al. 2024) commonly leverage both textual information and structured data to jointly train graph models. Despite its effectiveness, the main limitation lies in the reliance on large-scale, high-quality labels, which are costly and time-consuming to obtain. This constraint limits the scalability and generality of supervised training.

Self-supervised learning (SSL) has garnered significant attention in recent years, achieving GNN learning without labeled data by constructing a pretext task. As a mainstream paradigm in SSL, graph contrastive learning (GCL) is widely applied due to its simplicity and effectiveness in learning discriminative graph representations.

It typically employs graph augmentation to generate perturbed views. Then, GNN training can be achieved by maximizing the agreement between positive node pairs while minimizing the agreement with negative node pairs.

Exemplified by GRACE (Zhu et al. 2020) and BGRL (Thakoor et al. 2021), operate by creating augmented views via feature or edge augmentation, self-supervised learning is achieved by maximizing consensus between identical nodes across augmented views. Other methods (DGI (Veličković et al. 2019), GGD (Zheng et al. 2022)), create contrastive views by perturbing node features, yielding different representations of the same node and a global summary vector, and employ InfoNCE (Oord, Li, and Vinyals 2018) to achieve contrastive learning.

Despite GCL is widely applied, significant limitations are still exhibited when dealing with text-attributed graphs. Earlier studies typically represent textual node attributes using shallow encoders, such as Word2Vec (Mikolov et al. 2013) or bag-of-words (Salton, Wong, and Yang 1975). They operate only at the word or bag level and cannot capture comprehensive sentence-, paragraph-, and document-level semantics. Consequently, such encodings represent suboptimal design choices, further constrained by the limited expressiveness of their text transformation functions. Leveraging their powerful capabilities in comprehending and transforming textual data, LLMs offer a promising approach for graph augmentation. Surprisingly, we find that not all LLM-encoded embeddings universally enhance the performance of arbitrary GNNs, as illustrated in Figure 1.

*Corresponding Author.

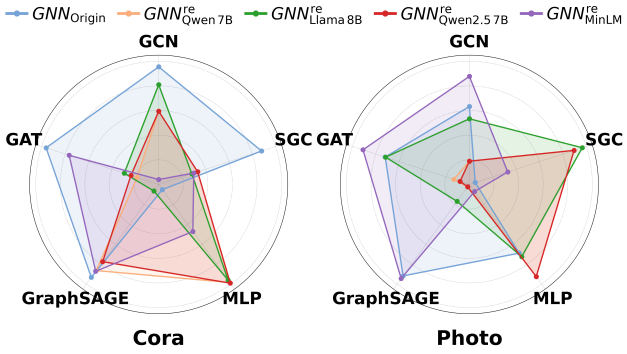

Figure 1: Performance of GNN_* with different GNN architectures on Cora and Photo datasets. $\text{GNN}_{\text{Origin}}$ denotes the original node features, GNN^{re}_* indicates that the original node features are replaced by LLM-encoded embeddings.

When adopting MLP as the downstream host encoder, $\text{GNN}_{\text{Origin}}$ consistently exhibits the significant performance gap compared to $\text{GNN}^{\text{re}}_{\text{Llama}}$ and $\text{GNN}^{\text{re}}_{\text{Qwen}}$. This indicates that, in the absence of structural information injection, LLMs can indeed substantially enhance the semantic richness of node representations. Nevertheless, when the host encoder is replaced by various GNNs, embeddings encoded by different LLMs ultimately exhibit pronounced and heterogeneous disparities. This discrepancy may be primarily attributable to differences in their model architectures and pre-training corpora. This naturally raises an intriguing research question: **Q1.** *How can the strengths of diverse LLMs be optimally reconciled to yield the node semantics most beneficial to downstream tasks?* Ideally, the above problem can be readily solved by exhaustively evaluating every candidate LLM on the downstream task and selecting the text encoder that maximizes task-specific performance. Unfortunately, the divergent local topological patterns of nodes impose fine-grained selection demands on node textual semantics (Zheng et al. 2023b). In other words, we need to assign an appropriate expert candidate combination to each node.

As discussed earlier, LLM enriches node semantics and consistently outperforms the original node features on MLP. However, directly feeding the node embedding encoded by LLMs into conventional GNNs may lead to *performance collapse*. For example, compared with GNN_{Origin} and GNN^{re}_*, Figure 1 shows a significant performance gap when the host GNN is either GCN (Kipf and Welling 2016) or GraphSAGE (Hamilton, Ying, and Leskovec 2017). The most likely reason lies in the fact that the semantic misalignment exists between LLM-encoded embeddings and the locally perceived topological patterns of nodes. This further prompts a second equally compelling question: **Q2.** *How can the node semantics encoded by LLMs be aligned with the locally aware topological patterns within the GNN?* To bridge this gap in **Q2**, recent efforts (Fang et al. 2024) have developed detailed prompt templates that describe local graph structures in natural language, enhancing the capacity of LLMs to comprehend structural knowledge. However, converting explicit link relations into textual prompts inevitably sacrifices critical structural information (Li et al. 2025). In contrast, GNNs excel at graph-structured reasoning by intergrating neighborhood information through the message-passing mechanism. This prompts us to ask whether a unified framework can be devised that harnesses the complementary strengths of GNNs and LLMs, thereby unlocking their full potential?

Building on these insights, we propose a self-supervised GNN named **SAGCL**, which unleashes the potential of LLMs for GNNs via contrastive learning. To bridge the potential semantic gap introduced by different LLMs in node embeddings, we propose NodeMoE, a node-level mixture-of-experts , which breaks through the bottleneck of having all nodes share a single language expert and harmonizes the textual embeddings encoded by various LLMs. Subsequently, to enable hierarchical semantic alignment, the message passing in GNN is decoupled into feature transformation ($\mathbf{F_T}$) and feature propagation ($\mathbf{F_P}$) stages. Then, the stage-aware graph contrastive learning is proposed to match the node semantics encoded by the LLM with the locally aware topological patterns within the GNN via contrastive learning. It alleviates the problem of semantic mismatches of LLMs at both the feature transformation and propagation stages, leading to more expressive graph representations.

