Learning Graph Structures via Sparse Training with Adaptive Edge Growth

Sisi Chen
School of Computer and Information Technology
Shanxi University
Taiyuan, China
chensisi1@sxu.edu.cn

Nan Li*
School of Computer and Information Technology
Shanxi University
Taiyuan, China
linan10@sxu.edu.cn

Ruwang Jiao
School of Future Science and Engineering
Soochow University
Suzhou, China
ruwangjiao@gmail.com

***Abstract*—Graph Neural Networks (GNNs) have achieved remarkable success in processing graph-structured data, yet they continue to face two major limitations: high computational cost and the presence of structural noise in real-world graphs, such as missing edges or spurious connections. Existing sparse training approaches effectively reduce computational overhead by pruning model weights and node features. However, they operate strictly on the initial graph topology, and the resulting trained graph is merely a subgraph of the original one, leaving the structural optimization problem unaddressed. By contrast, Graph Structure Learning (GSL) methods are capable of dynamically optimizing graph topology during training. However, these approaches typically rely on global pairwise similarity computation, resulting in a quadratic time complexity of $\mathcal{O}(N^2)$, which severely limits their scalability to large-scale graphs. Although anchor-based variants of Iterative Deep Graph Learning (IDGL) partially alleviate this issue, they depend on low-rank approximations that inevitably discard local structural information of the original graph and may further introduce additional optimization instability. Therefore, an efficient graph structure optimization method that preserves local structural details is highly desirable. To address these limitations, we propose a dual-sparsity framework that integrates the principles of IDGL with the Comprehensive Graph Gradual Pruning (CGP) paradigm, enabling the joint optimization of model parameters and graph topology. Specifically, we compute cosine similarity between node pairs to measure semantic relevance, which guides both the addition of necessary edges and the removal of redundant ones. Meanwhile, to overcome the high computational cost inherent in standard IDGL, we employ a hybrid sampling strategy that combines 2-hop neighborhood search with random negative sampling, reducing the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(E)$. This strategy also mitigates the risk of the optimization process falling into local minima. Experiments on the Cora, Citeseer, Pubmed and large-scale Physics datasets demonstrate that the designed framework achieves good performance while maintaining dual sparsity in both the model and graph structure, achieving comparable or superior performance to static sparse baselines and exhibiting particularly notable generalization improvements on the larger Pubmed dataset.**

***Index Terms*—Graph Neural Networks, Sparse Training, Graph Structure Learning, Metric Learning, Hybrid Sampling**

Corresponding author : *Nan Li

This work was supported in part by the National Natural Science Foundation of China under Grant 62406212, in part by the Science and Technology Program of Jiangsu Province under Grant BK20251895, in part by the Suzhou Planning Project of Science and Technology under Grant ZXP2025060, and in part by the China Postdoctoral Science Foundation under Grant 2025M781499.

I. Introduction

Currently, Graph Neural Networks (GNNs) have become the established standard for processing non-Euclidean structured data [1], achieving state-of-the-art performance on tasks such as node classification and link prediction (e.g., GCN [2], GAT [3]). Seminal works like GraphSAGE [4] further introduced inductive sampling to handle large-scale graphs. However, applying GNNs to large-scale real-world graphs remains constrained by both computational cost and limitations inherent in the graph structure itself. First, as graph size grows, the computational overhead and memory consumption of GNNs increase rapidly, hindering their deployment on resource-constrained devices. Second, real-world graphs are often noisy and incomplete, which not only degrades the quality of message passing but also introduces redundant computation [5].

To address the computational bottleneck, sparse training techniques have been introduced, such as the Comprehensive Graph Gradual Pruning framework [6]. These approaches are inspired by the Lottery Ticket Hypothesis (LTH), which posits that within the original graph exists a smaller subgraph (a "winning ticket") that, when trained, can achieve the target performance. However, traditional LTH requires exhaustive pruning and repeated training of dense models, which significantly increases training time. Moreover, the dense model must be stored in memory throughout the process, making such methods impractical for low-capacity devices [7]. Therefore, researchers proposed the CGP framework, which enables the training of a sparse GNN in a single training cycle without repeatedly retraining dense models. Nevertheless, CGP still relies on the initial graph structure, which is often incomplete, typically missing critical semantic connections while containing irrelevant or noisy edges. Simply optimizing model weights cannot recover these missing links or remove redundant ones, ultimately limiting the performance.

