Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 12, No. 01, March 2026.

GEA-CoPe: An Effective Model for Cross-domain Graph Pre-training

Yiming Zhao and Yongqing Wu

(Received: 10-Sep.-2025, Revised: 21-Nov.-2025, 10-Dec.-2025 and 13-Jan.-2026, Accepted: 14-Jan.-2026)

Abstract

This paper addresses the negative transfer problem in cross-domain graph pre-training under few-shot learning scenarios, it proposes a multi-component pre-training framework called Graph External Attention-enhanced Coordinators for Pre-training (GEA-CoPe). This framework integrates multi-head external attention with a graph coordinator. Tackling the structural and semantic discrepancies between cross-domain graphs is crucial for mitigating negative transfer; however, conventional methods often lack adaptability to complex, dynamic inter-domain variations and explicit constraints for intermediate feature-distribution consistency. The proposed framework leverages an external attention-based coordinator to mediate between different graph datasets, dynamically generating cross-graph semantic-alignment strategies to alleviate negative transfer induced by structural heterogeneity. It employs a dual-feature normalization strategy that incorporates a cross-layer distribution alignment loss on top of intra-layer node-similarity constraints, effectively suppressing feature drift. Furthermore, Kolmogorov-Arnold Networks (KANs) are introduced, whose parameter-adaptive activation functions better capture non-linear topological dependencies and enhance model interpretability. Experiments on ten real-world graph datasets demonstrate that GEA-CoPe exhibits superior cross-domain generalization capability and significantly improves performance in few-shot node classification tasks, with an average improvement of about 13.3% *compared to other methods. The model can more accurately focus on critical graph structures, providing a theoretical foundation and practical paradigms for deploying graph neural networks in complex scenarios.*

Keywords

Graph neural networks, Graph pre-training, Transfer learning, External attention.

1. INTRODUCTION

In recent years, in the fields of natural-language processing and computer vision, foundation models based on the Transformer architecture have acquired powerful general representation capabilities through pre-training on massive unlabeled data [1]. Subsequently, they can quickly adapt to various downstream tasks with minimal annotated data *via* fine-tuning, establishing a new "pre-training + fine-tuning" paradigm [2]. The success of this paradigm reveals the great potential of learning universal knowledge from large-scale data and transferring it to specific tasks. Inspired by this, the graph-learning community has also embarked on exploring the construction of "graph-foundation models," with cross-domain graph pre-training as their core component [3]. Cross-domain graph learning aims to train a universal graph encoder by integrating graph data from multiple sources with diverse structures and features, enabling it to learn transferable graph structural patterns and semantic knowledge transcending individual domains [4].

However, achieving this vision faces severe challenges. Real-world graph data exhibits extremely high heterogeneity. First, structurally, graphs from different domains may possess entirely distinct topological properties. For example, citation networks are typically homophilic [5], where connected nodes tend to belong to similar categories, whereas molecular networks or fraud-detection networks are often heterophilic, with connected nodes likely belonging to different categories. Second, at the feature level, node feature dimensions, physical meanings and distributions can vary significantly across different graphs [6]. This dual discrepancy in both structure and features makes effective knowledge transfer across different graph domains exceptionally difficult. Models are highly prone to learning knowledge on the source domain that cannot be applied to the target domain or even resulting in negative transfer [7]-[8]. Therefore, conducting research on cross-domain graph pre-training, exploring how to overcome the heterogeneity of graph data and building graph representation models capable of capturing universal patterns across domains, not only holds significant theoretical value, but is also an urgent

Y.-M. Zhao and Y.-Q. Wu (Corresponding Author) are with School of Software, Liaoning Technical University, Liaoning, China. Emails: 3130637271@qq.com and yqwuyywu@163.com

requirement for advancing graph intelligence technologies toward real-world applications.

Despite significant advances in cross-domain graph learning, existing methods still exhibit limitations when dealing with complex real-world graph data. Structure-oriented approaches [9]-[10] focus on mining commonalities in graph topology to achieve transferability through contrastive learning or structure generation. However, they often overlook the rich semantic information carried by node and edge features. When both the structure and the feature semantics differ significantly between the source and target domains, relying solely on structural similarity can lead to severe negative transfer. Feature-oriented methods [11]-[12] aim to align the feature spaces of different graph domains. Yet, they typically require consistent feature dimensions or depend on textual descriptions, which greatly restricts their applicability. For graph data with different feature dimensions or lacking explicit semantic annotations, feature alignment becomes particularly challenging. Hybrid approaches [13]-[15] often combine structure and feature information in a simple, sequential manner, failing to achieve deep and organic integration of both aspects. Furthermore, they struggle to effectively model global semantic relationships across graphs during training and are susceptible to feature-distribution shifts in deep networks, resulting in inefficient knowledge transfer and unstable model performance.

To address the aforementioned challenges, this paper proposes GEA-CoPe- an effective multi-component pre-training framework designed to alleviate negative transfer and feature drift in cross-domain graph learning. The framework demonstrates exceptional cross-domain generalization capability and significantly improves performance in few-shot node-classification tasks, achieving an average performance gain of approximately 13.3% compared to existing methods. It can be directly applied to cross-domain few-shot learning scenarios, such as transferring knowledge from a well-annotated citation network to classify nodes in a new social network or adapting a model from one e-commerce platform to another for user-interest recognition. Moreover, the framework serves as a robust foundational model for various downstream graph analytical tasks, particularly in target domains with limited supervisory signals. The main contributions of this work can be summarized as follows:

- An effective cross-domain graph pre-training framework is proposed. By leveraging a dynamic coordinator mechanism based on graph external attention, the model can implicitly learn deep semantic relationships across different graph domains. Through the dynamic interaction between coordinator nodes and external memory, it adaptively generates cross-graph semantic-alignment strategies, thereby effectively bridging domain gaps while preserving unique structural information of each graph, fundamentally mitigating negative transfer.
- A dual contrastive normalization module is designed to address feature drift in deep graph networks. It constrains feature smoothness among nodes within the same layer and ensures feature consistency during propagation through cross-layer distribution-alignment loss, enhancing the domain robustness and stability of pre-trained representations.
- In the downstream task-adaptation phase, Kolmogorov-Arnold Networks are introduced to replace traditional classification heads in cross-domain graph learning. With their superior non-linear fitting capability and higher parameter efficiency, KANs can better capture complex graph patterns and feature interactions, further improving the model's adaptability and generalization performance on target domains.
- Experiments were conducted on 10 datasets and the results proved the model's superiority.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 elaborates on the proposed GEA-CoPe framework, which is structured into the pre-training phase and the transfer-learning phase. Section 4 describes the experimental setup and evaluation metrics, followed by a detailed presentation of the results. Finally, Section 5 concludes the paper and discusses its limitations along with potential directions for future research.

2. RELATED WORK

2.1 Graph Pre-training

Graph pre-training has emerged as a promising paradigm in graph machine learning. Its core idea is to leverage self-supervised learning on large-scale unlabeled graph data to capture universal structural and attribute patterns, thereby providing a well-initialized model with rich knowledge and strong

generalization capability for downstream tasks. Typical pre-training strategies include node-level tasks, such as masked attribute reconstruction and context prediction, as well as graph-level objectives, like graph structure contrastive learning and property prediction. These tasks are designed to enable the model to deeply comprehend the complex dependencies among graph elements. Through such pre-training, the model can learn inherent and transferable domain knowledge, which significantly reduces dependence on labeled data in downstream tasks and effectively enhances generalization performance, convergence speed and final task performance. Current research primarily focuses on the following directions:

Contrastive Learning-based Graph Pre-training. Maximizes mutual information (MI) between graph structures or node sub-graphs to enhance the model's understanding of local and global feature correlations. For example, GraphCL [16] employs graph-augmentation strategies to generate multi-view contrastive samples, while SimGRACE [17] constructs positive-negative sample pairs *via* parameter perturbation to optimize node-level contrastive loss. These methods exhibit strong generalizability in molecular-property prediction and social-network analysis, but remain limited in modeling topological invariance for heterophilic graphs.

Generative Graph Pre-training. Forces models to learn the distribution patterns of graph data by reconstructing masked node attributes, edge connections or sub-graph structures. Representative methods include GPT-GNN [18], which adopts an auto-regressive approach to generate nodes and edges and GraphMAE [19], which introduces a masked auto-encoder to reconstruct node features. These methods excel in protein-interaction prediction but show low efficiency in reconstructing complex high-order relationships.

Cross-domain Universal Graph Pre-training Frameworks. For unified representation learning on multi-source heterogeneous graphs, recent studies proposed hierarchical contrastive pre-training [20], which separate domain-specific and shared features to enhance transferability in cross-domain tasks, like biomedicine and recommendation systems. However, challenges remain in integrating knowledge from large-scale heterogeneous graphs and adapting to temporal evolution in dynamic graphs.

2.2 Graph Transfer Learning

Graph transfer learning aims to transfer structural knowledge and semantic patterns learned from a source-graph domain to a target-graph domain to mitigate performance degradation caused by target-domain data scarcity or domain shifts. Its core challenge lies in aligning cross-domain topological heterogeneity and extracting domain-invariant representations. Recent research directions include:

Domain Adaptation-based Graph Transfer. Reduces structural discrepancies between source and target domains *via* adversarial training or distribution alignment. For instance, [21] introduced a graph convolutional adversarial framework that jointly aligns node features and topological structures by minimizing domain divergence through Wasserstein distance constraints, while [22] formalized Fused Gromov-Wasserstein distance for structured graph alignment, providing theoretical foundations for minimizing inter-domain Wasserstein distances. These methods perform robustly in cross-social network user-behavior prediction, but struggle to adapt to temporal dynamics in dynamic graphs.

Heterogeneous Graph Representation Transfer. Meta-path-aware transfer frameworks address node/edge type heterogeneity. The Heterogeneous Graph Transformer [23] dynamically adjusts relation-specific attention weights through meta-relation aware mechanisms, while [24] employs reinforcement learning for automated meta-path discovery across domains. These methods demonstrate effectiveness in cross-platform recommendation tasks without relying on predefined-schema constraints, as validated in Amazon eBay product alignment experiments.

Dynamic Graph Temporal Transfer. Recent advances handle structural evolution through temporal modeling. [25] decouples graph-convolution parameters into temporal trajectories using RNNs, capturing both topological persistence and variation patterns. [26] implements continuous-time graph representation learning *via* temporal point processes, effectively addressing domain shifts in financial-transaction networks with adaptive computation.

Unsupervised Cross-graph Transfer. Domain-invariant feature-learning methods achieve progress through novel objectives. DANE [27] disentangles domain-specific variations *via* adversarial alignment of graph embeddings, while Graph Optimal Transport [28] maximizes feature correspondence through

Wasserstein-distance minimization. These approaches show superior performance in cross-organism protein network analysis with explicit geometric-alignment constraints.

3. METHOD

Our pre-training dataset consists of M graphs, represented as $\mathcal{G}^{(i)} = \left(\mathcal{V}^{(i)}, \mathcal{E}^{(i)}\right)$, where $i \in \{1,2,\dots,M\}$, respectively. $\mathcal{V}^{(i)} = \left\{v_1^{(i)}, v_2^{(i)}, \dots, v_{|\mathcal{V}^{(i)}|}^{(i)}\right\}$ and $\mathcal{E}^{(i)} = \mathcal{V}^{(i)} \times \mathcal{V}^{(i)}$ represent the node sets and edge sets, respectively. Each $\mathcal{G}^{(i)}$ graph is associated with a feature matrix $X^{(i)} \in \mathbb{R}^{|\mathcal{V}^{(i)}| \times d_i}$, $E^{(i)} \in \mathbb{R}^{|\mathcal{E}^{(i)}| \times d_i}$ and an adjacency matrix $A^{(i)} \in \mathbb{R}^{|\mathcal{V}^{(i)}| \times |\mathcal{V}^{(i)}|}$. The main goal is to train a graph neural network(GNN) $h(\cdot)$ with learnable parameter Θ that captures domain-agnostic knowledge for adaptation to downstream applications. The downstream dataset is represented as $\mathcal{G}^{(t)} = \left(V^{(t)}, \mathcal{E}^{(t)}\right)$ with the feature matrix $X^{(t)}$ and adjacency matrix $A^{(t)}$.

3.1 Overview of Our Framework

In this sub-section, the proposed GEA-CoPe model is described in detail, which consists of two phases. In the first phase, pre-training is conducted on multiple cross-domain graph datasets and the established pre-training frameworks GraphCL [16] and SimGRACE [17] are used to guide the entire process. The second phase implements transfer learning to adapt the pre-trained knowledge to downstream tasks for addressing diverse applications. Several novel techniques are introduced to address the aforementioned issues and challenges, with the overall framework illustrated in Figure 1.