Our main contributions can be summarized below:

- We propose a node-oriented mixture-of-experts approach that harmonizes the complementary strengths of multiple LLMs. It offers an elegant low-rank approximation-based reparameterization to decompose the weight matrix, which makes the parameters simpler and creates a clever balance between global and local LLMs.
- We revisit message passing in GNNs and decouple it into the feature transformation (**FT**) and feature propagation (**FP**) stages. Then, a stage-aware graph contrastive learning is proposed to match the node semantics encoded by the LLM with the locally aware topological patterns within the GNN.
- Experiments on eight datasets and three downstream tasks demonstrate the effectiveness of SAGCL.

Related Work

Graph Contrastive learning. In recent years, graph self-supervised learning (Yan et al. 2025), which dispenses with labeled data has garnered significant attention. Graph contrastive learning (GCL), as the mainstream approach, has achieved performance comparable to semi-supervised methods on many representation learning tasks. Methods such as DGI (Veličković et al. 2019) and GGD (Zheng et al. 2022), which generate negative samples by perturbing the input graph, optimize contrastive objectives that maximize the mutual information between local node embeddings and the global graph embedding. Other methods, like GRACE (Zhu et al. 2020) or GCL-IVG (Wo et al. 2024), first create two augmented graph views by applying data augmentations like edge pruning and feature dropping. The two augmented views are subsequently processed by a parameter-shared graph encoder, yielding node embeddings. A contrastive loss is employed to pull together the representations

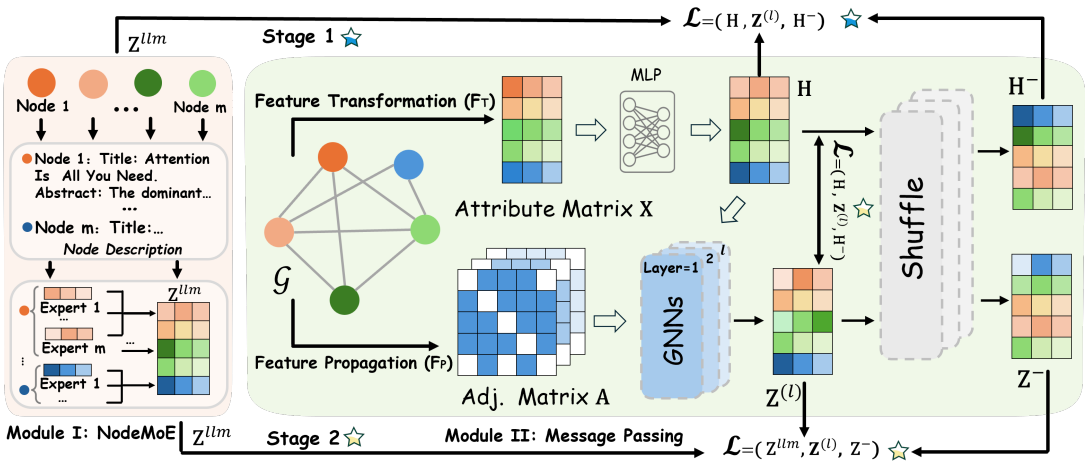

Figure 2: The Framework of SAGCL. It is composed of two modules: NodeMoE and Message Passing. NodeMoE balances the strengths of diverse language experts while aligning initial node features and topology-biased embeddings via self-supervised contrastive learning. In Message Passing module, the locally aware topological patterns in GNNs is captured, and node representations is dynamically updated.

of the same node across the two views while pushing apart representations of different nodes, thereby enabling effective self-supervised learning. These methods primarily depend on well-designed strategies for constructing positive and negative sample pairs in order to provide informative contrastive signals.

Representation learning on TAGs. Recently, LLMs have attracted wide attention in natural language processing tasks owing to their superior capacity to capture long-range contextual semantics. Several researchers try to employ them as text encoder to replace shallow encoders, and have further integrated it with GNNs to improve graph representation learning. Specifically, LLMs generate additional textual information, such as knowledge entity, and pseudo-labels, which is encoded to serve as initial node embeddings for GNNs. For example, TAPE (He et al. 2023) and KEA (Chen et al. 2024) generate explanations and knowledge entities to enrich textual attributes, while GIANT (Chien et al. 2021) and SimTeG (Duan et al. 2023) capture structural information by tasks such as link prediction. Other methods try to convert graph-structural information into textual descriptions, which are then fed into an LLM for prediction. For instance, GraphText (Zhao et al. 2023) uses a graph parse tree to transform the structure into a node sequence. GAugLLM (Fang et al. 2024) proposes the mixture-of-prompt-experts, which adaptively augments the original textual attributes with multiple prompt experts and enhances graph representation learning by the edge modifier and contrastive learning.

Preliminary

Attribute Graph. Given an attribute graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathcal{T})$ with node set $\mathcal{V}$ ($|\mathcal{V}| = N$), where $\mathbf{X} \in \mathbb{R}^{N \times f}$ is a f-dimensional node feature matrix, and $\mathcal{T} = \{c_i\}_{i=1}^{N}$ encapsulates textual descriptions c of each node. The edge set $\mathcal{E}$ denotes the connections between the nodes, which can be rewritten into the adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, and $\mathbf{A}_{i,j} \in \mathcal{E}$ denotes the link between node i and j.