To address the deficiencies of real-world graph data, researchers have proposed Graph Structure Learning approaches, such as Iterative Deep Graph Learning, which aim to alleviate structural noise by dynamically optimizing the graph topology

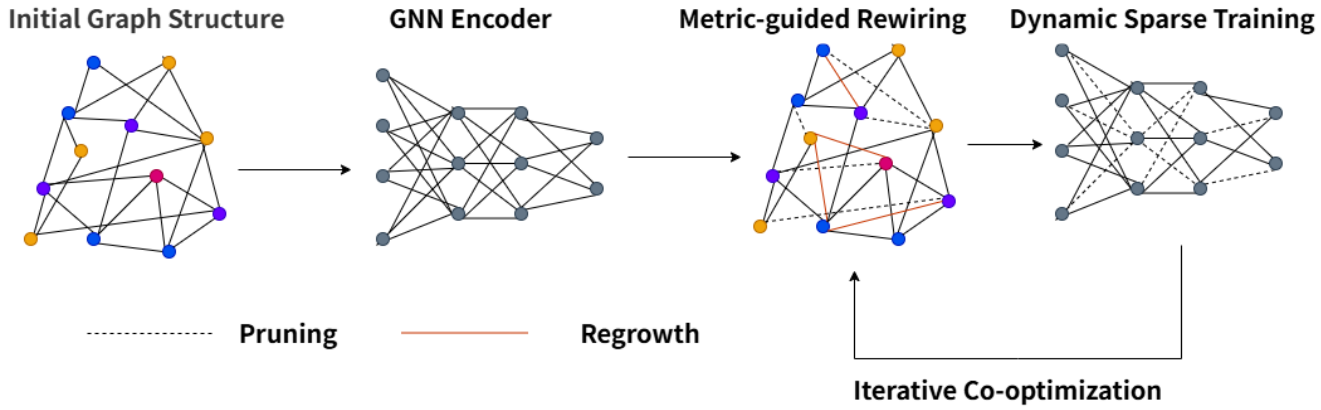

Fig. 1. **The Overall Framework of LGSF-AEG.** The proposed method performs joint optimization through two co-evolving processes: (1) **Metric-guided Graph Rewiring**, which dynamically refines the topology by pruning heterophilic edges (dashed lines) to remove noise and regrowing missing semantic connections (orange lines) via hybrid sampling; and (2) **Dynamic Sparse Training**, which adaptively optimizes model weights using a linear decay schedule. These two components iterate in a closed loop to achieve dual sparsity in both graph structure and model parameters.

based on node feature similarity. However, standard GSL implementations typically suffer from prohibitive $\mathcal{O}(N^2)$ computational complexity. Although some studies attempt to alleviate this bottleneck by introducing anchor-based approximations, such low-rank methods inevitably sacrifice fine-grained local structural details of the original graph and introduce additional optimization overhead. Consequently, existing GSL techniques remain difficult to scale to large graphs and fundamentally conflict with the efficiency goals pursued in sparse training.

To integrate the advantages of efficient sparse training and adaptive structural optimization, we propose a new dual-sparsity training framework (i.e., Learning Graph Structures with Adaptive Edge Growth (LGSF-AEG)). The overall process is shown in Fig. 1. By introducing a metric-guided topology rewiring mechanism, our method no longer relies on a fixed initial graph structure. Specifically, following the metric design of IDGL, we employ the cosine similarity between node pairs as the measurement criterion to determine which semantically meaningful edges should be added and which redundant edges ought to be removed. To further reduce the computational cost inherent in IDGL, we develop a hybrid sampling strategy that combines homophily-grounded 2-hop neighbor search with random negative sampling, reducing the overall complexity to $\mathcal{O}(E)$ while also mitigating the risk of the model falling into local minima. The contributions of this work are the following:

- We innovatively integrate the graph structure learning paradigm of IDGL into the sparse training framework of CGP. This breaks away from the traditional *static-topology* assumption and enables the *co-optimization* of model parameters and graph topology.
- To address the prohibitively expensive exhaustive computation in standard IDGL, we propose a hybrid sampling strategy that combines *triadic-closure-based 2-hop neighbor search* with *random negative sampling*, reducing the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(E)$.
- Building upon the metric-based similarity design in IDGL, we develop a lightweight, *parameter-free* rewiring mechanism that directly reuses the semantic features extracted by the sparse GNN backbone. This allows dynamic graph denoising and edge regeneration to be achieved at an extremely low computational cost.

II. Related Work

A. Sparse Training for GNNs

As the scale of graph datasets continues to grow, the high computational and memory costs of graph neural networks (GNNs) have become a major bottleneck for their practical deployment. Inspired by the Lottery Ticket Hypothesis, recent studies have begun to explore sparsification techniques for GNNs. Existing methods such as SGCN [8] and UGS [9] attempt to jointly optimize graph structures and model weights through iterative pruning. Although these approaches improve efficiency compared to the original LTH training paradigm, the alternating optimization process still incurs substantial computational overhead [10].

Subsequently, Comprehensive Graph Gradual Pruning proposed a more efficient progressive pruning framework that gradually removes redundant weights during training, significantly reducing resource consumption. However, existing sparse training methods (e.g., CGP) typically suffer from a critical limitation: they often assume that the input graph topology is static and reliable. As a result, these methods only achieve reductions in graph connectivity and model parameters, while overlooking the fact that the initial graph may contain noise or miss critical connections, ultimately leading to suboptimal performance.

B. Graph Structure Learning

To address the inherent noise and incompleteness in graph data, researchers have proposed graph structure learning meth-

ods, which aim to jointly optimize graph topology and graph neural network (GNN) parameters. Early work such as **LDS** [11] adopts a bilevel optimization framework, treating the discrete graph structure as a set of hyperparameters to be learned. Robustness-oriented approaches, such as **Pro-GNN** [12], enhance resistance to adversarial attacks by incorporating low-rank and sparsity constraints. More recently, **SLAPS** [13] introduces a self-supervised co-training framework that simultaneously optimizes the graph structure and GNN parameters. As a representative method, **Iterative Deep Graph Learning** [14] dynamically constructs graph structures based on learned node embeddings and leverages metric learning (e.g., weighted cosine similarity) to characterize semantic relationships between node pairs.

Despite their effectiveness, most GSL methods (e.g., LDS, Pro-GNN, and IDGL) rely on computing pairwise similarity matrices over all node pairs , resulting in a near-exhaustive computational cost of $\mathcal{O}(N^2)$. This severe complexity bottleneck significantly limits their applicability to large-scale graph datasets. Although some studies attempt to approximate full similarity computation via anchor-based sampling, such strategies may sacrifice fine-grained local topological information. In contrast, the proposed joint optimization framework adopts a hybrid sampling strategy that reduces the overall complexity to linear $\mathcal{O}(E)$ and seamlessly integrates graph structure learning into sparse training, thereby substantially lowering computational overhead while preserving predictive performance.

III. Methodology

A. Framework Overview

To address the challenges of *structural noise* and *high computational cost* encountered by graph neural networks (GNNs) in real-world scenarios, we propose a Dual-Dynamic Graph Neural Network framework. The proposed framework is built upon the original CGP model and mainly modifies its pruning and rewiring strategies [15].

Specifically, (1) we employ *cosine similarity* to measure node similarity and dynamically optimize the graph topology based on node semantic information; (2) we adopt *dynamic sparse training* to adaptively prune model weights during the training process.

Different from CGP, which performs pruning and rewiring only on the initially constructed graph, our method further introduces a *regrowth* mechanism to actively add missing yet semantically meaningful connections, enabling more flexible and effective topology evolution.