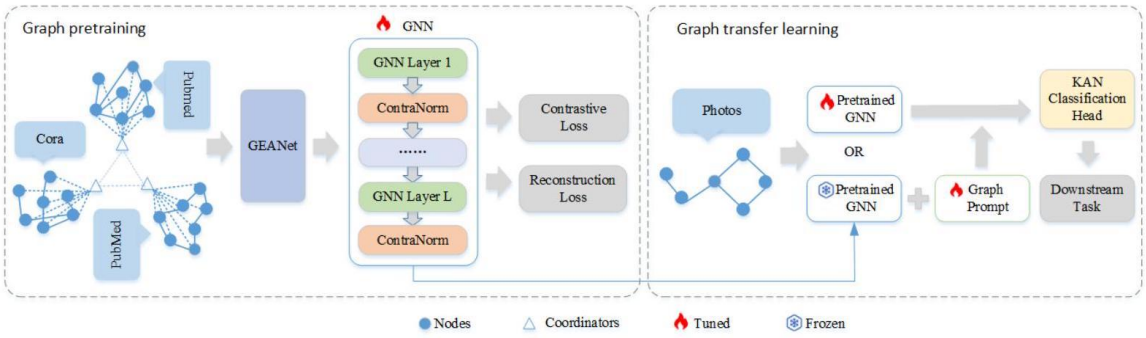


Figure 1. Overall framework of GEA-CoPe model. The first half is the graph pre-training stage and the second half is the graph transfer-learning stage.

3.2 Aligning Graphs by Coordinators

To address the heterogeneous feature representations and topological disparities across graph data, this paper uses an alignment framework. This architecture comprises two core stages: First, feature-space standardization transforms heterogeneous node features into a unified dimensional space through linear projections. Subsequently, a dynamic coordinator mechanism introduces learnable virtual nodes to establish cross-graph semantic correlations, enabling dual-level adaptive alignment of structural patterns and semantic relationships. This phased methodology systematically resolves both shallow feature-distribution discrepancies and deep pattern-expression variations in cross-domain graph datasets.

3.2.1 Data Pre-processing

During the data pre-processing stage, a series of crucial steps is performed to ensure the effectiveness of cross-domain graph pre-training. First, raw graph data is loaded from ten standard graph datasets and uniformly converted into a data object format, achieving standardized integration of multi-source data. Subsequently, data-cleaning operations are executed to remove pre-defined data-split masks (including training, validation and test-set identifiers) from the datasets. This step effectively prevents potential data-leakage risks and provides a clean data foundation for subsequently constructing a unified cross-dataset pre-training paradigm. Next, unified dimensionality processing is applied to each graph. When the original feature dimensionality (e.g. 1433 dimensions for Cora) is higher than the target dimensionality (100 dimensions), Singular Value Decomposition (SVD) is employed for dimensionality reduction, preserving over 95% of the variance. When the original feature dimensionality is lower than

the target, zero-padding is performed to ensure consistent node feature dimensions across all graphs. Following this, multiple independent graphs are merged into a unified large graph. For each original graph, a learnable coordinator node is added, generating learnable features. Edge-connection strategies include static full connection (increasing the edge count by 57.1%) and dynamic similarity-based connection. When dynamically adding edges based on node feature cosine similarity, only edges with a similarity above a threshold (default 0.1) are retained to control sparsity (resulting in edge-count increases ranging from 7.6% to 56.8%). Finally, sub-graphs are sampled from the large graph *via* a random walk algorithm, with a walk length of 30 and starting nodes comprising 10% of the total, covering $70\% - 85\%$ of all nodes. Sub-graphs with fewer than 5 nodes are filtered out to ensure sample quality, forming the final collection of sub-graphs for subsequent learning.

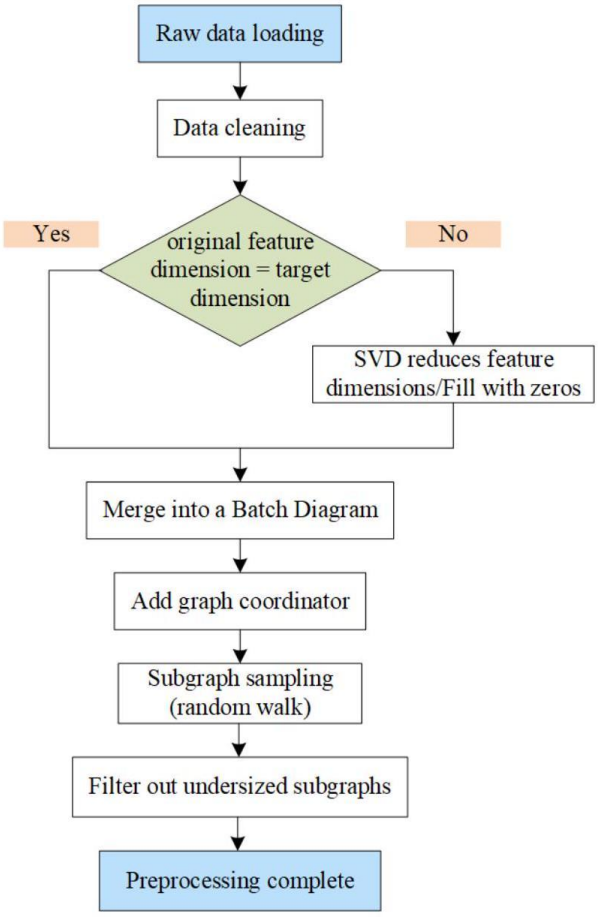


Figure 2. Data pre-processing flowchart.

The data pre-processing flowchart is illustrated in Figure 2. These pre-processing steps collectively form a systematic pipeline that transforms multiple graph datasets into a structurally unified and feature-aligned collection of sub-graphs. This pipeline provides a solid data foundation for the subsequent cross-domain graph pre-training model, ensuring the model's robustness and generalization capability when processing graph data.

3.2.2 Feature Projection

In the first stage of the method; namely, the pre-training phase, the initial step involves processing the data to align the feature dimensions across different domains, as shown in Figure 3. This is achieved through a projection module, with the specific implementation as follows:

$$\tilde{X}^{(i)} = \mathrm{Proj}\left(X^{(i)}\right) \in \mathbb{R}^{|v^{(i)}| \times dp}, \tag{1}$$

where $\mathrm{Proj}()$ denotes the projection operation and d_p denotes the pre-defined projected dimension. In this paper, the widely-addressed singular value decomposition (SVD) is employed for the projection operation. However, to address the feature-alignment problem, merely applying feature projection to the data is insufficient; additional calibration processes are required to further calibrate the data.

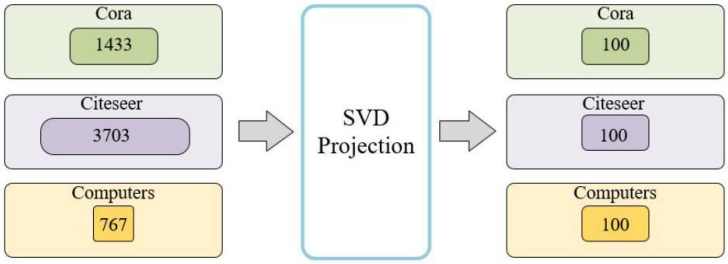


Figure 3. SVD feature projection.

3.2.3 Graph Coordinators

Following the feature-projection stage described above, a "coordinator" - virtual node is introduced, which is designed to bridge graphs from different domains and enhance feature and structural alignment.

Coordinator-Graph Connection. Considering that datasets originate from distinct domains where each graph exhibits unique structural properties and information flows, in order to preserve the intrinsic structural characteristics of individual graphs while enabling their participation in cross-graph information exchange, a dedicated coordinator is established for each dataset. Rather than being isolated from the graph data, each coordinator is fully connected to all nodes within its associated graph, forming a new sub-graph that becomes an integral part of the original graph. This design creates direct and efficient communication pathways between the coordinator and nodes, allowing the coordinator to effectively gather node-level information and facilitate coordinated interactions.

Coordinator-Coordinator Connection. Since our objective focuses on enabling cross-domain knowledge sharing rather than enhancing individual graph representations, inter-coordinator connections are established to serve as bridges for inter-graph communication. Specifically, edges are introduced between coordinators originally assigned to different graph datasets, thereby constructing inter-connected channels for global information exchange. This eliminates data isolation and creates a unified platform for comprehensive knowledge sharing across all domains. Through these operations, a joint adjacency matrix is constructed, including the original graph adjacency matrix and the newly added coordinator connection. The formula is:

$$\tilde{A} = \begin{bmatrix} A_{\text{diag}} & R_A^T \\ R_A & R_R \end{bmatrix} \tag{2}$$

where $A_{\text{diag}} = \text{Diag}\left(A^{(1)}, A^{(2)}, \ldots, A^{(M)}\right), R_A = \text{Stack}\left(R_A^{(1)}, R_A^{(2)}, \cdots, R_A^{(M)}\right), R_R = 1^{M\times M}$. Diag means concatenating matrices diagonally and Stack means stacking row-vectors into a matrix. $R_A^{(i)} \in \mathbb{R}^N$, the j th value of $R_A^{(i)}$:

$$N = \sum_{k}^{M} \left|v^{(k)}\right|, \tag{3}$$

$$R_A^{(i)}(j) = \begin{cases} 1, \sum_{1}^{i} \left|v^{(k)}\right| \le j < \sum_{1}^{i+1} \left|v^{(k)}\right|, \\ 0, \text{otherwise}. \end{cases} \tag{4}$$

The coordinator representation serves as a learnable parameter that can be trained jointly with GNNs. Through an end-to-end collaborative training design, adaptive units dynamically calibrate their topological connection weights and feature-aggregation patterns according to the underlying data distribution. This collaborative training mechanism ensures that the coordinator continuously self-improves as an information bridge, enabling more effective transmission of cross-domain graph knowledge.

Generate Graph Batches for Efficient Training. By strategically leveraging coordination mechanisms to bridge disparate graph structures, this framework implements cross-graph node sampling during training iterations. Such synergistic processing enhances pre-training through batch-level knowledge amalgamation while promoting cross-dataset feature alignment. The co-optimization paradigm compels the model to distill topological regularities transcending individual graph boundaries, thereby deriving unified latent representations that comprehensively synthesize graph information from diverse domains. The cross-fusion of graph characteristics during parameter updates establishes an inductive bias favoring

the extraction of fundamental relational patterns while simultaneously facilitating the advancement of cross-domain graph-learning frameworks through coordinated structural integration.

3.3 Graph External Attention

The self-attention mechanism assumes that the input graph is fully connected. Initially, each element of the input sequence is transformed into vector representations *via* an embedding layer. Each input vector is then linearly projected to generate three vectors: the query (Q), which explores correlations with other positions; the key (K), which is matched by queries from other positions; and the value (V), which stores the actual information to be aggregated. These operations are formally expressed as:

$$\mathrm{A}_{\mathrm{Self}} = \mathrm{softmax}\left(\frac{\mathrm{QK}^T}{\sqrt{d_{\mathrm{out}}}}\right) \in \mathbb{R}^{|v^{(i)}|\times|v^{(i)}|} \tag{5}$$

$$\mathrm{Self}-\mathrm{Attn}(\mathrm{X}) = \mathrm{A}_{\mathrm{Self}}\mathrm{V} \in \mathbb{R}^{|v^{(i)}|\times d_{\mathrm{out}}} \tag{6}$$

where, W_Q, W_K, W_V represent trainable parameters and d_{out} denotes the dimension of Q. Subsequently, attention scores are computed to perform weighted aggregation, where positions with higher relevance are assigned greater weights. However, conventional self-attention mechanisms predominantly focus on node features within a single graph and capture only superficial associations between nodes, which limits their functional capabilities.

Inspired by [29], a graph external attention network is introduced, which not only attends to node features within individual graphs, but also incorporates external units. By computing attention between these external units and the node features of the input graph, the proposed method enhances graph representation learning. This approach achieves:

$$\mathrm{A}_{GE} = \mathrm{norm}(\mathrm{XU}^T) \in \mathbb{R}^{|v^{(i)}|\times S}, \tag{7}$$

$$\mathrm{GE}-\mathrm{Attn}(\mathrm{X}) = \mathrm{A}_{GE}\mathrm{U} \in \mathbb{R}^{|v^{(i)}|\times d_i}, \tag{8}$$

where, $\mathrm{U} \in \mathbb{R}^{S\times d_i}$ as external units, is designed as learnable parameters containing S nodes, with the information being shared across all input graph data. A_{GE} denotes the similarity between the input-graph nodes and the external units. Subsequently, normalization operations [47] are applied to A_{GE}, specifically performing row-wise and column-wise normalization, respectively. To elaborate, the normalization is implemented by:

$$\tilde{\alpha}_{i,j} = (\mathbf{XU}^T)_{i,j}, \tag{9}$$

$$\hat{\alpha}_{i,j} = \frac{\exp(\tilde{\alpha}_{i,j})}{\sum_{k=0}^{n}\exp(\tilde{\alpha}_{k,j})}, \tag{10}$$

$$\alpha_{i,j} = \frac{\hat{\alpha}_{ij}}{\sum_{k=0}^{S}\hat{\alpha}_{i,k}}. \tag{11}$$

In specific implementations, to achieve enhanced performance, two external modules are used to store keys and values respectively. Furthermore, a separate external module is utilized to process edge features within the input graph, while node-edge connectivity information is incorporated into a shared module.

$$\mathrm{X}_{out} = \mathrm{norm}\left(\mathrm{XU}_s\mathrm{U}_{nk}^T\right)\mathrm{U}_{nv}, \tag{12}$$

$$\mathrm{E}_{out} = \mathrm{norm}\left(\mathrm{EU}_s\mathrm{U}_{ek}^T\right)\mathrm{U}_{ev}, \tag{13}$$

where $\mathrm{U}_s \in \mathbb{R}^{|v^{(i)}|\times|v^{(i)}|}$ represents shared units; $\mathrm{U}_{nk}, \mathrm{U}_{nv} \in \mathbb{R}^{S\times d_i}$ is the external unit for storage nodes, while $\mathrm{U}_{ek}, \mathrm{U}_{ev} \in \mathbb{R}^{S\times d_i}$ is the external unit for storage edges.