Graph Contrastive Learning (GCL). As a self-supervised learning paradigm, GCL aims to learn node representations by maximizing the mutual information between different augmented views of the graph (Zhu et al. 2020). Formally, given two augmented graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ derived from $\mathcal{G}$, GNN training can be achieved by maximizing the agreement between positive node pairs while minimizing the agreement with negative node pairs. This is typically achieved through the InfoNCE loss (Oord, Li, and Vinyals 2018):

$$\mathcal{L}_{GCL} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(sim(\mathbf{z}_i^{\mathcal{G}_1}, \mathbf{z}_i^{\mathcal{G}_2})/\tau)}{\sum_{j\neq i}\exp(sim(\mathbf{z}_i^{\mathcal{G}_1}, \mathbf{z}_j^{\mathcal{G}_2})/\tau)}, \quad (1)$$

where $sim(\cdot,\cdot)$ denotes a similarity function, τ denotes the temperature parameter. $\mathbf{z}_i^{\mathcal{G}_1} \in \mathbb{R}^d$ and $\mathbf{z}_i^{\mathcal{G}_2} \in \mathbb{R}^d$ are node representations of node i obtained by $\mathcal{G}_1$ and $\mathcal{G}_2$ through the GNN encoder $f_g : \mathcal{G} \to \mathbb{R}^{N \times d}$, respectively.

Problem. Given an attribute graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathcal{T})$ and M large language models serving as text encoders $\{f_m : \mathcal{T} \to \mathbb{R}^{N \times f}\}_m^M$. Without any supervision signal guidance, our purpose is to achieve alignment between the node semantics encoded by LLMs and the locally aware topological patterns captured by GNN.

Method

In this section, we propose the SAGCL, and Figure 2 illustrates its core framework. Specifically, in the stage of text semantic encoding, each LLM-derived node embedding is adaptively reweighted via the proposed NodeMoE, enabling fine-grained, node-level semantic fusion. Subsequently, the message passing in GNNs is decoupled into feature transformation ($\mathbf{F_T}$) and feature propagation ($\mathbf{F_P}$). Building on this, stage-aware graph contrastive learning is proposed to rectify the semantic mismatches of LLMs at both the feature

transformation and propagation stages, while enabling complementary integration between LLM and GNN in a self-supervised manner.

Node-oriented Mixture of Experts

As discussed above, different language experts exhibit distinct semantic preferences, revealing their heterogeneous understandings of textual content. To fully leverage the complementary strengths of diverse language experts (**Q1**), we propose a novel collaboration strategy termed NodeMoE. It aims to allocate a dedicated composition of experts to each node, and Figure 3 illustrates the process of NodeMoE.

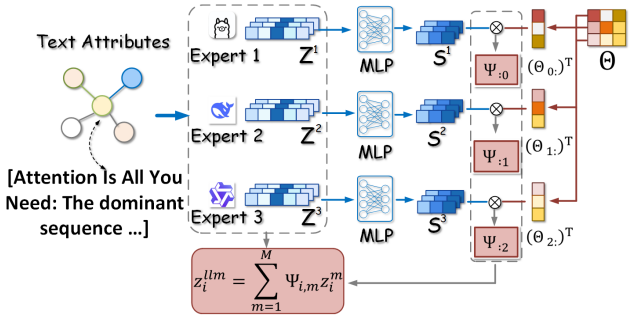

Figure 3: The framework of NodeMoE. It decouples the parameter Ψ into $\mathbf{S}^m$ and Θ, where $\mathbf{S}^m$ is node-related obtained by transforming $\mathbf{Z}^m$, Θ is the parameter matrix.

In contrast to conventional MoE frameworks, NodeMoE departs from gating mechanisms, instead deriving expert-specific node embeddings via a pre-training phase. Subsequently, the representations from different experts are adaptively fused through node-level synergistic coefficients. Specifically, to map the raw text attributes into a hidden space, a text encoder is pretrained. Instead of relying on shallow embeddings, LLMs (e.g., Llama (DeepSeek-AI 2025), Qwen (Yang et al. 2024)) are employed to encode the domain-specific text data. Formally, given an attribute graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathcal{T})$, the node embedding encoded by the m-th expert can be defined as:

$$\mathbf{z}_i^m = g_m(f_m(c_i)), \tag{2}$$

where $\mathbf{z}_i^m \in \mathbb{R}^d$ is the embedding of node i by m-th expert, $g_m : \mathbb{R}^f \to \mathbb{R}^d$ and $f_m : \mathcal{T} \to \mathbb{R}^{N \times f}$ denote the transformation function and text encoder of the m-th expert, respectively.

To enable each node to adaptively fuse different language experts, the synergistic coefficients $\Psi \in \mathbb{R}^{N \times M}$ is proposed to allocate node-specific expert weights.

$$\mathbf{z}_i^{llm} = \sum_{m=1}^{M} \Psi_{i,m} \mathbf{z}_i^m, \tag{3}$$

where $\Psi_{i,m}$ denotes the normalized attention weight of the m-th expert assigned to node i, with $\sum_{m=1}^{M} \Psi_{i,m} = 1$. The integrated node representation $\mathbf{z}_i^{llm} \in \mathbb{R}^d$ is formed by aggregating the M expert embeddings via weighted summation. However, in practice, direct learning the synergistic coefficients $\Psi \in \mathbb{R}^{N \times M}$ is limited, where N and M are related to the number of total nodes and experts, resulting in a substantial computational burden. To achieve a balance between learnable parameters and computational efficiency, we leverage the idea of low-rank matrix factorization and propose a separable reparameterization strategy that learns Ψ indirectly rather than directly, thereby effectively circumventing these challenges. The weight of m-th expert can be denoted as:

$$\Psi_{:,m} = softmax(\mathbf{S}^m \cdot \Theta_{m,:}^\top), \tag{4}$$

where $\Theta_{m,:} \in \mathbb{R}^{1 \times r}$ is a node-independent Learnable parameter matrix, $\mathbf{S}^m \in \mathbb{R}^{N \times r}$ can be regarded as node-dependent trainable matrix, which can be obtained by a simple yet effective nonlinear transformation function $F(\cdot)$:

$$\mathbf{S}^m = F(\mathbf{Z}^m, \mathbf{W}^m) := \sigma(\mathbf{Z}^m \mathbf{W}^m), \tag{5}$$

where $\mathbf{W}^m \in \mathbb{R}^{d \times r}$ is the weight matrix, $\mathbf{Z}^m = \{\mathbf{z}_i^m\}_{i=1}^N$ denotes the node embeddings encoded by m-th expert. Under the low-rank reparameterization, the weight for m-th expert is expressed as $\Psi_{:,m} = \mathbf{S}^m \Theta_{m,:}^\top = \sigma(\mathbf{Z}^m \mathbf{W}^m) \Theta_{m,:}^\top$, which simplifies to two matrix multiplications. Accordingly, the computational complexity of the low-rank formulation is $\mathcal{O}(n \times r \times (d + M))$.

The advantages of low-rank factorization are obvious. Firstly, the learnable parameters of Ψ shrink from $\mathbb{R}^{N \times M}$ to $\mathbb{R}^{M \times r}$, making them dependent solely on the number of experts M and the rank r, and completely independent of the node count N. Secondly, the reparameterization strategy elegantly addresses the optimization problem by learning a transformation function $F(\cdot; \mathbf{W}^m)$ to adaptively estimate the node correlation matrix $\mathbf{S}^m$, instead of optimizing $\Psi_{i,m}$ using only $\mathbf{z}_i^m$.

Stage-aware graph contrastive learning in message passing

As pointed out in (Zheng et al. 2023a), message passing of GNNs can be decoupled into two stages: feature transformation ($\mathbf{F_T}$) and feature propagation ($\mathbf{F_P}$). In the feature transformation stage, high-dimensional node features are transformed into low-dimensional yet informative representations. The feature propagation stage updates node representations by the aggregation of neighborhood information. For convenience, we adopt a vanilla example, which employs a l-order SGC (Wu et al. 2019) as the GNN:

$$(\mathbf{F_T}) : \mathbf{H} = \psi(\mathbf{X}, \mathbf{W}), \quad (\mathbf{F_P}) : \mathbf{Z}^{(l)} = f_g^{(l)}(\mathbf{Z}^{(l-1)}, \mathbf{A}), \tag{6}$$

where $\psi(\cdot, \cdot)$ is a transformation function that maps $\mathbf{X} \in \mathbb{R}^{N \times f}$ to $\mathbf{H} \in \mathbb{R}^{N \times d}$, and $\mathbf{W}$ represents the parameter matrix. Without loss of generality, $\psi(\cdot, \cdot)$ is usually defined as a one-layer MLP. $f_g^{(l)}(\cdot, \cdot)$ serves as the l-th propagation function, and $\mathbf{Z}^{(0)} = \mathbf{H}$.

Stage 1: Semantic alignment in feature transformation

As discussed earlier, message passing in GNNs involves two key processes: $\mathbf{F_T}$ and $\mathbf{F_P}$. Existing approaches that combine LLMs with GNNs primarily focus on the $\mathbf{F_P}$ stage, where LLM-generated node embeddings are used as an enhanced view and jointly trained with GNNs. LLMs leverage their Transformer architectures to model long-range dependencies and capture cross-sentence semantics, enriching

node-level semantic features. However, since these features are often high dimensional, the message passing mechanism of GNNs struggles to process them effectively, making it difficult to align the node semantics encoded by LLMs with the local perception-based message passing mechanism of GNNs (**Q2**).

To address the problem of node feature semantic mismatch, we propose stage-aware graph contrastive learning. It aligns representations between the feature propagation and transformation stages in GNNs. Inspired by contrastive learning, we employ it as a bridge for aligning different semantics, thus achieving implicit information injection. Notably, it requires no modifications to the GNN architecture. The alignment stage for feature transformation can be formulated as:

$$\mathcal{L}_{ft} = \frac{1}{N}\sum_{i=1}^{N}\left\{d\left(\mathbf{z}_i^{llm}, \mathbf{h}_i\right) - d\left(\mathbf{h}_i^{-}, \mathbf{h}_i\right) + \alpha\right\}_{\max}, \quad (7)$$

where $d(\cdot,\cdot)$ denotes the distance metric, $\{\cdot\}_{max} = \max\{\cdot, 0\}$, $\mathbf{h}_i = \mathbf{H}_{i,:} \in \mathbb{R}^d$, α $(\alpha \geq 0)$ is a non-negative value to ensure a safe distance between positive and negative embeddings. $\mathbf{h}_i^{-} = \mathbf{H}_{i,:}^{-} \in \mathbb{R}^d$ is negative sample, which is obtained by randomly permuting the rows of $\mathbf{H}$:

$$\mathbf{H}^{-} = Shuffle([\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_N]^\top), \quad (8)$$

where $Shuffle(\cdot)$ denotes the operation that directly row-shuffles the node embeddings.

Stage 2: Locally aware topological patterns alignment in feature propagation. During the feature propagation stage, we take shallow embeddings and graph structure as input and derives the node representations by aggregating neighborhoods. From this perspective, the first alignment step occurs between the raw features and the node embeddings that incorporate structural bias.

$$\mathcal{L}_{fp_1} = \frac{1}{N}\sum_{i=1}^{N}\left\{d\left(\mathbf{z}_i^{gnn}, \mathbf{h}_i\right) - d\left(\mathbf{h}_i^{-}, \mathbf{h}_i\right) + \alpha\right\}_{\max}, \quad (9)$$

where $\mathbf{z}_i^{gnn} = \mathbf{Z}_{i,:}^{(l)} \in \mathbb{R}^d$ is the embedding of node i following l layers of message passing. The next stage is the alignment of LLM-enhanced semantic features and embeddings derived from GNNs, as shown in Eq. (10).