B. Metric-guided Graph Rewiring via Hybrid Sampling

The original graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ often contains noisy edges and lacks critical semantic connections. Directly training on such graphs can lead to suboptimal performance. Inspired by Structure Learning methods such as AdaEdge [16] and GAUG [17], we design a metric-based rewiring module [18]. This module consists of three stages:

Algorithm 1 Training Procedure of LGSF-AEG

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, Feature $\mathbf{X}$, Labels $\mathbf{Y}$; Hyperparameters: Rewire freq K, Pruning rate τ_{drop}, Growth rate τ_{add}, Initial/Final density d_{init}, d_{final}.
Ensure: Optimized Sparse Model parameters $\boldsymbol{\theta}^* \odot \mathbf{M}^*$ and Refined Graph $\tilde{\mathcal{G}}$.

Initialize model weights $\boldsymbol{\theta}$ randomly;
Initialize binary mask $\mathbf{M}$ using **ERK distribution**;
$\mathcal{G}^{(0)} \leftarrow \mathcal{G}$;
for epoch $t = 1$ to T_{max} **do**
  $\mathbf{Z} \leftarrow \text{GNN}(\mathcal{G}^{(t-1)}, \mathbf{X}; \boldsymbol{\theta} \odot \mathbf{M})$;
  $\mathcal{L} \leftarrow \text{Loss}(\mathbf{Z}, \mathbf{Y})$;
  Update $\boldsymbol{\theta} \leftarrow \text{Optimizer}(\mathcal{L}, \boldsymbol{\theta})$;
  Update target density d_t via **Linear Decay Schedule**;
  Update mask $\mathbf{M}$ via **Magnitude Pruning** to density d_t;
  if $t > T_{warm}$ **and** $t < T_{fix}$ **and** $t \mod K == 0$ **then**
    $\mathbf{H} \leftarrow \text{GetEmbeddings}(\mathcal{G}^{(t-1)})$;
    Compute Cosine Similarity Matrix $\mathbf{S}$;
    $\mathcal{E}_{drop} \leftarrow$ Identify edges where $S_{ij} < \text{threshold}(\tau_{drop})$;
    $\mathcal{E}_{cand} \leftarrow$ Generate candidates via **Hybrid Sampling** (2-hop + Random);
    $\mathcal{E}_{add} \leftarrow$ Select Top-k edges from $\mathcal{E}_{cand}$ based on $\mathbf{S}$;
    $\mathcal{E}^{(t)} \leftarrow (\mathcal{E}^{(t-1)} \setminus \mathcal{E}_{drop}) \cup \mathcal{E}_{add}$;
    $\mathcal{G}^{(t)} \leftarrow (\mathcal{V}, \mathcal{E}^{(t)})$;
  else
    $\mathcal{G}^{(t)} \leftarrow \mathcal{G}^{(t-1)}$;
  end if
end for
return Best model parameters and the final graph structure.

1) Metric Learning and Edge Pruning: At training epoch t, let the node embeddings generated by the encoder be $\mathbf{H}^{(t)} \in \mathbb{R}^{N \times d}$. We utilize cosine similarity as the metric to evaluate node affinity:

$$S_{ij} = \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\|_2 \, \|\mathbf{h}_j\|_2} \tag{1}$$

For the existing edge set $\mathcal{E}_{current}$, we remove edges where the similarity score is below a threshold τ_{drop} (i.e., the *Bottom-k* strategy). As demonstrated by Chen et al. [16], this feature-based pruning strategy effectively blocks noise propagation and alleviates the *Over-smoothing* problem at the topological level [19].