The multi-head self-attention mechanism serves as the core component of Transformer models, with its fundamental principle being the processing of input sequences through multiple parallel self-attention modules followed by result integration to enhance the model's expressive power. For instance, both the node-node relationships within a graph and the node-external unit relationships exhibit complex diversity. Therefore, the processing analogous to the multi-head self-attention mechanism is adopted:

$$h_i = \mathrm{GE}-\mathrm{Attn}(\mathrm{X}_i, \mathrm{U}_{nk}, \mathrm{U}_{nv}) \tag{14}$$

$$\mathrm{X}_{out} = \mathrm{MultiHeadGEA}\,(\mathrm{X}, \mathrm{U}_{nk}, \mathrm{U}_{nv}) = \mathrm{Concat}(h_1, \ldots, h_H)\mathrm{W}_o \tag{15}$$

where $\mathbf{U}_{nk}, \mathbf{U}_{nv} \in \mathbb{R}^{S \times d_i}$ denotes the memory unit shared by all heads. h_i represents the i-th head, H represents the total number of heads and W_o is a linear transformation matrix Finally, a skip connection is applied to the output.

3.4 Pre-training on Multi-domain Graphs

This paper proposes a universal cross-domain graph pre-training framework compatible with various pre-training methods, which generates more expressive embeddings at both node and graph levels. Existing works predominantly focus on paradigms utilizing homogeneous data domains for pre-training [30]. GraphCL [16] systematically constructs positive sample pairs through structured graph data-augmentation strategies, explicitly enhancing data diversity to guide models in learning invariant features while maximizing mutual information between augmented and original samples through contrastive loss. SimGRACE [17] directly generates positive sample pairs by applying subtle perturbations to GNN encoder parameters. Since parameter perturbations preserve the topological connectivity of original graph structures, they comprehensively retain global graph attributes. Based on these considerations, GraphCL and SimGRACE were selected as our pre-training methods.

During pre-training, significant mean and variance discrepancies in node features across different GNN layers lead to gradual dilution of shallow semantic information in deeper layers. Traditional methods exacerbate feature-distribution oscillation in few-shot scenarios due to biased mini-batch statistical estimations. To address these issues, ContraNorm [31] is introduced, which is a systematic solution. Conventional normalization techniques solely focus on single-layer feature distributions, whereas our dual contrastive-normalization method incorporates dual optimization objectives: intra-layer feature smoothness and cross-layer distribution consistency. By synchronously implementing feature-space compactness and inter-layer distribution alignment after each GNN layer, expressed as:

$$H_t = \text{LayerNorm}\left(H_b - \frac{s}{\tau} \times \text{softmax}\left(H_b H_b^{\top}\right) H_b\right), \tag{16}$$

Where H_b and H_t represent the feature matrices before and after the update, respectively, s denotes the step size of gradient descent and τ is the temperature.

To ensure the integrity of graph structural information, this framework introduces an auxiliary feature-reconstruction loss. The loss is measured through Mean Squared Error (MSE), which quantifies the preservation of node-feature information by computing the MSE between linearly transformed raw node-feature vectors and reconstructed feature vectors. Specifically, the framework employs MLP to decode low-dimensional node embeddings, generating reconstructed features aligned with the original feature space. This mechanism aims to achieve dual objectives: at the single-graph level, it preserves crucial node characteristics during dimensionality reduction; at the multi-graph alignment level, it enhances compatibility among different graph-embedding spaces through feature-fidelity constraints, thereby mitigating information redundancy caused by feature-distribution discrepancies in cross-graph tasks. Taking GraphCL as an example, the pre-training objective is formulated as:

$$\mathcal{L} = -\log \frac{\exp\left(\sin\left(h\left(\text{PS}\left(\tilde{X}, \tilde{A}, a_i\right)\right), h\left(\text{PS}\left(\tilde{X}, \tilde{A}, a_j\right)\right)/\tau\right)\right)}{\sum \exp\left(\sin\left(h\left(\text{PS}\left(\tilde{X}, \tilde{A}, a_i\right)\right), h\left(\text{NS}\left(\tilde{X}, \tilde{A}, a_j\right)\right)/\tau\right)\right)} + \|\tilde{X} - \hat{X}\|_2, \tag{17}$$

where $\tilde{X}$ denotes the feature matrix formed by concatenating all pre-training datasets, $\tilde{A}$ represents the adjacency matrix connected *via* the coordinator, PS and NS correspond to Positive Sampling and Negative Sampling, respectively, sim indicates the similarity measurement, a_i and a_j are two distinct graph-augmentation methods, λ serves as the reconstruction loss coefficient governing the emphasis on the reconstruction task.

3.5 Applying Knowledge to Downstream Data

Our pre-training method GEA-CoPe demonstrates compatibility with diverse techniques through its task agnostic nature and task-space adaptability. During the transfer phase, the node classification is selected as the downstream task, where conventional approaches typically employ MLP as classification heads. This paper proposes replacing MLP with KAN, the core advantage of which lies in dynamically capturing complex non-linear relationships between node features through a kernel attention mechanism. KAN [32] explicitly models node similarity through this mechanism, proving particularly

effective for heterophilic graphs. The incorporation of sparse attention mechanisms reduces computational overhead while maintaining suitability for large-scale graph data. During cross-domain transfer, attention weights adaptively adjust feature importance to mitigate inter-domain distribution discrepancies.

Building upon recent advancements in graph neural networks [33], a graph-level framework is constructed for downstream tasks. Since knowledge-transfer efficiency improves when pre-training tasks and downstream applications maintain topological-space alignment, both stages employ graph-level representations. Specifically, adjacency-matrix reconstruction techniques are implemented to lift node-level tasks to the graph space, as detailed in Algorithm 1.

Algorithm 1: GEA-CoPe

Input: Source graphs $\{\mathcal{G}^{(i)}\}_{i=1}^{M}$, target graph $\mathcal{G}^{(t)}$, GNN parameters Θ, projection operation $\mathrm{Proj}(\cdot)$, pre-training objective $\mathcal{L}(\cdot)$, learning rate α, transferring pipeline $\mathrm{Trans}(\cdot)$

Output: The optimal model on the target graph $g_t(\cdot)$

for $i \leftarrow 0$ **to** M **do**

  $\tilde{X}^{(i)} = \mathrm{Proj}\left(X^{(i)}\right)$

end for

$\tilde{X} = \mathrm{Cat}\left(\tilde{X}^{(1)}, \tilde{X}^{(2)}, \ldots, \tilde{X}^{(M)}\right)$

$\tilde{A} = \begin{bmatrix} A_{\mathrm{diag}} & R_A^T \\ R_A & R_R \end{bmatrix}$

while not converge **do**

  $\Theta \leftarrow \Theta - \alpha \nabla_{\Theta} \mathcal{L}\left(\tilde{X}, \tilde{A}, \Theta\right)$

end while

$g_t(\cdot) = \mathrm{Trans}\left(\mathcal{G}^{(t)}, \Theta\right)$

return $g_t(\cdot)$

3.6 Complexity Analysis

The feature complexity of the coordinator is $O(Md_{\mathrm{A}})$, exhibiting a linear relationship with the number of pre-training datasets. In practical scenarios, situations with a large number of pre-training datasets are rare. Assuming that the employed GNN comprises L layers with a maximum layer width of d and letting $N = \sum_{k=1}^{M} \left|\mathcal{V}^{(k)}\right|$ and $E = \sum_{k=1}^{M} \left|\mathcal{E}^{(k)}\right|$, the computational cost of GEANet scales linearly with the number of nodes and edges, with a complexity of $O(N + E)$. It is noteworthy that the time complexity of a typical graph model (e.g. Graph Convolutional Network, GCN) is $O(LNd^2 + LEd + Nd)$. After incorporating the coordinator, the time complexity becomes $(L(N + M)d^2 + L(E + N + M)d + (N + M)d + (N + E))$, with an additional time complexity of $O(LMd^2 + L(N + M)d + Md + (N + E))$. When $M \ll N$, the $O(N + E)$ term from GEANet is incorporated into the linear terms of the coordinator. The dominant term remains $O(LNd^2)$ from the GNN layers and the supplementary time cost exhibits an approximately linear relationship with the original number of nodes.

4. EXPERIMENTS

In this section, experiments are conducted on various graph datasets to evaluate the methods proposed in this paper and the baseline methods and analyze the experimental results. All experiments were conducted on a server equipped with a single NVIDIA GeForce RTX 3080 GPU (10 GB memory), an Intel Xeon Platinum 8352 V CPU (12 cores @ 2.10 GHz) and 48 GB of RAM. The software environment consisted of the Ubuntu 22.04 operating system, PyTorch 2.1.2 deep-learning framework, Python 3.10 programming language and CUDA 11.8 parallel-computing platform. During the training phase, a batch size of 100 was used, with training proceeding for 100 epochs and a total training time of approximately 1.5 hours.

4.1 Experimental Setup

4.1.1 Dataset

To evaluate the accuracy of the assessment, experiments were conducted on ten real-world benchmark

datasets. These datasets include five homophilic datasets: Cora [34], Citeseer [34], Pubmed [35], Computers and Photos [36]-[37], as well as five heterophilic datasets: three sub-datasets from WebKB [38] (Cornell, Texas and Wisconsin) and two page networks extracted from Wikipedia [38] (Chameleon and Squirrel). Detailed information is presented in Table 1, where the values from [39] are used to measure the degrees of homophily and heterophily. As shown in the table, the first five datasets exhibit strong homophily, while the latter five demonstrate significant heterophily [39]-[40]. The varying degrees of homophily and heterophily reflect distinct semantic representations in graph structures.

Table 1. Statistics of datasets.

Homophilic Data	Cora	Citeseer	Pubmed	Computers	Photos
#Nodes	2,708	3,327	19,717	13,752	7,650
#Edges	10,556	9,104	88,648	491,722	238,162
#Features	1,433	3,703	500	767	745
#Labels	7	6	3	10	8
$h(G)$	0.810	0.736	0.802	0.777	0.827
Heterophilic Data	Wisconsin	Texas	Cornell	Chameleon	Squirrel
#Nodes	251	183	183	2,277	5,201
#Edges	515	325	298	62,792	396,846
#Features	1,703	1,703	1,703	2,325	2,089
#Labels	5	5	5	5	5
$h(G)$	0.196	0.108	0.305	0.231	0.222

4.1.2 Baselines

To evaluate the performance of GEA-CoPe, the framework is compared with the following baselines, which are broadly categorized into three groups and briefly summarized.

Supervised Methods: These approaches typically train GNN models on downstream tasks for direct inference. In this study, two widely-used GNN architectures are implemented: GCN [41] and FAGCN [42]. These models are selected as the backbone of our proposed GEA-CoPe method, because FAGCN is specifically tailored for both homophilic and heterophilic graphs [39], while GCN serves as a widely-used foundational GNN model that underpins FAGCN.

Isolated Pre-training with Fine-tuning: These methods leverage multiple cross-domain datasets as source datasets, which are combined in an isolated manner to pre-train GNN models in a self-supervised fashion (e.g. GraphCL [16] and SimGRACE [17]). Here, "isolated" indicates that the datasets are merged into a single batch object, resulting in an adjacency matrix composed of distinct blocks. Subsequently, the pre-trained model is fine-tuned for new downstream tasks.

Graph Coordinator for Pre-training (GCOPE) [45]: This methodology integrates disconnected source datasets into a unified large-scale graph through a coordination mechanism that establishes cross-dataset dependencies during pre-training. The resulting model is then transferred to downstream applications.

External Attention-Augmented Graph Coordinator for Pre-training (GEA-CoPe): Our proposed method employs an external attention-augmented learnable coordinator to act as a bridge for information interaction across diverse graph datasets. The pre-trained GNN model is then transferred to downstream tasks through fine-tuning or prompting, alleviating the negative-transfer problem [15].

4.1.3 Metrics and Implementations

Three universally adopted metrics were selected for evaluating node-classification tasks [39], [43]-[44]: classification accuracy (Acc), mean AUC-ROC value (AUC) and mean F1-score (F1). A 10-fold partition strategy was applied to divide ten real-world benchmark datasets, with nine serving as cross-domain source datasets for model pre-training and the remaining one designated as the target domain for transfer learning. To harmonize feature-distribution discrepancies across multiple cross-domain sources, SVD was employed for dimensionality reduction, compressing original features to 100 dimensions. Subsequently, an independent coordination module is assigned to each source dataset, with the default reconstruction-weight coefficient set to 0.2. For the external-attention module, the number of attention heads is set to 4.

In the pre-training phase, a contrastive learning framework is adopted. The number of graph neural

network layers is set to 8 and the hidden dimension is set to 128. Standard dropout regularization is applied to prevent overfitting; a dropout rate of 0.2 is used to enhance model robustness to some extent while avoiding excessive loss of information flow [46]. All networks are optimized with the Adam optimizer, with a base learning rate uniformly set to 0.0001 to ensure stable learning of general representations. Weight decay is set to 0.00001, which prevents overfitting without unduly weakening the model's expressive power.