$$\mathcal{L}_{fp_2} = \frac{1}{N}\sum_{i=1}^{N}\left\{d\left(\mathbf{z}_i^{gnn}, \mathbf{z}_i^{llm}\right) - d\left(\mathbf{z}_i^{-}, \mathbf{z}_i^{gnn}\right) + \alpha\right\}_{\max}, \quad (10)$$

where $\mathbf{z}_i^{-} = \mathbf{Z}_{i,:}^{-} \in \mathbb{R}^d$ is obtained by directly reshuffling $\mathbf{Z}^{(l)}$:

$$\mathbf{Z}^{-} = Shuffle([\mathbf{z}_1^{(l)}, \mathbf{z}_2^{(l)}, \cdots, \mathbf{z}_N^{(l)}]^\top), \quad (11)$$

Notably, in feature propagation stage, excessive accumulation of neighborhood information can drive node embeddings away from their original representations, rendering the $\{\cdot\}_{max}$ term in Eq. (9) nonzero and enlarging intra-class dispersion. To alleviate this issue, we formulate an upper-bound loss that enforces a hard constraint on the distance between each node's initial features and its final embedding, ensuring that they remain within a predefined safety region.

$$\mathcal{L}_{fp_1}^{up} = \frac{-1}{N}\sum_{i=1}^{N}\left\{d\left(\mathbf{z}_i^{gnn}, \mathbf{h}_i\right) - d\left(\mathbf{h}_i^{-}, \mathbf{h}_i\right) + \alpha + \beta\right\}_{\min}, \quad (12)$$

where β is a non-negative tuning parameter, $\{\cdot\}_{min} = \min\{\cdot, 0\}$.

The fianal loss can be derived as:

$$\mathcal{L}_{total} = \omega(\mathcal{L}_{ft} + \mathcal{L}_{fp_1}) + \mathcal{L}_{fp_2} + \mathcal{L}_{fp_1}^{up}, \quad (13)$$

where ω is a hyperparameter that establishes a dynamic balance between the shallow embedding in GNN and the semantic features enhanced by LLM

Experiments

Experimental Settings

Datasets. To comprehensively evaluate the effectiveness of SAGCL, eight graph datasets (Cora (Sen et al. 2008), CiteSeer (Giles, Bollacker, and Lawrence 1998), PubMed (Yang, Cohen, and Salakhudinov 2016), Wikics (Mernyei and Cangea 2007), Instagram (Huang et al. 2024), Photo and Computer (Ni, Li, and McAuley 2019), ogbn-arxiv(Hu et al. 2020)) are employed for three tasks: node classification (NC), node clustering (NClu), link prediction (LP).

Baselines & Implementations. We compare the proposed SAGCL with twelve representative graph self-supervised learning methods, including six graph contrastive learning methods (DGI (Veličković et al. 2019), GRACE (Zhu et al. 2020), BGRL (Thakoor et al. 2021), GREET (Liu et al. 2023), SUGRL (Mo et al. 2022), ROSEN (Zhuo et al. 2024)) and six graph autoencoder methods (MaskGAE (Li et al. 2023), GiGaMAE (Shi et al. 2023), S2GAE (Tan et al. 2023), Bandana (Zhao et al. 2024), BSG (Jung and Park 2025), TEDMAE (Zhang, Li, and Zhao 2025)). Average performance and its standard deviation are computed across five runs, each initialized with a different random seed.

Task Parameter Settings. For NC and NClu tasks, we follow the default split ratios reported in LLMNodeBed (Wu et al. 2025). Regarding datasets lacking predefined splits (e.g., Photo, and Computer), we implement a standardized partitioning protocol with 10% for training, 10% for validation, and 80% for testing. For LP task, we adopt a standardized 85%/10%/5% split ratio for the training, validation, and test sets, respectively.

For our proposed SAGCL, we utilize three language experts ($M = 3$) as text encoder: Llama-8B (DeepSeek-AI 2025), Qwen-7B (Yang et al. 2024), Qwen-2.5-7B (Team 2025). The rank $r = 3$ in Eq. (4) by default, the weighting coefficient $\omega = 10$ in Eq. (13) for all datasets. More details of datasets, baselines and task settings, and experiment environment are provided in the code of SAGCL.

Experimental Results

Performance of SAGCL on Various Downstream Tasks. To demonstrate the effectiveness of SAGCL, we conducted experiments on three downstream tasks: node classification (NC), node clustering (NClu), & link prediction (LP).

Method	Cora	CiteSeer	PubMed	WikiCS	Instagram	Photo	Computer	Arxiv
DGI	82.28±0.01	61.25±0.23	79.36±0.10	75.59±0.10	65.90±0.01	80.81±0.05	80.67±0.80	53.98±0.02
BGRL	82.99±0.03	65.83±0.02	76.10±0.05	74.67±0.05	64.78±0.05	80.58±0.05	80.62±0.37	OOM
GRACE	83.00±0.20	69.60±0.04	76.10±0.16	67.50±0.17	65.20±0.09	78.20±0.04	OOM	OOM
GREET	79.62±0.43	70.09±0.17	78.52±0.11	64.95±0.24	64.95±0.24	81.65±0.07	OOM	OOM
SUGRL	83.24±0.08	68.55±0.02	77.50±0.01	75.00±0.02	66.00±0.02	81.50±0.02	83.50±0.02	66.00±0.09
ROSEN	84.41±0.14	70.58±0.04	79.27±0.07	74.22±0.05	65.93±0.05	83.08±0.04	84.65±0.06	66.23±0.05
MaskGAE	82.99±0.25	71.50±0.08	78.60±0.06	67.30±0.21	65.60±0.03	82.27±0.07	84.37±0.03	59.63±0.16
GiGaMAE	84.30±0.06	61.50±0.21	77.60±0.04	76.20±0.02	65.97±0.04	83.00±0.02	OOM	OOM
S2GAE	80.80±0.07	68.30±0.02	78.60±0.05	63.40±0.07	65.70±0.01	82.58±0.04	83.56±0.05	65.97±0.10
Bandana	83.90±0.06	68.30±0.15	80.11±0.03	68.60±0.06	65.10±0.04	82.43±0.10	81.96±0.08	63.60±0.69
BSG	83.30±0.03	70.00±0.05	80.00±0.22	69.30±0.03	65.60±0.02	81.90±0.04	83.30±0.12	64.12±0.06
TEDMAE	84.10±0.13	70.50±0.07	78.20±0.03	76.80±0.04	66.20±0.04	82.81±0.02	OOM	OOM
SAGCL	**85.40±0.06**	**71.60±0.03**	**80.34±0.01**	**77.20±0.02**	**66.44±0.02**	**83.70±0.03**	**85.81±0.02**	**66.80±0.17**