2) Hybrid Candidate Sampling: To avoid the prohibitive $\mathcal{O}(N^2)$ complexity of exhaustive similarity calculation (as utilized in IDGL) while maintaining representation quality, we propose a *Hybrid Candidate Sampling* strategy to generate the candidate edge set $\mathcal{E}_{cand}$:

- **2-Hop Neighborhood Sampling:** This design is mainly motivated by two key theoretical considerations.
  (1) Homophily Guarantee. AdaEdge observes that node pairs with small topological distances (i.e., reachable within only a few hops) are statistically more likely to

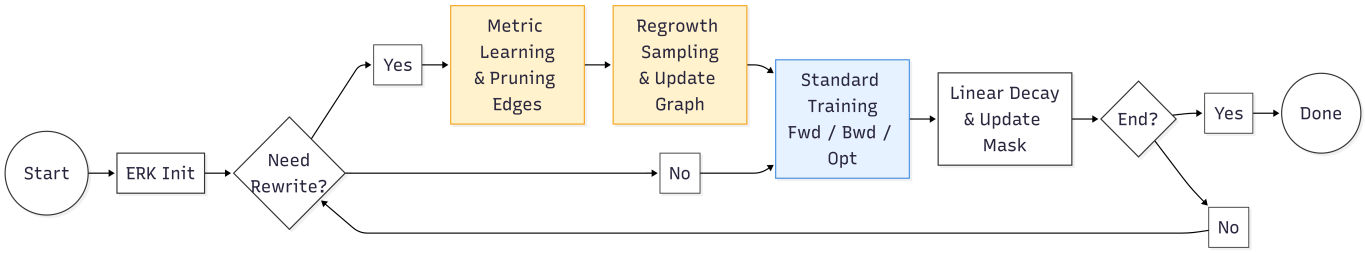

Fig. 2. **The Workflow of LGSF-AEG.** The framework proceeds from ERK initialization (left) to a dual-dynamic loop (right), where graph topology (orange path) and model sparsity (blue path) are jointly optimized.

belong to the same class, and vice versa [16]. Excessively large hop ranges tend to introduce unnecessary computational overhead and noise, while overly small hop ranges may fail to capture sufficient structural details [20]. Consequently, 2-hop neighborhoods strike a favorable balance and serve as high-quality candidates for semantic connection modeling.

(2) Feature Context. According to MixHop [21], higher-order neighborhoods provide complementary structural and semantic context beyond direct neighbor information. Under the widely accepted assumption that "friends of friends are also likely to be friends", 2-hop nodes exhibit strong potential for establishing semantically meaningful and efficient relational cues.

- **Random Negative Sampling:** To prevent the model from getting trapped in local optima, we randomly sample a ratio of non-connected node pairs from the global graph. This strategy introduces stochastic perturbations similar to DropEdge [22], preventing the model from converging to local structural optima and empowering it to capture *Long-range Dependencies*.

3) Topology Regrowth: After constructing the candidate set $\mathcal{E}_{cand} = \mathcal{E}_{2hop} \cup \mathcal{E}_{random}$, we compute the semantic similarity scores for each edge. To enhance the graph structure, we select the Top-k edges with the highest similarity to add, forming a new adjacency matrix $\tilde{\mathbf{A}}$:

$$\tilde{\mathcal{E}} = (\mathcal{E}_{current} \setminus \mathcal{E}_{drop}) \cup \text{Top-}k(\mathcal{E}_{cand}) \tag{2}$$

This "regrowth" mechanism enables the graph topology to update dynamically with the evolution of node features, thereby better serving downstream classification tasks. By restricting the candidate edge space to 2-hop neighbors and a linear ratio of random samples, we reduce the prohibitive $\mathcal{O}(N^2)$ complexity of standard GSL methods to $\mathcal{O}(E)$. Detailed theoretical derivations are provided in Section I of the Supplementary Material.

C. Dynamic Sparse Training Strategy

To reduce computational overhead and prevent overfitting, we adopt a Dynamic Sparse Training (DST) strategy to optimize the neural network parameters $\boldsymbol{\theta}$. As illustrated in Fig. 2, based on ERK-based sparse initialization and mask initialization, we perform a dual-dynamic training framework that jointly conducts dynamic graph structure rewiring and weight sparsification. Through iterative cycles of staged pruning and regrowth, this framework achieves efficient sparse GNN training while maintaining performance [23].