In the transfer-learning phase, node classification serves as the primary downstream task and the training sets are constructed following the C-way-K-shot few-shot learning paradigm described in reference [48]. The remaining data is randomly split into validation and test sets in a 1: 9 ratio. The split_ratio is set to 0.1, indicating that 10% of all nodes are randomly selected as starting nodes for random walks, with each random walk length set to 30. A split ratio of 0.1 better simulates the scarcity of labeled data in real-world scenarios, thereby more effectively evaluating the model's generalization ability.

Table 2. Hyper-parameter settings.

Hyper-parameter	Value
Node Feature Dimension	100
Reconstruction Loss Weight	0.2
Number of Attention Heads	4
Number of Convolutional Layers	8
Hidden Dimension	128
Dropout Rate	0.2
Optimizer(Learning Rate)	Adam(1e-4)
Optimizer(Weight Decay)	Adam(1e-5)
Random Walk Split Ratio	0.1
Random Walk Length	30

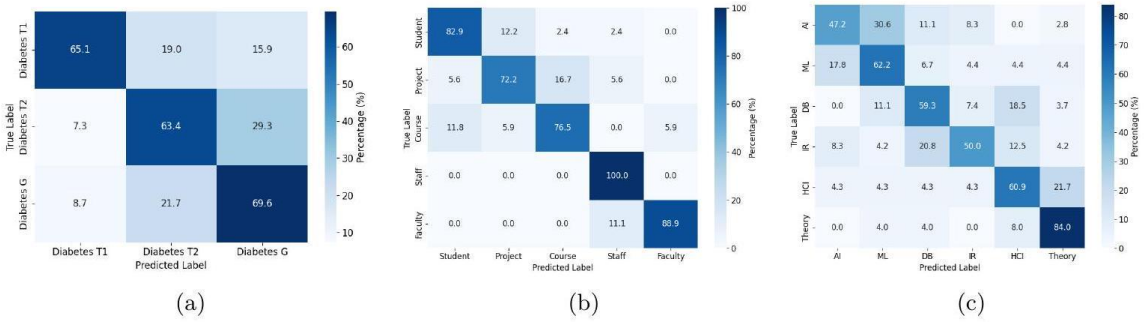
The hyper-parameter settings are listed in Table 2. To ensure robust performance across datasets and avoid performance degradation, the pre-training phase prioritizes tuning the learning rate and batch size to guarantee stable convergence, then gradually introduces reconstruction loss weight and dynamic edge pruning thresholds to enhance generalization. During fine-tuning, the learning rate is adjusted dynamically according to the sample size of the downstream task; in few-shot scenarios, the batch size is reduced and the number of training epochs is increased. The number of neural-network layers is adjusted based on the graph diameter and signs of overfitting are monitored to regulate the dropout rate. The Adam optimizer is employed throughout the experiments. Only 1-2 hyper-parameters are adjusted at a time, with evaluation *via* cross-validation. When transferring across datasets, adaptive adjustments are made according to differences in graph scale and feature distribution between the source and target domains.

4.2 Few-shot Performance Evaluation

The GEA-CoPe was compared with three baseline groups on node-classification tasks under the C-way-1shot setting. Results on homophilic graph datasets are presented in Table 3, while those on heterophilic graphs are shown in Table 4. By analyzing the performance of supervised-learning methods, the effectiveness of pre-training GNN transfer is verified and the necessity of knowledge transfer is demonstrated. Undoubtedly, the core objective of pre-training lies in learning universal features or knowledge from large-scale data to provide foundational models for downstream tasks, thereby enhancing model performance, efficiency and generalization capabilities, particularly under few-shot conditions.

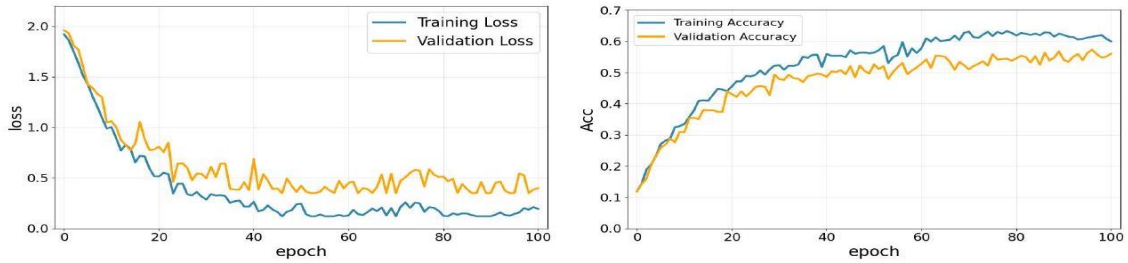
Based on our findings, the performance of supervised methods is notably inferior, with negative transfer being particularly prominent. The primary issue stems from the substantial divergence in data structures and distributions across datasets from different domains. During pre-training, samples contain information from only a single dataset and remain isolated; consequently, they fail to integrate comprehensive graph information. This consequently leads to compromised effectiveness in GNNs' learning of graph representations. It is observed that IP with fine-tuning often fails to achieve performance comparable to supervised methods, manifesting as the negative-transfer phenomenon. This is attributed to significant distribution shifts across different source domains. Under the IP strategy, each graph sample originates from one of nine distinct data distributions. As a result, graph neural networks

struggle to reconcile these disparate distributions into a unified representation space, thereby limiting their ability to learn generalizable graph features. Although the GCOPE method with graph coordinator connects cross-domain graphs into a unified framework, enabling better representation learning across graphs, its lack of effective feature-enhancement modules for node and edge attributes constrains model expressiveness, resulting in unstable feature distributions and weak generalization. In contrast, our proposed GEA-CoPe method significantly outperforms these baselines. The incorporated multi-head self-attention mechanism enhances data representation by enabling simultaneous focus on diverse feature sub-spaces, distributing attention focus and mitigating single-attention bias. Through attention-driven feature enhancement and structured computational optimization, our method improves both accuracy and efficiency, upgrading the coordinator from a basic parameter-matching framework to an efficient universal processor suitable for complex graph-structured data. Therefore, during pre-training, our approach enables more effective integration of multi-dataset information and enhances graph representations for downstream applications.



(a) (b) (c)

Figure 4. Node-classification confusion matrix of GEA-CoPe (c-way-1-shot). (a)Confusion matrix of node classification on Cora. (b)Confusion matrix of node classification on Texas. (c)Confusion matrix of node classification on Citeser.



(a) (b)

Figure 5. Node-classification accuracy and loss of GEA-CoPe on PubMed (c-way-1-shot). (a)Accuracy curve. (b)Loss curve.

Additionally, to more intuitively demonstrate the framework's performance, partial confusion matrices are plotted. As shown in Figure 4, which displays the classification results on the Cora, Texas and Citeseer datasets from left to right, the distribution within the confusion matrices reveals that the framework demonstrates significant advantages in multi-class classification tasks, particularly exhibiting strong robustness when handling complex feature interactions and ambiguous class boundaries. In the diabetes-type classification task, the framework's high accuracy for Gestational Diabetes (69.6%) reflects its strong ability to identify categories with distinct feature differences. In the user-role classification task, the perfect identification of the Staff category (100%) indicates the framework's effectiveness in capturing the unique patterns of minority or distinctively featured classes, showcasing its adaptability to extremely distributed data. In the academic-domain classification task, the high accuracy for the Theory category (84.0%) confirms the framework's capability for hierarchical modeling of classes with clear semantic features. Overall, through multi-dimensional feature-decoupling and contextual-relationship modeling, the framework efficiently identifies well-separated categories while clearly exposing bottlenecks related to feature overlap and label ambiguity. The accuracy and loss variations of the framework in node classification on PubMed are shown in Figure 5.

Furthermore, to evaluate the framework's competitiveness in cross-domain graph learning, four state-of

Table 3. Transfer learning performance (mean±std Acc/AUC/F1) on homophilic datasets (C-way-1-shot). GCL and Sim respectively represent GraphCL and SimGRACE.

Training schemes	Methods	Cora			Citeseer			Pubmed			Computers			Photos		
		Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
Supervised	GCN	$0.3027_{\pm.06}$	$0.6436_{\pm.06}$	$0.2783_{\pm.07}$	$0.3760_{\pm.04}$	$0.7230_{\pm.03}$	$0.3280_{\pm.04}$	$0.3959_{\pm.01}$	$0.5443_{\pm.02}$	$0.3575_{\pm.08}$	$0.2537_{\pm.07}$	$0.6602_{\pm.01}$	$0.2289_{\pm.04}$	$0.4092_{\pm.04}$	$0.7817_{\pm.04}$	$0.3849_{\pm.07}$
	FAGCN	$0.3359_{\pm.02}$	$0.6401_{\pm.10}$	$0.2839_{\pm.10}$	$0.5351_{\pm.02}$	$0.8335_{\pm.01}$	$0.4867_{\pm.02}$	$0.4730_{\pm.03}$	$0.5638_{\pm.04}$	$0.3828_{\pm.08}$	$0.4084_{\pm.06}$	$0.7194_{\pm.05}$	$0.2731_{\pm.06}$	$0.5335_{\pm.01}$	$0.8231_{\pm.02}$	$0.4489_{\pm.01}$
IP	GCL +GCN	$0.2507_{\pm.06}$	$0.6320_{\pm.03}$	$0.2230_{\pm.03}$	$0.3157_{\pm.02}$	$0.6631_{\pm.04}$	$0.2597_{\pm.02}$	$0.4282_{\pm.02}$	$0.5297_{\pm.05}$	$0.2994_{\pm.07}$	$0.2356_{\pm.04}$	$0.6347_{\pm.03}$	$0.1693_{\pm.06}$	$0.4093_{\pm.01}$	$0.7767_{\pm.01}$	$0.3754_{\pm.01}$
	GCL +FAGCN	$0.3749_{\pm.05}$	$0.7224_{\pm.03}$	$0.3616_{\pm.05}$	$0.4472_{\pm.02}$	$0.7682_{\pm.01}$	$0.4493_{\pm.02}$	$0.4517_{\pm.02}$	$0.5725_{\pm.03}$	$0.4137_{\pm.04}$	$0.4071_{\pm.06}$	$0.7116_{\pm.01}$	$0.2694_{\pm.03}$	$0.5407_{\pm.01}$	$0.8472_{\pm.01}$	$0.5138_{\pm.03}$
	Sim +GCN	$0.2492_{\pm.02}$	$0.5779_{\pm.03}$	$0.1597_{\pm.04}$	$0.2980_{\pm.06}$	$0.6273_{\pm.06}$	$0.2074_{\pm.06}$	$0.3993_{\pm.01}$	$0.5082_{\pm.02}$	$0.2807_{\pm.01}$	$0.2466_{\pm.10}$	$0.6248_{\pm.01}$	$0.1603_{\pm.03}$	$0.4293_{\pm.04}$	$0.7645_{\pm.02}$	$0.3967_{\pm.02}$
	Sim +FAGCN	$0.3763_{\pm.03}$	$0.7246_{\pm.02}$	$0.3561_{\pm.01}$	$0.5161_{\pm.03}$	$0.7984_{\pm.01}$	$0.4625_{\pm.04}$	$0.4386_{\pm.01}$	$0.5547_{\pm.01}$	$0.4018_{\pm.02}$	$0.3983_{\pm.01}$	$0.7118_{\pm.02}$	$0.3020_{\pm.02}$	$0.5411_{\pm.02}$	$0.8549_{\pm.02}$	$0.4955_{\pm.01}$
GCOPE	GCL +GCN	$0.3482_{\pm.07}$	$0.6701_{\pm.05}$	$0.3051_{\pm.07}$	$0.3856_{\pm.04}$	$0.7221_{\pm.04}$	$0.3052_{\pm.06}$	$0.4805_{\pm.04}$	$0.6517_{\pm.04}$	$0.4562_{\pm.03}$	$0.2479_{\pm.01}$	$0.6567_{\pm.00}$	$0.2204_{\pm.01}$	$0.4101_{\pm.03}$	$0.7846_{\pm.01}$	$0.3887_{\pm.03}$
	GCL +FAGCN	$0.3803_{\pm.01}$	$0.7314_{\pm.01}$	$0.3900_{\pm.01}$	$0.5714_{\pm.00}$	$0.8382_{\pm.01}$	$0.5214_{\pm.02}$	$0.4755_{\pm.02}$	$0.5804_{\pm.03}$	$0.4464_{\pm.03}$	$0.4015_{\pm.01}$	$0.7278_{\pm.03}$	$0.2736_{\pm.03}$	$0.5778_{\pm.05}$	$0.8650_{\pm.02}$	$0.5156_{\pm.07}$
	Sim +GCN	$0.3465_{\pm.04}$	$0.6529_{\pm.03}$	$0.2809_{\pm.03}$	$0.3428_{\pm.02}$	$0.6809_{\pm.02}$	$0.3102_{\pm.02}$	$0.3968_{\pm.00}$	$0.5430_{\pm.01}$	$0.3595_{\pm.08}$	$0.2388_{\pm.01}$	$0.6466_{\pm.01}$	$0.2240_{\pm.02}$	$0.4592_{\pm.02}$	$0.8160_{\pm.01}$	$0.4548_{\pm.03}$
	Sim +FAGCN	$0.3867_{\pm.00}$	$0.7345_{\pm.00}$	$0.3774_{\pm.00}$	$0.5645_{\pm.01}$	$0.8457_{\pm.00}$	$0.5169_{\pm.01}$	$0.4654_{\pm.02}$	$0.5676_{\pm.02}$	$0.3913_{\pm.04}$	$0.4079_{\pm.00}$	$0.7356_{\pm.02}$	$0.3070_{\pm.03}$	$0.5511_{\pm.01}$	$0.8642_{\pm.02}$	$0.5332_{\pm.02}$
GEA-CoPe	GCL +GCN	$0.4513_{\pm.02}$	$0.7712_{\pm.01}$	$0.4413_{\pm.01}$	$0.5129_{\pm.06}$	$0.7968_{\pm.02}$	$0.4580_{\pm.06}$	$0.6091_{\pm.04}$	$0.7818_{\pm.02}$	$0.6037_{\pm.04}$	$0.3510_{\pm.08}$	$0.6776_{\pm.01}$	$0.2932_{\pm.01}$	$0.4613_{\pm.04}$	$0.8253_{\pm.02}$	$0.4440_{\pm.03}$
	GCL +FAGCN	$0.4799_{\pm.03}$	$0.7767_{\pm.02}$	$0.4296_{\pm.01}$	$0.5878_{\pm.02}$	$0.8409_{\pm.01}$	$0.5425_{\pm.02}$	$0.4922_{\pm.02}$	$0.5952_{\pm.03}$	$0.4482_{\pm.03}$	$0.3951_{\pm.04}$	$0.6763_{\pm.04}$	$0.2705_{\pm.03}$	$0.6179_{\pm.03}$	$0.8804_{\pm.01}$	$0.5544_{\pm.03}$
	Sim +GCN	$0.4186_{\pm.05}$	$0.7482_{\pm.02}$	$0.4142_{\pm.06}$	$0.5056_{\pm.04}$	$0.7905_{\pm.02}$	$0.4559_{\pm.03}$	$0.5542_{\pm.03}$	$0.7040_{\pm.01}$	$0.5442_{\pm.04}$	$0.3550_{\pm.05}$	$0.6749_{\pm.03}$	$0.3155_{\pm.03}$	$0.4642_{\pm.03}$	$0.8377_{\pm.02}$	$0.4382_{\pm.03}$
	Sim +FAGCN	$0.4526_{\pm.03}$	$0.7717_{\pm.01}$	$0.4364_{\pm.04}$	$0.5990_{\pm.01}$	$0.8546_{\pm.00}$	$0.5605_{\pm.01}$	$0.4975_{\pm.04}$	$0.6966_{\pm.03}$	$0.4799_{\pm.03}$	$0.4427_{\pm.02}$	$0.7363_{\pm.03}$	$0.2956_{\pm.03}$	$0.6156_{\pm.04}$	$0.8727_{\pm.01}$	$0.5199_{\pm.02}$