Table 1: Performance comparison on eight datasets (Accuracy %). Bold and underlined scores denote the best and second-best results, respectively. OOM denotes out of memory.

Method	Cora	CiteSeer	PubMed	WikiCS	Instagram	Photo	Computer	Arxiv
DGI	0.652/0.640	0.324/0.320	0.419/0.497	0.557/0.560	0.030/0.071	0.618/0.576	0.614/0.572	0.378/0.354
BGRL	0.643/0.642	0.366/0.379	0.360/0.470	0.539/0.558	0.022/0.056	0.601/0.588	0.608/0.593	OOM
GRACE	0.637/0.655	0.424/0.444	0.373/0.407	0.465/0.460	0.024/0.058	0.574/0.538	OOM	OOM
GREET	0.504/0.405	0.375/0.318	0.408/0.447	0.519/0.543	0.003/0.006	0.625/0.619	OOM	OOM
SUGRL	0.654/0.652	0.390/0.409	0.394/0.436	0.537/0.552	0.031/0.066	0.611/0.609	0.643/0.635	0.496/0.520
ROSEN	0.667/0.675	0.434/0.455	0.418/0.471	0.528/0.544	0.025/0.054	0.644/0.634	0.669/0.664	0.503/0.520
MaskGAE	0.642/0.656	**0.444/0.474**	0.415/0.457	0.471/0.468	0.027/0.059	0.630/0.646	0.658/0.653	0.449/0.475
GiGaMAE	0.660/0.679	0.318/0.336	0.392/0.438	0.487/0.491	0.029/0.073	0.627/0.637	OOM	OOM
S2GAE	0.604/0.613	0.399/0.423	0.400/0.456	0.414/0.414	0.028/0.061	0.605/0.618	0.637/0.629	0.479/0.483
Bandana	0.653/0.676	0.389/0.416	0.429/0.489	0.477/0.479	0.021/0.038	0.634/0.613	0.626/0.602	0.409/0.392
BSG	0.648/0.667	0.425/0.446	0.429/0.482	0.487/0.492	0.026/0.054	0.623/0.602	0.640/0.635	0.336/0.267
TEDMAE	0.656/0.684	0.437/0.459	0.401/0.449	0.539/0.570	0.034/0.069	0.639/0.620	OOM	OOM
SAGCL	**0.681/0.707**	0.439/0.463	**0.435/0.496**	**0.563/0.583**	**0.036/0.073**	**0.656/0.641**	**0.682/0.681**	**0.509/0.529**

Table 2: Clustering performance (NMI / ARI).

Method	Cora	CiteSeer	PubMed
DGI	94.2±0.27	95.1±0.19	81.9±3.31
BGRL	96.2±0.43	97.5±0.12	96.3±0.48
GRACE	93.1±0.93	97.1±0.87	96.2±0.05
GREET	96.7±0.14	96.2±0.73	96.0±0.12
SUGRL	98.2±0.09	97.3±0.61	96.6±0.04
ROSEN	94.8±0.11	97.0±0.06	97.1±0.05
MaskGAE	97.2±0.04	93.5±0.43	98.8±0.06
GiGaMAE	95.2±0.21	96.2±0.25	96.4±0.11
S2GAE	95.4±0.60	88.4±0.84	**99.0±0.15**
Bandana	95.6±0.19	90.8±4.03	97.5±0.24
BSG	97.1±0.09	96.4±0.29	98.9±0.04
TEDMAE	96.3±0.31	96.0±0.20	96.9±0.22
SAGCL	**99.1±0.17**	**98.6±0.17**	97.3±0.02

Table 3: Link prediction performance (AUC %).

Node Classification. For NC task, we employ accuracy (ACC) to evaluate the efficacy of SAGCL, as shown in Table 1. SAGCL consistently outperforms all baselines on NC task, demonstrating that the proposed stage-aware graph contrastive learning fully exploits the potential of GNNs.

Node Clustering. For NClu task, we adopt the normalized mutual information (NMI) and adjusted rand index (ARI) as evaluation metrics, as shown in Table 2. It can be observed that SAGCL achieves the best/runner-up performance on eight datasets. Compared with GCL and GAE methods, SAGCL implicitly integrates the node features encoded by the language expert model into the GNN via contrastive learning, thereby improving the capability of graph representation learning.

Link Prediction. For LP task, we employ the area under the curve (AUC) as evaluation metrics, as shown in Table 3. Graph Autoencoders (GAEs), which reconstruct the original graph structure to learn node embeddings, serve as a natural baseline for LP. In contrast, SAGCL employs LLMs as the text encoder to generate node representations with high separability, which is further enhanced by the reparameterization strategy of NodeMoE. More results on LP task are detailed in the code of SAGCL.

Parameter Sensitivity Analysis

In this section, we examine the influence of the two key hyperparameters in proposed SAGCL, as shown in Figure 4.