1) Sparse Initialization via ERK: Initializing sparse neural networks is inherently challenging. Conventional uniform initialization often leads to the so-called *layer collapse* problem, where layers with fewer parameters are excessively pruned, resulting in broken information flow paths and hindered gradient propagation. To address this issue, we adopt the **Erdős–Rényi–Kernel (ERK)** distribution [24] to initialize the binary mask $\mathbf{M} \in \{0,1\}^{|\boldsymbol{\theta}|}$. Its formal definition is given as:

$$1 - d_l \propto \frac{n_{l-1} + n_l + w + h}{n_{l-1} \cdot n_l \cdot w \cdot h}, \tag{3}$$

where d_l denotes the parameter density of the l-th layer, and n represents the channel dimensions [25].

The ERK scheme adaptively adjusts the sparsity of each layer according to its dimensionality, allocating higher connection density to critical and relatively narrow layers at initialization. In this way, ERK not only retains more parameters but also preserves the integrity of information flow paths, effectively alleviating the gradient vanishing issues that may arise from uniform sparse initialization [26].

2) Magnitude Pruning with Linear Annealing: Inspired by the Lottery Ticket Hypothesis, we adopt an iterative mask update strategy based on *magnitude pruning*, where weights with small absolute values are progressively removed as they are deemed less contributory to the model output. Instead of fixing the sparsity level throughout training, we employ a *linear decay schedule* to dynamically adjust the network density.

Let d_{init} and d_{final} denote the initial and target final densities, respectively. The target density d_t at training epoch t is defined as:

$$d_t = d_{init} + (d_{final} - d_{init}) \cdot \frac{t - t_{start}}{t_{end} - t_{start}}, \tag{4}$$

where t_{start} and t_{end} specify the starting and ending epochs of the pruning process.

This strategy offers two key advantages. First, the high-density model at early training stages enables effective exploration of the parameter space, facilitating the identification of potential *winning subnetworks*. Second, as training progresses, the network density linearly decays toward d_{final}, forcing the

model to consolidate information into a compact and highly sparse structure.

3) Joint Training Protocol: We divide the training strategy into three stages:

1. **Warmup Phase**: The original graph structure is kept fixed while only the model weights are updated, allowing node embeddings to acquire preliminary semantic representations.

2. **Dynamic Evolution Phase**: Periodic graph rewiring is performed every K epochs, concurrently with sparse updates of the model weights.

3. **Fix & Fine-tune Phase**: Graph rewiring is halted, and the final optimized topology $\tilde{A}_{\text{final}}$ along with the sparse mask is fixed. The model then undergoes final fine-tuning until convergence.

This strategy ensures the stability of both structural learning and parameter optimization. The detailed process is illustrated in Fig. 3.

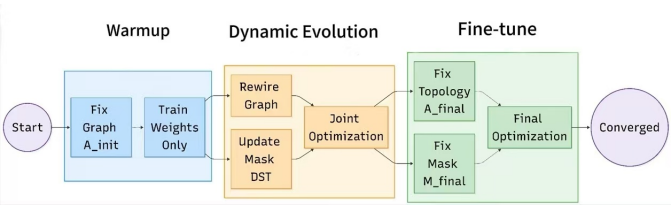

Fig. 3. Joint Training Protocol

IV. Experiments

A. Experimental Setup

All models are trained for 200 epochs with a fixed learning rate of 0.01. Experiments are conducted on four representative benchmark datasets: Cora, Citeseer, Pubmed, and the large-scale Coauthor Physics dataset . We adopt a dense GCN without any sparsification as the first baseline (Control Group I), compare it with the CGP method as the second baseline (Control Group II), and treat the proposed LGSF-AEG method as the experimental group. For LGSF-AEG, graph rewiring is enabled after a warmup phase. In the standard setting for Cora, Citeseer, and Pubmed, the warmup phase is 20 epochs, and rewiring is performed every 20 epochs with both edge removal and addition rates set to 0.03. To preserve the high-quality initial topology of the large-scale Physics dataset, we adopt a more conservative adaptive strategy: the warmup phase is extended to 50 epochs, and the rewiring rate is reduced to 0.01. All reported results are obtained by averaging over 3 runs with different random seeds.

B. Main Results

The average test accuracy (%) across the three benchmark datasets is reported in Table I. In addition to test accuracy, we further compute the Macro-F1 score to evaluate the model's robustness under class-imbalanced settings, ensuring that the observed performance gains are not dominated by majority classes.