Table 4. Transfer learning performance (mean±std Acc/AUC/F1) on heterophilic datasets (C-way-1-shot).GCL and Sim respectively represent GraphCL and SimGRACE.

Training schemes	Methods	Wisconsin			Texas			Cornell			Chameleon			Squirrel		
		Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
Supervised	GCN	$0.4878_{\pm.08}$	$0.7890_{\pm.05}$	$0.4334_{\pm.07}$	$0.6000_{\pm.06}$	$0.6699_{\pm.02}$	$0.4787_{\pm.05}$	$0.3650_{\pm.16}$	$0.5881_{\pm.09}$	$0.2821_{\pm.07}$	$0.2271_{\pm.00}$	$0.5311_{\pm.01}$	$0.1863_{\pm.03}$	$0.2180_{\pm.00}$	$0.5169_{\pm.00}$	$0.1518_{\pm.02}$
	FAGCN	$0.5303_{\pm.06}$	$0.8108_{\pm.04}$	$0.4919_{\pm.09}$	$0.6700_{\pm.04}$	$0.6173_{\pm.05}$	$0.4909_{\pm.08}$	$0.4188_{\pm.17}$	$0.6260_{\pm.08}$	$0.3579_{\pm.11}$	$0.2675_{\pm.02}$	$0.5568_{\pm.00}$	$0.1959_{\pm.01}$	$0.2165_{\pm.00}$	$0.5264_{\pm.00}$	$0.1595_{\pm.03}$
IP	GCL +GCN	$0.5273_{\pm.03}$	$0.7836_{\pm.03}$	$0.4417_{\pm.05}$	$0.6350_{\pm.01}$	$0.6593_{\pm.02}$	$0.4936_{\pm.09}$	$0.3772_{\pm.04}$	$0.6251_{\pm.02}$	$0.3035_{\pm.04}$	$0.2249_{\pm.02}$	$0.5224_{\pm.00}$	$0.1423_{\pm.04}$	$0.2117_{\pm.01}$	$0.5092_{\pm.01}$	$0.1103_{\pm.03}$
	GCL +FAGCN	$0.6049_{\pm.04}$	$0.8362_{\pm.01}$	$0.5588_{\pm.07}$	$0.7433_{\pm.03}$	$0.7038_{\pm.03}$	$0.6141_{\pm.09}$	$0.2688_{\pm.04}$	$0.6267_{\pm.04}$	$0.3642_{\pm.04}$	$0.2412_{\pm.00}$	$0.5470_{\pm.01}$	$0.1845_{\pm.01}$	$0.2143_{\pm.00}$	$0.5086_{\pm.00}$	$0.1728_{\pm.02}$
	Sim +GCN	$0.5058_{\pm.04}$	$0.7749_{\pm.05}$	$0.4610_{\pm.06}$	$0.5938_{\pm.05}$	$0.6425_{\pm.07}$	$0.4257_{\pm.14}$	$0.3638_{\pm.05}$	$0.5852_{\pm.09}$	$0.2768_{\pm.09}$	$0.2237_{\pm.01}$	$0.5293_{\pm.02}$	$0.1569_{\pm.03}$	$0.2063_{\pm.01}$	$0.5103_{\pm.02}$	$0.1550_{\pm.02}$
	Sim +FAGCN	$0.6215_{\pm.02}$	$0.8575_{\pm.00}$	$0.5830_{\pm.04}$	$0.6754_{\pm.12}$	$0.6582_{\pm.02}$	$0.4906_{\pm.04}$	$0.2725_{\pm.06}$	$0.6159_{\pm.04}$	$0.3417_{\pm.04}$	$0.2401_{\pm.01}$	$0.5303_{\pm.00}$	$0.1801_{\pm.00}$	$0.2137_{\pm.00}$	$0.5247_{\pm.00}$	$0.1715_{\pm.01}$
GCOPE	GCL +GCN	$0.5783_{\pm.06}$	$0.8230_{\pm.01}$	$0.4850_{\pm.04}$	$0.6425_{\pm.08}$	$0.6516_{\pm.07}$	$0.5061_{\pm.14}$	$0.3675_{\pm.03}$	$0.6302_{\pm.02}$	$0.2785_{\pm.08}$	$0.2266_{\pm.00}$	$0.5405_{\pm.03}$	$0.2092_{\pm.03}$	$0.2205_{\pm.01}$	$0.5256_{\pm.01}$	$0.1713_{\pm.01}$
	GCL +FAGCN	$0.6317_{\pm.04}$	$0.8417_{\pm.01}$	$0.5799_{\pm.06}$	$0.7787_{\pm.03}$	$0.7359_{\pm.01}$	$0.6202_{\pm.05}$	$0.5413_{\pm.06}$	$0.7959_{\pm.02}$	$0.4465_{\pm.01}$	$0.2597_{\pm.01}$	$0.5523_{\pm.01}$	$0.1982_{\pm.03}$	$0.2029_{\pm.00}$	$0.5098_{\pm.00}$	$0.1779_{\pm.01}$
	Sim +GCN	$0.4932_{\pm.08}$	$0.7885_{\pm.05}$	$0.4344_{\pm.07}$	$0.6025_{\pm.13}$	$0.6976_{\pm.01}$	$0.4232_{\pm.11}$	$0.3800_{\pm.02}$	$0.6142_{\pm.03}$	$0.3066_{\pm.05}$	$0.2264_{\pm.00}$	$0.5309_{\pm.01}$	$0.1855_{\pm.03}$	$0.2171_{\pm.00}$	$0.5249_{\pm.01}$	$0.1561_{\pm.03}$
	Sim +FAGCN	$0.6670_{\pm.04}$	$0.8684_{\pm.04}$	$0.6287_{\pm.07}$	$0.6800_{\pm.02}$	$0.6677_{\pm.01}$	$0.4850_{\pm.06}$	$0.4200_{\pm.17}$	$0.6265_{\pm.08}$	$0.3582_{\pm.11}$	$0.2786_{\pm.01}$	$0.5589_{\pm.02}$	$0.1997_{\pm.02}$	$0.2093_{\pm.00}$	$0.5206_{\pm.00}$	$0.1792_{\pm.00}$
GEA-CoPe	GCL +GCN	$0.6000_{\pm.05}$	$0.8210_{\pm.00}$	$0.5885_{\pm.05}$	$0.6590_{\pm.04}$	$0.6591_{\pm.02}$	$0.5788_{\pm.06}$	$0.3812_{\pm.08}$	$0.6344_{\pm.05}$	$0.2848_{\pm.04}$	$0.2371_{\pm.00}$	$0.5440_{\pm.00}$	$0.2028_{\pm.00}$	$0.2464_{\pm.00}$	$0.5474_{\pm.00}$	$0.2203_{\pm.01}$
	GCL +FAGCN	$0.7484_{\pm.01}$	$0.9058_{\pm.00}$	$0.7222_{\pm.01}$	$0.8100_{\pm.03}$	$0.7359_{\pm.01}$	$0.7375_{\pm.05}$	$0.6337_{\pm.01}$	$0.8281_{\pm.02}$	$0.4786_{\pm.01}$	$0.2794_{\pm.02}$	$0.5671_{\pm.02}$	$0.2306_{\pm.01}$	$0.2230_{\pm.00}$	$0.5253_{\pm.00}$	$0.1868_{\pm.00}$
	Sim +GCN	$0.6262_{\pm.04}$	$0.8215_{\pm.01}$	$0.5539_{\pm.04}$	$0.7225_{\pm.05}$	$0.7066_{\pm.01}$	$0.6257_{\pm.06}$	$0.4087_{\pm.08}$	$0.6688_{\pm.03}$	$0.2981_{\pm.03}$	$0.2382_{\pm.02}$	$0.5363_{\pm.02}$	$0.1801_{\pm.01}$	$0.2109_{\pm.00}$	$0.5193_{\pm.01}$	$0.1910_{\pm.00}$
	Sim +FAGCN	$0.7774_{\pm.00}$	$0.9243_{\pm.01}$	$0.7469_{\pm.01}$	$0.7475_{\pm.00}$	$0.6810_{\pm.00}$	$0.5957_{\pm.03}$	$0.5237_{\pm.06}$	$0.7996_{\pm.03}$	$0.3814_{\pm.05}$	$0.2407_{\pm.02}$	$0.5324_{\pm.01}$	$0.1993_{\pm.01}$	$0.2204_{\pm.00}$	$0.5342_{\pm.00}$	$0.2073_{\pm.01}$

-the-art methods (MDGPT [49], MDGFM [50], SAMGPT [51] and UniPrompt [52]) were selected for comparison. As shown in Table 5. Compared to MDGPT, which employs domain tokens for explicit

feature semantic alignment, our model implicitly enhances the discriminative power and domain invariance of features by introducing contrastive-learning signals during the normalization process, thereby avoiding potential semantic bias caused by explicit token alignment. While MDGFM relies on complex graph-structure learning for explicit topological reconstruction, our model utilizes the more lightweight ContraNorm to implicitly improve robustness, maintaining efficiency while avoiding the significant overhead and potential structural distortion risks associated with graph topology-aware alignment. Unlike SAMGPT, which depends on structural tokens for layer-wise topological alignment, our model achieves dynamic, attention-weighted fusion of multi-source domain contributions *via* GEANet within the coordinator, eliminating the need for introducing fixed structural parameters. In contrast to the general prompt framework UniPrompt, our model is specifically designed for cross-domain graph learning. The dynamic domain-adaptation capability provided by GEANet is significantly superior to UniPrompt's static task templates. Simultaneously, the powerful function-approximation capability of the KAN classifier head far exceeds that of commonly used linear or shallow classifiers in few-shot scenarios.

Overall, through the synergistic design of "dynamic fusion, contrastive enhancement and strong-fitting classification," our model demonstrates excellent performance across three key aspects: adaptive integration of multi-domain knowledge, robustness of representations and adaptation to downstream tasks. The best reported node-classification performance of these methods across ten datasets was compared with the best performance achieved by our proposed framework. As shown in Table 6, the best performance of our proposed framework clearly surpasses that of the other methods, demonstrating its effectiveness.

Table 5. Cross-domain graph methods.