The coefficient α regulates the separation between positive and negative samples, whereas β defines a safety margin that prevents the distance between positive pairs from becoming excessively large. Notably, neither α nor β should be set too low. The two hyperparameters are coupled, a small α requires a larger β, whereas an excessively large α calls for a smaller β. Otherwise, positive samples may become overly compact, risking representation-space collapse or gradient explosion.

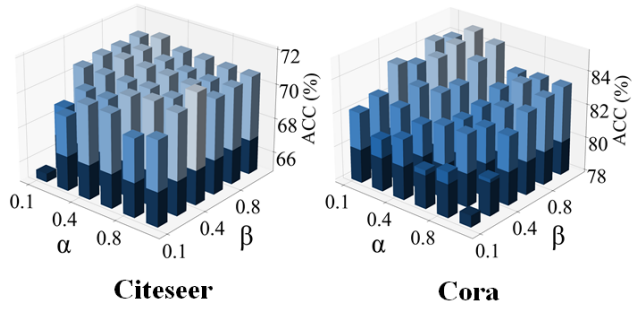

Figure 4: Parameter sensitivity analysis of α and β (Accuracy %).

Datasets	w/o $\mathbf{F_T}$	w/o $\mathbf{F_P}$	w/o up	SAGCL
Cora	83.7	80.1	77.3	**85.4**
CiteSeer	71.0	67.3	71.0	**71.6**
Photo	82.8	82.3	79.0	**83.7**
Computer	84.8	80.1	81.9	**85.8**
Ogbn-arxiv	49.2	64.3	60.5	**66.7**

Table 4: Ablation study of SAGCL (Accuracy %).

Ablation Study of SAGCL

In this section, to verify the effectiveness of each module, we remove certain components of the SAGCL, resulting in three variants as shown in Table 4. 'w/o $\mathbf{F_T}$' represents that SAGCL removes the contrastive learning in $\mathbf{F_T}$ stage and 'w/o $\mathbf{F_P}$' denotes that SAGCL ignores the alignment loss in $\mathbf{F_P}$ stage. 'w/o up' is the variant of SAGCL that eliminates the hard constraint on the distance between initial node features and structure-aware node representations.

We observe a heterogeneous contribution of individual modules across datasets. Specifically, on Cora dataset, the alignment performed in the $\mathbf{F_P}$ stage is markedly more influential than the alignment employed during the $\mathbf{F_T}$ stage. In addition, aligning the original features with the message-passed node representations alleviates the over-smoothing in GNNs. On Arxiv dataset, alignment in the $\mathbf{F_T}$ stage exerts a substantially greater influence than alignment in the $\mathbf{F_P}$ stage, highlighting the importance of node semantics.

Analysis of NodeMoE

To further verify that NodeMoE can select the most appropriate expert for each node, we visualize the language expert

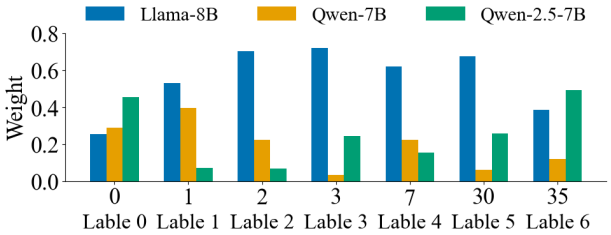

Figure 5: Node weights of NodeMoE learned by different language models on Cora.

weights assigned to nodes with different labels, as shown in Figure 5. Owing to the reparameterization strategy, every node in Cora dataset is encoded by the most suitable language expert, and the learned weights enable an adaptive fusion of their outputs.

Furthermore, we employ language experts Sentence-Bert(Reimers and Gurevych 2019) and MiniLM (Wang et al. 2020) to evaluate the effectiveness of NodeMoE. As Table 5 demonstrates, increasing language experts (M) enhances performance across all datasets until peak accuracy at M = 3 or 4. Beyond this point (M = 5), accuracy declines as the additional expert causes excessive semantic smoothing of node representations, introducing noise that degrades performance.

M	Cora	CiteSeer	PubMed	Photo	Computer
$M=1$	84.3	70.5	78.2	82.5	84.4
$M=2$	84.6	70.2	78.4	83.0	85.5
$M=3$	**85.4**	**71.6**	80.3	83.7	85.5
$M=4$	**85.4**	71.4	**80.6**	**83.8**	**85.6**
$M=5$	85.3	71.2	79.8	83.5	85.3

Table 5: Performance of NodeMoE with M experts (Accuracy %).

Conclusions

In this paper, we focus on the challenges that how to harmonize the strengths of diverse language experts and align the semantics of nodes at different stages. For the first challenge, we propose NodeMoE, a node-oriented mixture-of-experts. It offers an elegant low-rank approximation-based reparameterization to decompose the weight matrix and dynamically assigns each node appropriate language experts. To address the second challenge, we revisit the message-passing in GNNs and decouple it into $\mathbf{F_T}$ and $\mathbf{F_P}$ stages. Unlike previous methods that employ labeled data to align node semantics, we propose stage-aware graph contrastive learning, which leverages self-supervised contrastive learning to match the augmented node semantics encoded by LLMs with the local-aware topological patterns in GNNs. Experimental results consistently verify the effectiveness of SAGCL , showing its generalization on various tasks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. U23B2027, No. W2411053, the Guangdong Basic and Applied Basic Research Foundation under Grant No. 2024A1515010214, and Key Laboratory of Equipment Data Security and Guarantee Technology, Ministry of Education under Grant No.2024020200.

References

Chen, J.; Gao, C.; Yuan, S.; Liu, S.; Cai, Q.; and Jiang, P. 2025. Dlcrec: A novel approach for managing diversity in llm-based recommender systems. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, 857–865.

Chen, Z.; Mao, H.; Li, H.; Jin, W.; Wen, H.; Wei, X.; Wang, S.; Yin, D.; Fan, W.; Liu, H.; et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2): 42–61.