According to Table I, LGSF-AEG achieves the best performance on Cora and Pubmed. On the Pubmed dataset, LGSF-AEG attains a peak accuracy of 80.0% and a Macro-F1 of 79.6%, significantly outperforming the static CGP method. On the large-scale Physics dataset, LGSF-AEG demonstrates exceptional scalability. While CGP maintains a slight edge in raw accuracy, our method achieves the highest Macro-F1 score (91.5%), indicating that metric-guided rewiring effectively captures class-specific semantic features even in massive, high-homophily graphs.

However, on Citeseer, LGSF-AEG (70.8%) is slightly lower than the dense GCN. This performance degradation is attributed to Citeseer's inherently fragmented graph structure, where aggressive pruning may inadvertently damage critical local connectivity. To further explore this issue and evaluate the quality of the learned node representations, we conducted dedicated supplementary experiments and t-SNE visualization analyses, as detailed in Sections II and III of the Supplementary Material. As demonstrated in the supplementary results, while Dense GCN and CGP exhibit blurred inter-class boundaries and significant cluster overlap, LGSF-AEG consistently produces more distinct clusters with clearer separation margins on Cora and Pubmed. This qualitative evidence confirms that our structural rewiring mechanism remains essential for learning highly discriminative semantic representations, effectively capturing underlying class structures even when the global connectivity is challenged.

C. Resilience to Structural Perturbations

To evaluate the robustness of LGSF-AEG against structural noise, we deliberately introduce random noisy edges into the datasets with noise rates ranging from 0% to 10%. Each model is trained multiple times under each noise level, and the averaged results are reported. Detailed analysis and corresponding visualizations are provided in Section IV of the Supplementary Material.

D. Efficiency Analysis

To evaluate the model performance under different parameter budgets, we conduct experiments with parameter densities ranging from 5% to 50%, and compare the test accuracy of LGSF-AEG and CGP. The dashed gray line denotes the performance of the unpruned dense GCN (i.e., 100% of parameters). The experimental results and corresponding analysis are provided in Section V of the Supplementary Material.

V. Conclusion

This paper proposes a novel dual dynamic graph neural network framework, termed LGSF-AEG, which aims to simultaneously address the challenges of structural noise and high computational cost in graph neural networks. Overall, LGSF-AEG demonstrates superior performance compared to the static sparse baseline. LGSF-AEG adopts a co-evolutionary paradigm of graph topology and model parameters, integrating structure learning with sparse training to explore a new

TABLE I
PERFORMANCE COMPARISON (ACCURACY & MACRO-F1) ON FOUR DATASETS. "OURS" DENOTES LGSF-AEG. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	Config	Cora		Citeseer		Pubmed		Physics (Large)	
		Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)
Dense GCN	Fully Connected	81.6	80.6	71.8	68.5	79.0	78.7	93.2	91.0
CGP (Static)	Sparse Weights	81.7	80.5	71.6	68.4	79.2	78.8	93.5	91.2
LGSF-AEG (Ours)	Dual-Sparse	81.9	80.8	70.8	67.5	80.0	79.6	93.4	91.5

training methodology for GNNs. By introducing homophily-guided pruning and a hybrid-sampling-based growth strategy, LGSF-AEG successfully overcomes the low-rank assumption limitations and computational bottlenecks of traditional graph structure learning methods (e.g., IDGL), achieving efficient denoising and semantic completion with a computational complexity of $\mathcal{O}(E)$.

Experimental results on three benchmark datasets demonstrate that LGSF-AEG consistently outperforms static sparse baselines and achieves notable performance improvements on the large-scale Pubmed graph, thereby validating the necessity of structural optimization. Meanwhile, the adaptive "prune–grow" mechanism designed for sparse graph scenarios enables flexible structural adjustment. Furthermore, ablation studies and sensitivity analyses indicate that the dual-process rewiring mechanism is indispensable and that the proposed framework exhibits high stability with respect to hyperparameter variations. In future work, we plan to extend LGSF-AEG to larger-scale heterogeneous graphs and explore its integration with self-supervised learning paradigms.