Method	Core Architecture	Alignment Mechanism	Domain Adaptation	Classification Head
MDGPT [48]	Domain Tokens +Dual Prompts	Domain Token Explicit Alignment	Unified Prompt +Mixed Prompt	Linear Classifier or Prototypical Classifier
MDGFM [49]	Graph Structure Learning +Dual Prompts	Graph Structure Learning Explicit Alignment	Meta Prompt +Task Prompt	Prototypical Classifier or Linear Classifier
SAMGPT [50]	Structure Tokens +Dual Prompts	Structure Token Explicit Alignment	Global Prompt +Specific Prompt	Prototypical Classifier
UniPrompt [51]	Unified Task Template+Learnable Prompts	Task Template Alignment	General Prompting	Linear Classifier or Shallow MLP
GEA-CoPe	External Attention Enhanced Coordinator	Coordinator Implicit Semantic Alignment	Coordinator Adaptive Weighting Prompt /Fine-tuning	KAN: Strong Nonlinear Function Approximation

4.3 Reconstruction Loss Analysis

On the Citeseer dataset, the proposed method was systematically evaluated for its impact on downstream node classification tasks under different reconstruction loss coefficients, with a comparative analysis of supervised-learning methods used to assess the effectiveness of the reconstruction module. In the specific experimental setup, FAGCN was adopted as the backbone network architecture and GraphCL was employed as the graph contrastive-learning pre-training strategy. To ensure a fair comparison, all other hyper-parameter configurations were kept identical across the compared methods. A detailed comparison of the experimental results is shown in Figure 6, with a comprehensive analysis conducted based on three evaluation metrics: node-classification accuracy (Acc), area under the ROC curve (AUC) and F1-score.

Based on the experimental data, the following conclusions can be drawn: First, without the reconstruction module ($\lambda = 0.0$), the framework already outperforms supervised pre-training, demonstrating the effectiveness of the coordinator design. Second, when the reconstruction module is introduced and λ is set to 0.2, the model achieves optimal performance, surpassing not only supervised pre-training, but also the framework without reconstruction ($\lambda = 0.0$). This improvement benefits from the reconstruction module's ability to align graph features across datasets, enabling the graph neural network to more effectively learn common information from multi-source cross-domain data. However,

when λ exceeds 0.2, model performance begins to decline, eventually falling below both supervised pre-training and the performance without reconstruction. This is attributed to excessively large λ values causing the model to over-prioritize the reconstruction task, thereby weakening the learning effectiveness of the primary pre-training task. In summary, introducing the reconstruction module with a relatively small λ value is a key factor in ensuring the effectiveness of the framework method.

Table 6. Node-classification accuracy for cross-domain graph pre-training methods.

Methods	Cora	Citeseer	Pubmed	Computers	Photos	Wisconsin	Texas	Cornell	Chameleon	Squirrel
MDGPT [48]	$0.4226_{\pm.10}$	$0.4240_{\pm.09}$	$0.4982_{\pm.08}$	$0.4216_{\pm.11}$	$0.5496_{\pm.10}$	$0.5040_{\pm.15}$	$0.5976_{\pm.12}$	$0.5419_{\pm.13}$	$0.2804_{\pm.04}$	$0.2441_{\pm.07}$
MDGFM [49]	$0.4483_{\pm.07}$	$0.4218_{\pm.06}$	$0.4684_{\pm.07}$	-	-	-	-	$0.4077_{\pm.05}$	$0.2836_{\pm.03}$	$0.2430_{\pm.03}$
SAMGPT [50]	$0.4680_{\pm.11}$	$0.3638_{\pm.09}$	$0.5025_{\pm.10}$	$0.4522_{\pm.08}$	$0.5871_{\pm.08}$	$0.5229_{\pm.14}$	$0.6679_{\pm.10}$	$0.5934_{\pm.09}$	$0.2812_{\pm.08}$	$0.2475_{\pm.06}$
UniPrompt[51]	$0.4537_{\pm.09}$	$0.4325_{\pm.09}$	$0.5501_{\pm.03}$	-	-	-	-	$0.5158_{\pm.09}$	$0.2514_{\pm.05}$	$0.2429_{\pm.03}$
GEA-CoPe	$0.4799_{\pm.03}$	$0.5990_{\pm.01}$	$0.6091_{\pm.04}$	$0.4427_{\pm.02}$	$0.6179_{\pm.03}$	$0.7774_{\pm.00}$	$0.8100_{\pm.03}$	$0.6337_{\pm.01}$	$0.2794_{\pm.00}$	$0.2464_{\pm.00}$

* "——" denotes that the official code has not been released for implementation on these datasets.

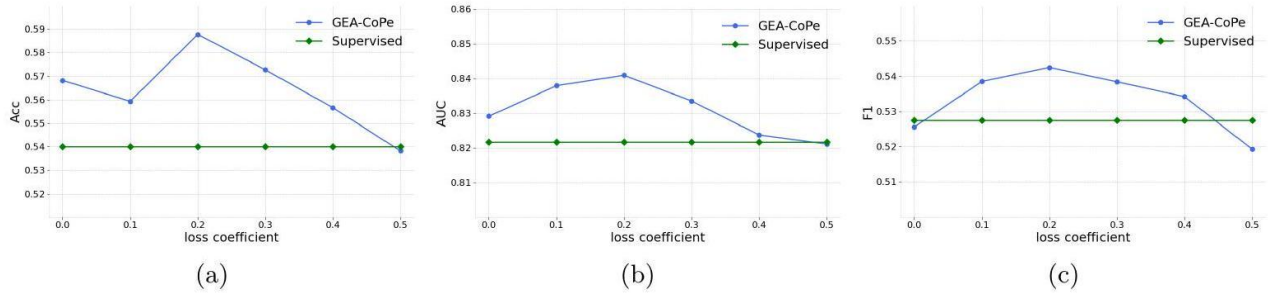


Figure 6. Node-classification performance of GEA-CoPe on Citeseer under C-way-1-shot setting. (a)Variation of Acc with reconstruction loss coefficient. (b)Variation of AUC with reconstruction loss coefficient. (c)Variation of F1-score with reconstruction loss coefficient.

4.4 Transferring by Graph Prompt

To transfer and apply knowledge learned from upstream tasks to downstream tasks, two methods are selected: fine-tuning and graph-prompting techniques. Next, the feasibility of knowledge transfer *via* graph prompting techniques is tested. More specifically, ProG [17] method is adopted, which is a revolutionary graph neural network transfer-learning paradigm. It constructs a lightweight, learnable "prompt graph" relevant to the downstream task and structurally integrates this prompt graph with the original input graph, thereby effectively "prompting" the frozen pre-trained GNN model with task information.

The downstream datasets Cora, citeseer, Wisconsin and Texas were selected for the node classification task, including two homophilic and two heterophilic datasets, to evaluate model performance. The experimental results are shown in Table 7. To rigorously and intuitively assess the viability of the ProG method, the results were compared with the results of supervised methods and the results of GEA-CoPe using fine-tuning.

By comparing the experimental results, the following conclusions can be drawn: GEA-CoPe demonstrates superior performance compared to other methods, regardless of whether knowledge is transferred using fine-tuning or the ProG method. Particularly in the node-classification task, GEA-CoPe utilizing ProG achieves positive transfer with the fewest tunable parameters. However, the model using ProG performs slightly worse than the model using fine-tuning. Models employing these two methods generally outperform supervised methods. Through analysis of the results, it can be concluded that our proposed framework is favorable for prompt learning on downstream tasks.

4.5 Impact of Attention Heads

To investigate the impact of the number of external attention heads on GEA-CoPe, the performance of GEA-CoPe method with varying numbers of attention heads in downstream node-classification task was

Table 7. Cross-domain transfer-learning performance (mean ± std Acc/AUC/F1) of GEA-CoPe with ProG (C-way-1-shot). GCL and Sim, respectively, representing GraphCL and SimGRACE.

Training schemes	Methods	Cora			Pubmed		
		Acc	AUC	F1	Acc	AUC	F1
Supervised	FAGCN	$0.3359_{\pm.02}$	$0.6401_{\pm.10}$	$0.2839_{\pm.10}$	$0.4730_{\pm.03}$	$0.5638_{\pm.04}$	$0.3828_{\pm.08}$
GEA-CoPe +ProG	GCL-FAGCN	$0.3419_{\pm.01}$	$0.7230_{\pm.02}$	$0.3041_{\pm.08}$	$0.4750_{\pm.02}$	$0.6732_{\pm.04}$	$0.4205_{\pm.01}$
	Sim-FAGCN	$0.4015_{\pm.02}$	$0.7265_{\pm.01}$	$0.3700_{\pm.03}$	$0.4450_{\pm.00}$	$0.5922_{\pm.01}$	$0.4384_{\pm.01}$
GEA-CoPe +finetuning	GCL-FAGCN	$0.4699_{\pm.03}$	$0.7767_{\pm.02}$	$0.4296_{\pm.01}$	$0.4922_{\pm.02}$	$0.5952_{\pm.03}$	$0.4482_{\pm.03}$
	Sim-FAGCN	$0.4526_{\pm.03}$	$0.7717_{\pm.01}$	$0.4364_{\pm.04}$	$0.4975_{\pm.04}$	$0.6966_{\pm.03}$	$0.4799_{\pm.03}$
Training schemes	Methods	Wisconsin			Texas		
		Acc	AUC	F1	Acc	AUC	F1
Supervised	FAGCN	$0.5303_{\pm.06}$	$0.8108_{\pm.04}$	$0.4919_{\pm.09}$	$0.6700_{\pm.04}$	$0.6173_{\pm.05}$	$0.4909_{\pm.08}$
GEA-CoPe +ProG	GCL-FAGCN	$0.5467_{\pm.00}$	$0.8216_{\pm.00}$	$0.4863_{\pm.02}$	$0.7712_{\pm.03}$	$0.6847_{\pm.00}$	$0.6412_{\pm.07}$
	Sim-FAGCN	$0.7394_{\pm.00}$	$0.8944_{\pm.01}$	$0.6982_{\pm.02}$	$0.7400_{\pm.03}$	$0.6645_{\pm.00}$	$0.6420_{\pm.07}$
GEA-CoPe +finetuning	GCL-FAGCN	$0.7484_{\pm.01}$	$0.9058_{\pm.00}$	$0.7222_{\pm.01}$	$0.8104_{\pm.03}$	$0.7359_{\pm.01}$	$0.7375_{\pm.05}$
	Sim-FAGCN	$0.7774_{\pm.00}$	$0.9243_{\pm.01}$	$0.7469_{\pm.01}$	$0.7475_{\pm.00}$	$0.6810_{\pm.00}$	$0.5957_{\pm.03}$

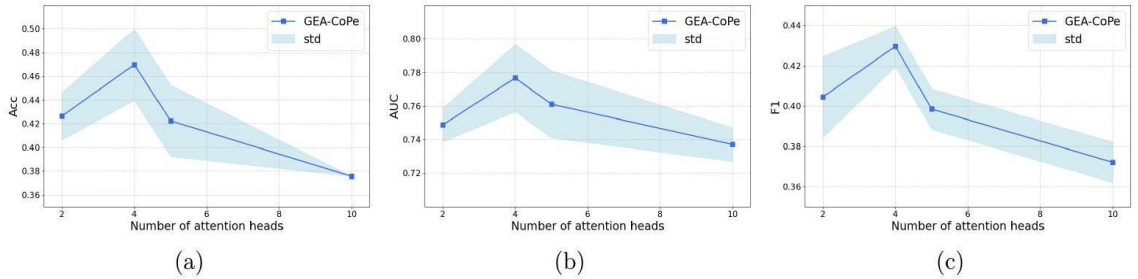


Figure 7. Node-classification performance (mean ± std) of GEA-CoPe on Cora under C-way-1shot setting. (a)Variation of Acc with the number of attention heads. (b)Variation of AUC with the number of attention heads. (c)Variation of F1-score with the number of attention heads.

compared. Specifically, FAGCN is selected as the backbone model, GraphCL is selected as the pre-training strategy, all other super-parameters are consistent and the node-classification task is performed on the Cora dataset. The experimental results, presented in Figure 7, primarily demonstrate the Acc, AUC and F1-score metrics.

From the figure, it can be observed that performance initially increases and then decreases with the growing number of attention heads: An insufficient number of attention heads leads to lower performance due to insufficient representation capacity. Increasing the number of heads enables the model to capture richer neighborhood information, significantly improving the Acc, AUC and F1-score. However, when the number of attention heads becomes excessive, performance declines as the model suffers from over-fitting or noise interference. When the external number of attention heads is 4, all metrics reach their peaks, resulting in the best node-classification performance.

4.6 Analysis of Neural Network Layers

To systematically evaluate the impact of neural-network depth on model performance, a controlled variable experiment was designed. While keeping the hidden-layer dimensionality and other hyper-parameters fixed, the number of graph neural-network layers was progressively increased. Experiments were conducted uniformly using the GraphCL and FAGCN methods on the Photos and Texas datasets to assess the influence of GNN depth on framework performance, with evaluation metrics including Accuracy, AUC and F1-score. The results are shown in Figure 8.

