UniCom: Towards a Unified and Cohesiveness-aware Framework for Community Search and Detection

Yifan Zhu
University of New South Wales
yifan.zhu4@student.unsw.edu.au

Hanchen Wang
University of Technology Sydney
hanchen.wang@uts.edu.au

Wenjie Zhang
University of New South Wales
wenjie.zhang@unsw.edu.au

Alexander Zhou
Hong Kong Polytechnic University
alexander.zhou@polyu.edu.hk

Ying Zhang
University of Technology Sydney
ying.zhang@uts.edu.au

ABSTRACT

Searching and detecting communities in real-world graphs underpins a wide range of applications. Despite the success achieved, current learning-based solutions regard community search, i.e., locating the best community for a given query, and community detection, i.e., partitioning the whole graph, as separate problems, necessitating task- and dataset-specific retraining. Such a strategy limits the applicability and generalization ability of the existing models. Additionally, these methods rely heavily on information from the target dataset, leading to suboptimal performance when supervision is limited or unavailable. To mitigate this limitation, we propose UniCom, a unified framework to solve both community search and detection tasks through knowledge transfer across multiple domains, thus alleviating the limitations of single-dataset learning. UniCom centers on a Domain-aware Specialization (DAS) procedure that adapts on the fly to unseen graphs or tasks, eliminating costly retraining while maintaining framework compactness with a lightweight prompt-based paradigm. This is empowered by a Universal Graph Learning (UGL) backbone, which distills transferable semantic and topological knowledge from multiple source domains via comprehensive pre-training. Both DAS and UGL are informed by local neighborhood signals and cohesive subgraph structures, providing consistent guidance throughout the framework. Extensive experiments on both tasks across 16 benchmark datasets and 22 baselines have been conducted to ensure a comprehensive and fair evaluation. UniCom consistently outperforms all state-of-the-art baselines across all tasks under settings with scarce or no supervision, while maintaining runtime efficiency.

PVLDB Reference Format:
Yifan Zhu, Hanchen Wang, Wenjie Zhang, Alexander Zhou, and Ying Zhang. UniCom: Towards a Unified and Cohesiveness-aware Framework for Community Search and Detection. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Yifan-Andy/UniCom.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

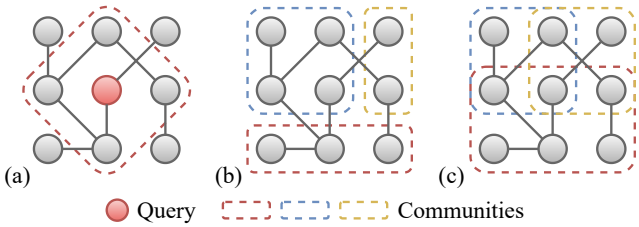


Figure 1: Overview of core tasks handled by UniCom. (a) Community Search (CS), (b) Disjoint Community Detection (DCD), and (c) Overlapping Community Detection (OCD).

1 INTRODUCTION

Graphs have shown remarkable effectiveness in modeling the relationships and dependencies between objects, enabling applications across multiple real-world domains, including social networks [17, 24], e-commerce networks [51], and citation networks [78], with each domain referring to a distinct source or type of graph. In particular, graphs from different domains often exhibit diverse structures, varying feature semantics, and differences in scale and sparsity. A community [46, 69], as a graph-derived pattern, denotes a subgraph in which nodes exhibit dense intra-connections both structurally and feature-wise. Finding communities underpins various practical applications, such as fraud detection [38, 82, 83] and recommendation systems [87]. As depicted in Figure 1, we focus on these tasks: *Community Search* (CS) [14, 34], which identifies a community based on a given query, and *Community Detection* (CD) [22, 71], which partitions the entire graph into multiple communities. Notably, in these tasks, communities may be either disjoint or overlapping based on whether nodes belong to multiple communities.