Chien, E.; Chang, W.-C.; Hsieh, C.-J.; Yu, H.-F.; Zhang, J.; Milenkovic, O.; and Dhillon, I. S. 2021. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Duan, K.; Liu, Q.; Chua, T.-S.; Yan, S.; Ooi, W. T.; Xie, Q.; and He, J. 2023. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*.

Fang, Y.; Fan, D.; Zha, D.; and Tan, Q. 2024. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 747–758.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, 89–98.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, X.; Bresson, X.; Laurent, T.; Hooi, B.; et al. 2023. Explanations as features: Llm-based features for text-attributed graphs. *arXiv preprint arXiv:2305.19523*, 2(4): 8.

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

Huang, X.; Han, K.; Yang, Y.; Bao, D.; Tao, Q.; Chai, Z.; and Zhu, Q. 2024. Can gnn be good adapter for llms? In *Proceedings of the ACM Web Conference 2024*, 893–904.

Jung, H.; and Park, H. 2025. Balancing graph embedding smoothness in self-supervised learning via information-theoretic decomposition. In *Proceedings of the ACM on Web Conference 2025*, 2621–2632.

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

Kong, L.; Feng, J.; Liu, H.; Huang, C.; Huang, J.; Chen, Y.; and Zhang, M. 2024. GOFA: A Generative One-For-All Model for Joint Graph Language Modeling. In *The Thirteenth International Conference on Learning Representations*.

Li, J.; Wu, R.; Sun, W.; Chen, L.; Tian, S.; Zhu, L.; Meng, C.; Zheng, Z.; and Wang, W. 2023. What's Behind the Mask: Understanding Masked Graph Modeling for Graph Autoencoders. In *KDD*, 1268–1279. ACM.

Li, J.; Wu, R.; Zhu, Y.; Zhang, H.; Chen, L.; and Zheng, Z. 2025. Are Large Language Models In-Context Graph Learners? *arXiv preprint arXiv:2502.13562*.

Liu, H.; Feng, J.; Kong, L.; Liang, N.; Tao, D.; Chen, Y.; and Zhang, M. 2024. One For All: Towards Training One Graph Model For All Classification Tasks. In *The Twelfth International Conference on Learning Representations*.

Liu, Y.; Zheng, Y.; Zhang, D.; Lee, V.; and Pan, S. 2023. Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. In *AAAI*.

Mernyei, P.; and Cangea, C. W.-C. 2007. A wikipedia-based benchmark for graph neural networks. arXiv 2020. *arXiv preprint arXiv:2007.02901*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mo, Y.; Peng, L.; Xu, J.; Shi, X.; and Zhu, X. 2022. Simple unsupervised graph representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 7797–7805.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 188–197.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Salton, G.; Wong, A.; and Yang, C.-S. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11): 613–620.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Shi, Y.; Dong, Y.; Tan, Q.; Li, J.; and Liu, N. 2023. Gigamae: Generalizable graph masked autoencoder via collaborative

latent space reconstruction. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2259–2269.

Tan, Q.; Liu, N.; Huang, X.; Choi, S.-H.; Li, L.; Chen, R.; and Hu, X. 2023. S2gae: Self-supervised graph autoencoders are generalizable learners with graph masking. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, 787–795.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 491–500.

Team, Q. 2025. Qwen2.5-VL.

Thakoor, S.; Tallec, C.; Azar, M. G.; Munos, R.; Veličković, P.; and Valko, M. 2021. Bootstrapped representation learning on graphs. In *ICLR 2021 workshop on geometrical and topological representation learning*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *International Conference on Learning Representations*.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. arXiv:2002.10957.

Wo, Z.; Shao, M.; Wang, W.; Guo, X.; and Lin, L. 2024. Graph contrastive learning via interventional view generation. In *Proceedings of the ACM Web Conference 2024*, 1024–1034.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. Pmlr.

Wu, X.; Shen, Y.; Ge, F.; Shan, C.; Jiao, Y.; Sun, X.; and Cheng, H. 2025. A comprehensive analysis on llm-based node classification algorithms. *arXiv e-prints*, arXiv–2502.

Yan, Y.; Zheng, S.; Hui, W.; Zhu, X.; Chen, D.; Zhu, Z.; Zhao, Y.; and He, K. 2025. Towards Pre-trained Graph Condensation via Optimal Transport. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Fan, Z. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, 40–48. PMLR.

Zhang, Q.; Li, C.; and Zhao, Z. 2025. Teacher-guided Edge Discriminator for Personalized Graph Masked Autoencoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 13269–13276.

Zhao, J.; Zhuo, L.; Shen, Y.; Qu, M.; Liu, K.; Bronstein, M.; Zhu, Z.; and Tang, J. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*.

Zhao, Z.; Li, Y.; Zou, Y.; Tang, J.; and Li, R. 2024. Masked graph autoencoder with non-discrete bandwidths. In *Proceedings of the ACM Web Conference 2024*, 377–388.

Zheng, S.; Liu, Z.; Zhu, Z.; Zhang, X.; Li, J.; and Zhao, Y. 2023a. Unleashing the potential of GNNs via Bi-directional Knowledge Transfer. *arXiv preprint arXiv:2310.17132*.

Zheng, S.; Zhu, Z.; Liu, Z.; Li, Y.; and Zhao, Y. 2023b. Node-oriented Spectral Filtering for Graph Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zheng, Y.; Pan, S.; Lee, V.; Zheng, Y.; and Yu, P. S. 2022. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, 35: 10809–10820.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.

Zhuo, J.; Cui, C.; Fu, K.; Niu, B.; He, D.; Wang, C.; Guo, Y.; Wang, Z.; Cao, X.; and Yang, L. 2024. Graph contrastive learning reimagined: Exploring universality. In *Proceedings of the ACM Web Conference 2024*, 641–651.