The experimental results clearly demonstrate that as the number of layers increases, the model performance shows an upward trend. When the number of layers is less than 8, the node-classification

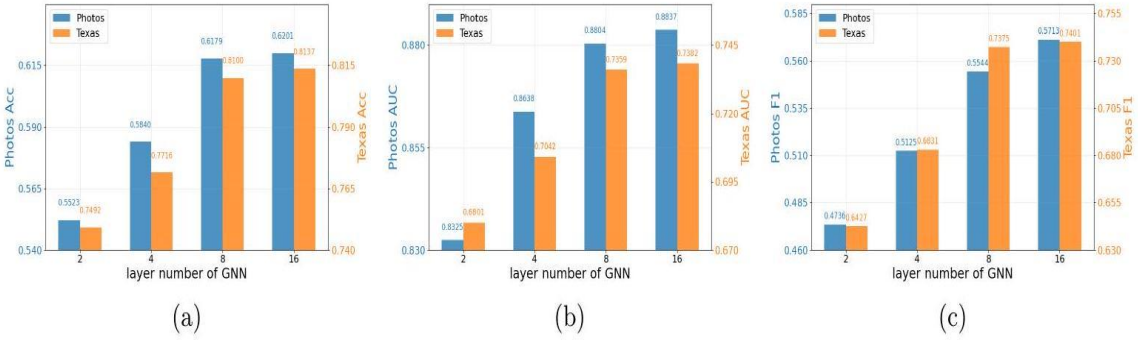


Figure 8. Node-classification performance of GEA-CoPe on Photos and Texas under C-way-1shot setting. (a) The variation of Acc with the number of layers in GNNs. (b) The variation of AUC with the number of layers in GNNs. (c) The variation of F1-score with the number of layers in GNNs.

performance increases markedly, whereas beyond 8 layers, the improvement becomes more gradual. This occurs, because excessively deep network structures are prone to issues, such as gradient vanishing or over-smoothing during propagation, which can impair the model's ability to discriminate local node features. While too few layers may lead to under-fitting, too many layers significantly increase computational time. Selecting 8 GNN layers achieves an optimal balance between node-classification accuracy and runtime.

4.7 Ablation Experiments

To thoroughly investigate the impact of individual components in GEA-CoPe on the overall model performance, multiple ablation studies were conducted, analyzing the effects of the graph external-attention mechanism, dual contrast normalization and the KAN classification head. Under the unified framework employing both SimGRACE and FAGCN, four variants were constructed: Variant 1 incorporates only graph external attention, excludes dual contrast normalization and uses a traditional classification head; Variant 2 removes the external attention from the base model; Variant 3 utilizes traditional graph neural networks for data processing; Variant 4 adopts a traditional classification head. Experiments were performed on the base model and all variants, as shown in Table 8 and Table 9.

Table 8. Node-classification performance on homophilic datasets (C-way-1-shot).

Methods	Cora			Pubmed			Photos		
	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
Base Model	$0.3867_{\pm.00}$	$0.7345_{\pm.00}$	$0.3774_{\pm.02}$	$0.4654_{\pm.02}$	$0.5676_{\pm.02}$	$0.3913_{\pm.04}$	$0.5541_{\pm.01}$	$0.8342_{\pm.02}$	$0.5012_{\pm.02}$
Variant 1	$0.4023_{\pm.02}$	$0.7419_{\pm.03}$	$0.3948_{\pm.01}$	$0.4763_{\pm.03}$	$0.6192_{\pm.01}$	$0.4358_{\pm.02}$	$0.5924_{\pm.03}$	$0.8671_{\pm.00}$	$0.5136_{\pm.01}$
Variant 2	$0.4186_{\pm.02}$	$0.7463_{\pm.02}$	$0.4292_{\pm.02}$	$0.4792_{\pm.02}$	$0.6271_{\pm.04}$	$0.4030_{\pm.06}$	$0.6030_{\pm.02}$	$0.8460_{\pm.03}$	$0.5163_{\pm.02}$
Variant 3	$0.4072_{\pm.03}$	$0.7128_{\pm.02}$	$0.4037_{\pm.03}$	$0.4629_{\pm.00}$	$0.5716_{\pm.04}$	$0.4559_{\pm.01}$	$0.5943_{\pm.02}$	$0.8654_{\pm.01}$	$0.5276_{\pm.03}$
Variant 4	$0.4012_{\pm.01}$	$0.7427_{\pm.01}$	$0.4009_{\pm.01}$	$0.4886_{\pm.01}$	$0.6401_{\pm.02}$	$0.4689_{\pm.01}$	$0.6120_{\pm.03}$	$0.8686_{\pm.01}$	$0.5047_{\pm.03}$
GEA-CoPe	$0.4526_{\pm.03}$	$0.7717_{\pm.01}$	$0.4364_{\pm.04}$	$0.4975_{\pm.04}$	$0.6966_{\pm.03}$	$0.4799_{\pm.03}$	$0.6156_{\pm.04}$	$0.8727_{\pm.01}$	$0.5199_{\pm.02}$

As evidenced by the table, the base model performs the worst across all datasets, while the variant methods exhibit certain advantages in specific scenarios, but demonstrate inconsistent performance. GEACoPe achieves particularly marked improvements on heterophilic datasets, indicating its effectiveness in handling class-distribution imbalance and complex connectivity patterns. The proposed framework outperforms both the base model and the variants in the vast majority of cases, highlighting its comprehensive superiority, especially on heterophilic datasets where it shows significant enhancements. This demonstrates the framework's strong generalization capability in effectively addressing node-classification tasks across diverse graph structures.

4.8 Robustness Analysis

To evaluate the robustness of the model, three typical types of feature perturbation-Gaussian noise injection, feature sparsification and node-feature masking - were introduced to simulate common data-

Table 9. Node-classification performance on heterophilic datasets (C -way-1-shot).

Methods	Wisconsin			Texas			Squirrel		
	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
Base Model	$0.6070_{\pm.04}$	$0.8284_{\pm.04}$	$0.5287_{\pm.07}$	$0.6800_{\pm.02}$	$0.6477_{\pm.01}$	$0.4850_{\pm.06}$	$0.2093_{\pm.00}$	$0.5106_{\pm.00}$	$0.1692_{\pm.01}$
Variant 1	$0.7153_{\pm.02}$	$0.8914_{\pm.03}$	$0.7140_{\pm.02}$	$0.7092_{\pm.05}$	$0.6526_{\pm.03}$	$0.5413_{\pm.02}$	$0.2146_{\pm.01}$	$0.5283_{\pm.01}$	$0.1892_{\pm.00}$
Variant 2	$0.7285_{\pm.01}$	$0.8987_{\pm.01}$	$0.7206_{\pm.01}$	$0.6525_{\pm.17}$	$0.6739_{\pm.05}$	$0.5308_{\pm.13}$	$0.2133_{\pm.00}$	$0.5219_{\pm.00}$	$0.1751_{\pm.01}$
Variant 3	$0.6100_{\pm.04}$	$0.8317_{\pm.00}$	$0.5398_{\pm.05}$	$0.7125_{\pm.06}$	$0.6509_{\pm.04}$	$0.4995_{\pm.02}$	$0.2174_{\pm.00}$	$0.5279_{\pm.00}$	$0.1604_{\pm.01}$
Variant 4	$0.5919_{\pm.02}$	$0.8278_{\pm.04}$	$0.4861_{\pm.10}$	$0.7300_{\pm.03}$	$0.6686_{\pm.03}$	$0.5738_{\pm.05}$	$0.2142_{\pm.00}$	$0.5265_{\pm.00}$	$0.1719_{\pm.02}$
GEA-CoPe	$0.7774_{\pm.00}$	$0.9243_{\pm.01}$	$0.7469_{\pm.01}$	$0.7475_{\pm.00}$	$0.6810_{\pm.00}$	$0.5957_{\pm.03}$	$0.2197_{\pm.00}$	$0.5302_{\pm.00}$	$0.1806_{\pm.01}$

quality issues in real-world applications, such as noise, sparse node features or partially missing attributes. The experiments were conducted on both homophilic and heterophilic datasets as target domains under a 1-shot learning setting. Methods including GraphCL and FAGCN were employed, with pre-training performed on the remaining nine datasets and downstream tasks carried out on the target dataset. Perturbations of the same type and intensity were applied in both stages to comprehensively assess the model's robustness under impaired feature conditions.

Table 10. Node-classification performance with Gaussian noise on GEA-CoPe (C-way-1-shot).

Standard deviation	Cora			Texas		
	Acc	AUC	F1	Acc	AUC	F1
0.0	$0.4799_{\pm.03}$	$0.7767_{\pm.02}$	$0.4296_{\pm.01}$	$0.8100_{\pm.03}$	$0.7359_{\pm.01}$	$0.7375_{\pm.05}$
0.3	$0.4648_{\pm.01}$	$0.7628_{\pm.02}$	$0.4055_{\pm.01}$	$0.7875_{\pm.05}$	$0.7047_{\pm.03}$	$0.6563_{\pm.09}$
0.5	$0.4512_{\pm.03}$	$0.7514_{\pm.03}$	$0.3921_{\pm.03}$	$0.7650_{\pm.06}$	$0.6925_{\pm.04}$	$0.6314_{\pm.11}$
0.7	$0.4326_{\pm.04}$	$0.7398_{\pm.04}$	$0.3787_{\pm.04}$	$0.7412_{\pm.07}$	$0.6783_{\pm.05}$	$0.6059_{\pm.12}$

The experimental results are shown in Tables 10, 11 and 12. Overall, the model demonstrates notable robustness and superiority when facing various feature perturbations. Under Gaussian-noise interference, the model achieves cross-graph feature smoothing through its coordinator. Even under high-intensity noise, it maintains high accuracy, indicating its strong filtering capability against random errors. In the feature-sparsification experiments, when 90% of features are zeroed out, the model exhibits only a slight drop in accuracy, benefiting from the cross-graph information compensation and structural enhancement enabled by the coordinator and external-attention mechanism. Particularly in heterophilic graphs, the rich topological structure provides critical information compensation, resulting in significantly better performance retention compared to homophilic graphs. In the most challenging scenario of node-feature masking, where 70% of node features are completely absent, the accuracy on the Texas dataset remains at 72.64%. This suggests that the model does not simply rely on complete feature inputs, but can effectively capture key patterns from partially masked features and integrate graph structural information for reliable inference. Comprehensive analysis indicates that the model's robustness stems from its dynamic adaptive mechanism: the external attention and coordinator can intelligently adjust the weights of intra- and inter-graph information flow according to the type and intensity of interference, achieving synergy among feature smoothing, missing feature compensation and structural enhancement. Moreover, multi-graph pre-training endows the model with more generalized feature invariance. This enables the model not only to perform well under ideal data conditions, but also to maintain stable performance with low-quality and incomplete feature data commonly encountered in real-world scenarios.

Table 11. Node-classification performance with feature sparsification on GEA-CoPe (C-way-1shot).

Sparse scale	Cora			Texas		
	Acc	AUC	F1	Acc	AUC	F1
0%	$0.4799_{\pm.03}$	$0.7767_{\pm.02}$	$0.4296_{\pm.01}$	$0.8100_{\pm.03}$	$0.7359_{\pm.01}$	$0.7375_{\pm.05}$
50%	$0.4646_{\pm.02}$	$0.7678_{\pm.01}$	$0.4181_{\pm.03}$	$0.7737_{\pm.04}$	$0.7252_{\pm.02}$	$0.6485_{\pm.09}$
70%	$0.4482_{\pm.03}$	$0.7543_{\pm.03}$	$0.4027_{\pm.04}$	$0.7419_{\pm.05}$	$0.7128_{\pm.03}$	$0.6184_{\pm.10}$
90%	$0.4185_{\pm.04}$	$0.7326_{\pm.04}$	$0.3789_{\pm.05}$	$0.7013_{\pm.07}$	$0.6934_{\pm.04}$	$0.5742_{\pm.12}$

Table 12. Node classification performance with node feature masking on GEA-CoPe (C-way-1shot).

Mask scale	Cora			Texas		
	Acc	AUC	F1	Acc	AUC	F1
0%	$0.4799_{\pm.03}$	$0.7767_{\pm.02}$	$0.4296_{\pm.01}$	$0.8100_{\pm.03}$	$0.7359_{\pm.01}$	$0.7375_{\pm.05}$
30%	$0.4324_{\pm.02}$	$0.7716_{\pm.03}$	$0.4094_{\pm.04}$	$0.7925_{\pm.04}$	$0.7263_{\pm.01}$	$0.7278_{\pm.07}$
50%	$0.4018_{\pm.03}$	$0.7582_{\pm.04}$	$0.3876_{\pm.05}$	$0.7637_{\pm.05}$	$0.7129_{\pm.02}$	$0.6843_{\pm.09}$
70%	$0.3685_{\pm.04}$	$0.7427_{\pm.05}$	$0.3621_{\pm.06}$	$0.7264_{\pm.06}$	$0.6985_{\pm.03}$	$0.6328_{\pm.10}$

5. CONCLUSION

This study addresses the negative-transfer problem in cross-domain graph pre-training under few-shot learning scenarios by proposing a novel multi-component framework named GEA-CoPe. The inherent structural and semantic discrepancies between graph domains significantly hinder effective knowledge transfer, while existing methods often fail to resolve this issue due to their limited adaptability and lack of explicit constraints on feature consistency. The proposed framework innovatively integrates multi-head external attention with a graph coordinator, enabling dynamic and adaptive cross-graph semantic alignment to bridge domain gaps while preserving unique structural information. The introduced dual feature-normalization strategy, which combines intra-layer node-similarity constraints with a cross-layer distribution-alignment loss, effectively mitigates feature drift and enhances the robustness and stability of pre-trained representations. Furthermore, by incorporating Kolmogorov-Arnold Networks (KAN) with parameter-adaptive activation functions, the model gains superior non-linear representation capability and improved interpretability, allowing it to better capture complex topological dependencies. Extensive experiments conducted on ten real-world graph datasets demonstrate that GEA-CoPe significantly outperforms existing methods in both cross-domain generalization and few-shot node-classification tasks. The model's ability to focus on critical graph structures while maintaining consistent feature distributions throughout propagation highlights its practical potential in complex and resource-constrained environments.