Supplementary Material is available at: https://github.com/vivid-css/LGSF-AEG

REFERENCES

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[3] P. Veličković *et al.*, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[4] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] Y. Zhu, W. Xu, F. Liu, and Q. Liu, "Deep graph structure learning for robust representations: A survey," *arXiv preprint arXiv:2103.03036*, 2021.

[6] C. Liu, H. Xue, K. Zhou, W. Hu, B. Ling, and F. Wang, "Comprehensive graph gradual pruning for sparse training in graph neural networks," *arXiv preprint arXiv:2207.08470*, 2022, source code: https://github.com/LiuChuang0059/CGP.

[7] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations (ICLR)*, 2019.

[8] B. Li *et al.*, "Sparsified graph convolutional networks," *arXiv preprint arXiv:1912.06248*, 2019.

[9] Y. Chen, K. Zheng, Y. Zhang, J. Zhao, M. Li, and J. Tang, "A unified lottery ticket hypothesis for graph neural networks," in *International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 1695–1706.

[10] T. Zhang, J. Cheng, L. Miao, H. Chen, Q. Li, and Q. He, "Multi-hop reasoning with relation based node quality evaluation for sparse medical knowledge graph," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[11] L. Franceschi *et al.*, "Learning discrete structures for graph neural networks," in *International Conference on Machine Learning (ICML)*, 2019.

[12] W. Jin *et al.*, "Graph structure learning for robust graph neural networks," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2020.

[13] B. Fatemi *et al.*, "Slaps: Self-supervision co-training for better graph structure learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[14] Y. Chen, L. Wu, and M. Zaki, "Iterative deep graph learning for graph neural networks: Better and robust node embeddings," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 19 314–19 326.

[15] Y. Zhang, R. Jiao, B. Xue, and M. Zhang, "Evolutionary streaming feature selection via incremental feature clustering," *IEEE Transactions on Evolutionary Computation*, 2026.

[16] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, no. 04, 2020, pp. 3438–3445.

[17] T. Zhao, Y. Liu, L. Neves, O. Woodford, M. Jiang, and N. Shah, "Data augmentation for graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, no. 17, 2021, pp. 14 820–14 828.

[18] Z. Chen, Q. Fan, R. Jiao, B. Xue, H. Huang, Y. Dai, and M. Zhang, "Multi-expert genetic programming based ensemble for long-tailed image classification," *IEEE Transactions on Evolutionary Computation*, 2026.

[19] N. Li, L. Ma, G. Yu, B. Xue, M. Zhang, and Y. Jin, "Survey on evolutionary deep learning: Principles, algorithms, applications and open issues," *arXiv preprint arXiv:2208.10658*, 2022.

[20] N. Li, B. Xue, L. Ma, and M. Zhang, "Automatic fuzzy architecture design for defect detection via classifier-assisted multiobjective optimization approach," *IEEE Transactions on Evolutionary Computation*, 2025.

[21] S. Abu-El-Haija *et al.*, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *International Conference on Machine Learning (ICML)*, 2019.

[22] Y. Rong *et al.*, "Dropedge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations (ICLR)*, 2020.

[23] T. Zhang, L. Ma, S. Cheng, Y. Liu, N. Li, and H. Wang, "Automatic prompt design via particle swarm optimization driven LLM for efficient medical information extraction," *Swarm and Evolutionary Computation*, vol. 95, p. 101922, 2025.

[24] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, "Rigging the lottery: Making all tickets winners," in *International Conference on Machine Learning (ICML)*, 2020.

[25] N. Li, B. Xue, L. Ma, and M. Zhang, "Transferable relativistic predictor: Mitigating cross-task cold-start issue in NAS," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)*, 2025.

[26] A. Mei, N. Li, L. Ma, K. Gao, M. Huang, and B. Xue, "Binary-to-decimal encoding-based performance predictor for evolutionary graph fusion architecture search and applications to medical data," *IEEE Transactions on Evolutionary Computation*, 2024.