Despite the encouraging results, the proposed framework has certain limitations. Its performance still partially depends on the quality and diversity of the pre-training data. Moreover, the increased model complexity may require additional computational resources during training. Future work will focus on extending the framework to handle more dynamic and heterogeneous graph structures, optimizing its efficiency for large-scale deployment and exploring its integration with other advanced pre-training paradigms.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 52174184).

REFERENCES

[1] D. Bhattacharjee et al., "Vision Transformer Adapters for Generalizable Multitask Learning," Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV), pp. 19015-19026, Paris, France, 2023.

[2] M. Sun et al., "GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks," Proc. of the 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD), pp. 1717-1727, DOI: 10.1145/3534678.3539249 2022.

[3] J. Liu et al., "Graph Foundation Models: Concepts, Opportunities and Challenges," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. 47, no. 6, pp. 5023-5044, 2025.

[4] X. Sun et al., "All in One: Multi-task Prompting for Graph Neural Networks," Proc. of the 29th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD), pp. 2120–2131, DOI: 10.1145/3580305.3599256, 2023.

[5] A. A. Khan et al., "Blockchain-enabled Secure Internet of Medical Things (IoMT) Architecture for Multi-Modal Data Fusion in Precision Cancer Diagnosis and Continuous Monitoring," Journal of Cloud Computing, vol. 14, p. 58, 2025.

[6] B.-S. Shi et al., "Domain Adaptation for Graph Representation Learning: Challenges, Progress and Prospects," Journal of Computer Science and Technology, vol. 40, pp. 283–300, 2025.

[7] Y. Xue et al., "A Review on Transferability Estimation in Deep Transfer Learning," IEEE Transactions on Artificial Intelligence (IEEE TAIS), vol. 5, no.12, pp. 5894 - 5914, 2024.

[8] A. A. Laghari et al., "A Novel and Secure Artificial Intelligence Enabled Zero Trust Intrusion Detection

in Industrial Internet of Things Architecture," Scientific Reports, vol. 15, p. 26843, 2025.

[9] X. Wu et al., "ProCom: A Few-shot Targeted Community Detection Algorithm," Proc. of the 30th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD), pp. 3414–3424, DOI: 10.1145/3637528.3671749, 2024.

[10] L. Sun et al., "RiemannGFM: Learning a Graph Foundation Model from Riemannian Geometry," Proc. of the ACM on Web Conf. (WWW), pp. 1154–1165, DOI: 10.1145/3696410.3714952, 2025.

[11] Z. Wang et al., "Negative as Positive: Enhancing Out-of-distribution Generalization for Graph Contrastive Learning," Proc. of the 47th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR), pp. 2548–2552, DOI: 10.1145/3626772.3657927 2024.

[12] Q. Chen et al., "DAGPrompt: Pushing the Limits of Graph Prompting with a Distribution-aware Graph Prompt Tuning Approach," Proc. of the ACM on Web Conf. (WWW), pp. 4346–4358, DOI: 10.1145/3696410.3714917, 2025.

[13] X. Huang et al., "Enhancing Cross-domain Link Prediction *via* Evolution Process Modeling," Proc. of the ACM on Web Conf. (WWW), pp. 2158–2171, DOI: 10.1145/3696410.3714792, 2025.

[14] L. Kong et al., "Gofa: A Generative One-for-all Model for Joint Graph Language Modeling," Proc. of the 13th Int. Conf. on Learning Representations (ICLR), DOI:10.1021/acsengineeringau.3c00058.s001, 2025.

[15] M. Zhang et al., "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks," Proc. of the ACM Web Conf. (WWW), pp. 1003–1014, DOI: 10.1145/3589334.3645682, 2024.

[16] Y. You et al., "Graph Contrastive Learning with Augmentations," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 5812–5823, 2020.

[17] J. Xia et al., "SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation," Proc. of the ACM Web Conf. (WWW), pp. 1070–1079, DOI: 10.1145/3485447.3512156, 2022.

[18] Z. Hu et al., "GPT-GNN: Generative Pre-training of Graph Neural Networks," Proc. of the 26th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 1857–1867, DOI: 10.1145/3394486.3403237, 2020.

[19] Z. Hou et al., "GraphMAE: Self-supervised Masked Graph Auto-encoders," Proc. of the 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD), pp. 594–604, DOI: 10.1145/3534678.3539321, 2022.

[20] H. Yan et al., "Hierarchical Graph Contrastive Learning," Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), vol. 14170, pp. 700–715, 2023.

[21] Q. Dai et al., "Graph Transfer Learning *via* Adversarial Domain Adaptation with Graph Convolution," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 5, pp. 4908–4922, 2022.

[22] T. Vayer et al., "Fused Gromov-Wasserstein Distance for Structured Objects," Algorithms, vol. 13, no. 9, p. 212, 2020.

[23] Z. Hu, Y. Dong, K. Wang and Y. Sun, "Heterogeneous Graph Transformer," Proc. of the Web Conf. (WWW), pp. 2704–2710, DOI: 10.1145/3366423.3380027, 2020.

[24] G. Wan et al., "Reinforcement Learning-based Meta-path Discovery in Large-scale Heterogeneous Information Networks," Proc. of the AAAI Conf. on Artifi. Intell., vol. 34, no. 04, pp. 6094–6101, 2020.

[25] A. Pareja et al., "EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs," Proc. of the AAAI Conf. on Artificial Intelligence, vol. 34, no. 04, pp. 5363–5370, 2020.

[26] R. Trivedi et al., "DyRep: Learning Representations over Dynamic Graphs," Proc. of Int. Conf. on Learning Representations (ICLR), DOI:10.32920/26883523.v1, 2019.

[27] G. Song, Y. Zhang, L. Xu and H. Lu, "Domain Adaptive Network Embedding," IEEE Transactions on Big Data, vol. 8, no. 5, pp. 1220–1232, 2020.

[28] L. Chen et al., "Graph Optimal Transport for Cross-Domain Alignment," Proc. of the 37th Int. Conf. on Machine Learning (ICML), vol.119, pp. 1542–1553, 2020.

[29] J. Liang, M. Chen and J. Liang, "Graph External Attention Enhanced Transformer," Proc. of Int. Conf. on Machine Learning (ICML), Vol. 235 pp. 29560–29574, 2024.

[30] W. Jin et al., "Self-supervised Learning on Graphs: Deep Insights and New Direction," [Online], Available: https://doi.org/10.48550/arXiv.2006.10141, 2020.

[31] X. Guo et al., "ContraNorm: A Contrastive Learning Perspective on Over-smoothing and Beyond," Proc. of the 11th Int. Conf. on Learning Representations (ICLR), DOI: 10.48550/arXiv.2303.06562, 2023.

[32] Z. Liu et al., "KAN: Kolmogorov-Arnold Networks," Proc. of the 13th Int. Conf. on Learning Representations (ICLR), DOI: 10.31224/5413, 2025.

[33] R. Rossi and N. Ahmed, "The Network Data Repository with Interactive Graph Analytics and Visualization," Proc. of the 29th AAAI Conf. on Artificial Intelligence, vol. 29, no. 1, DOI: 10.1609/aaai.v29i1.9277, 2015.

[34] P. Sen et al., "Collective Classification in Network Data," AI Magazine, vol. 29, no. 3, p. 93, 2008.

[35] G. Namata et al., "Query-driven Active Surveying for Collective Classification," Proc. of the 10th Int. Workshop on Mining and Learning with Graphs (MLG), vol. 8, pp. 1-8, Edinburgh, UK, 2012.

[36] J. McAuley et al., "Image-based Recommendations on Styles and Substitutes," Proc. of the 38th Int. ACM

SIGIR Conf. on Research and Development in Information Retrieval (SIGIR), pp. 43–52, DOI: 10.1145/2766462.2767755, 2015.
[37] O. Shchur, M. Mumme, A. Bojchevski and S. Günnemann, "Pitfalls of Graph Neural Network Evaluation," [Online], Available: https://doi.org/10.48550/arXiv.1811.05868, 2018.
[38] H. Pei et al., "Geom-GCN: Geometric Graph Convolutional Networks," Proc. of the Int. Conf. on Learning Representations (ICLR), DOI:10.48550/arXiv.2002.05287, 2020.
[39] Z. Xu et al., "Node Classification Beyond Homophily: Towards a General Solution," Proc. of the 29th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD), pp. 2862–2873, 2023.
[40] S. Luan et al., "When Do Graph Neural Networks Help with Node Classification: Investigating the Homophily Principle on Node Distinguishability," NeurIPS J., vol. 36, pp. 28748–28760, 2023.
[41] T. N. Kipf and M. Welling, "Semi-supervised Classification with Graph Convolutional Networks," Proc. of the Int. Conf. on Learning Representations (ICLR), DOI: 10.18178/wcse.2019.06.016, 2016.
[42] D. Bo, X. Wang, C. Shi and H. Shen, "Beyond Low-frequency Information in Graph Convolutional Networks," Proc. of the AAAI Conf. on Artificial Intell. (AAAI), vol. 35, no.5, pp. 3950–3957, 2021.
[43] R. Hart, L. Yu, Y. Lou and F. Chen, "Improvements on Uncertainty Quantification for Node Classification *via* Distance-based Regularization," NeurIPS, vol. 36, pp. 55454–55478, 2023.
[44] J. Jeong et al., "iGraphMix: Input Graph Mixup Method for Node Classification," Proc. of the 12th Int. Conf. on Learning Representations (ICLR), DOI: 10.1145/3442381.3449796, 2024.
[45] H. Zhao, A. Chen, X. Sun, H. Cheng and J. Li, "All in One and One for All: A Simple Yet Effective Method towards Cross-domain Graph Pre-training," Proc. of the 30th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD), pp. 4443–4454, DOI: 10.1145/3637528.3671913, 2024.
[46] J. Tang, J. Li, Z. Gao and J. Li, "Re-thinking Graph Neural Networks for Anomaly Detection," Proc. of the 39th Int. Conf. on Machine Learning (ICML), vol.162, pp. 21076–21089, Baltimore, USA, 2022.
[47] M.-H. Guo et al., "PCT: Point Cloud Transformer," Computational Visual Media, vol. 7, pp. 187–199, 2021.
[48] Z. Liu et al., "GraphPrompt: Unifying Pre-training and Downstream Tasks for Graph Neural Networks," Proc. of the ACM on Web Conf. (WWW'23), pp. 417—428, DOI: 10.1145/3543507.3583386, 2023.
[49] X. Yu, C. Zhou, Y. Fang and X. Zhang, "Text-free Multi-domain Graph Pre-training: Toward Graph Foundation Models," [Online], Available: https://doi.org/10.48550/arXiv.2405.13934, 2024.
[50] S. Wang et al., "Multi-domain Graph Foundation Models: Robust Knowledge Transfer *via* Topology Alignment," [Online], Available: https://doi.org/10.48550/arXiv.2502.02017, 2025.
[51] X. Yu et al., "SAMGPT: Text-free Graph Foundation Model for Multi-domain Pre-training and Cross-domain Adaptation," Proc. of the ACM on Web Conf. (WWW), pp. 1142–1153, DOI: 10.1145/3696410.3714828 2025.
[52] Y. Huang et al., "One Prompt Fits All: Universal Graph Adaptation for Pre-trained Models," [Online], Available: https://doi.org/10.48550/arXiv.2509.22416, 2025.

ملخص البحث:

تُعالج هذه الورقة مشكلة النّقل السّلبي في التّدريب المسبق للرّسوم البيانية عبر المجالات في ظلّ سيناريوهات التّعلُّم بعددٍ قليلٍ من الأمثلة، وتقترح إطار عملٍ للتّدريب المسبق متعدّد المكوّنات يُسمّى (نظام تنسيق التّدريب المسبق المعزَّز بالانتباه الخارجي للرسوم البيانية). ويدمج هذا الأطار الانتباه الخارجي متعدّد الرّؤوس مع منسّق الرُّسوم البيانية، علماً بأنّ الأساليب التّقليدية تفتقر للقُدرة على التّكيف مع التّفاعلات المعقّدة والديناميكية، وأنّ معالجة التّباينات الهيكلية والدّلالية بين الرّسوم البيانية عبر المجالات تُعدّ أمراً بالغ الأهمية.

وقد بينت التّجارب على عشر مجموعات بياناتٍ للرّسوم البيانية تنتمي للعالم الحقيقي أنّ النّموذج المقترح أظهر قُدرةً فائقةً على التّعميم عبر المجالات، وأنّه كان ذا أداءٍ مُحَسَّن في مهامّ التّصنيف المتعلّقة بالتّعلُّم بعددٍ قليلً من الأمثلة، مع أفضليةٍ للنّموذج المقترح بلغت 13.3% مقارنةً بالطُّرق الأخرى. حيث يمكن للنّظام المقترح التركّيز بصورةٍ أدقّ على البِنى الحرجة للرّسوم البيانية، موفِّراً بذلك أساساً نظرياً وعملياً لتوظيف الشّبكات العصبية للرّسوم البيانية في السّيناريوهات المعقّدة.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).