Due to the importance of community-level tasks, numerous solutions [16, 54, 85] have been proposed. Specifically, traditional algorithms [1, 15, 63, 65, 68, 86] primarily rely on rigid predefined constraints such as k-core [14], k-truss [33], or k-clique [13], which are rarely satisfied in real-world networks and render them overly *sensitive to graph density*. Moreover, they generally *overlook the importance of node features*. GNN-based methods [23, 34, 52, 72, 76] mitigate the drawback by leveraging node features. However, their dataset- and task-specific training often leads to *slow convergence* and *instability* when adapting to different graphs. One potential solution is to involve a pre-training stage [70, 81] before task-specific tuning, which assists the model in initializing with generalizable representations. Nevertheless, on small graphs (e.g., Facebook [42]), these methods often *fail to capture sufficient information* for effective

Table 1: Comparison with Existing Methods.

Approaches	CS	DCD	OCD	Multi-domain Knowledge
SMN [52]	✓	✗	✗	✗
CCGC [76]	✗	✓	✗	✗
NOCD [50]	✗	✗	✓	✗
UCoDe [44]	✗	✓	✓	✗
MDGFM [64]	✗	✗	✗	✓
UniCom	✓	✓	✓	✓

model initialization, and on large graphs (e.g., Reddit [24]), it becomes challenging to extract comprehensive information from the entire graph due to *computational and memory constraints*. Recently, graph foundation models (GFMs) [55, 64, 84] have been proposed as a general solution to graph tasks with effective and robust representation learning. Nonetheless, these models are specifically designed for node- and graph-level classification tasks, overlooking the importance of *cohesive information* and failing to adapt effectively to *community-level tasks*. In addition, graphs often differ in structure and feature distribution, while existing GFMs lack effective alignment and fusion mechanisms for transfer [64, 88], leading to *information loss or conflicts* during adaptation.

Based on these observations, existing methods either lack multi-domain knowledge or fail to effectively solve community-level tasks. In addition, they typically *treat CS and CD as separate problems*, which leads to the requirement for multiple task-specific models or dataset-specific training processes. We provide a concise comparison between UniCom and existing works in Table 1 to further illustrate the current limitations. These observations indicate that leveraging multi-domain knowledge for both CS and CD under a unified framework remains largely unexplored, presenting three major challenges. **(1) Local and global cohesiveness.** In feature-rich graphs, each node should remain cohesive with community members at both local and global levels. Maintaining cohesiveness is particularly challenging for semantically relevant but distant nodes. **(2) Information loss and mismatch.** Effective cross-domain transfer hinges on aligning the source and target feature spaces. It is challenging to align feature spaces without incurring information loss or domain-related semantic mismatches. **(3) Effective knowledge fusion strategy.** Knowledge from different sources often exhibits distinct focuses, and direct fusion may lead to substantial conflicts, making multi-domain knowledge integration challenging due to potential mutual interference.

To solve the aforementioned challenges, we introduce UniCom, a unified approach that jointly tackles CS and CD by leveraging cross-domain knowledge transfer. Specifically, the proposed framework focuses on a Domain-aware Specialization (DAS) stage for community-level tasks, which is supported by a Universal Graph Learning (UGL) backbone. To tackle **Challenge (1)**, we generate input token vectors for each node in two stages. First, we construct local subgraphs through a conductance-based procedure that yields subgraphs with high internal cohesion and low external connectivity. Second, we introduce a cohesive subgraph prompting mechanism that integrates structure- and feature-level information, enabling the model to capture relevant yet potentially long-range node dependencies. Targeting **Challenge (2)**, we introduce a domain-adaptation prompt and a feature-alignment projector that jointly address both semantic consistency and domain discrepancy. Specifically, the prompting mechanism injects learnable prompts into input tokens, enabling efficient cross-domain adaptation with only a small number of trainable parameters while keeping the pre-trained backbone frozen. Furthermore, the projection module aligns the feature dimensionalities of the two domains, preserving domain-specific information with minimal distortion. To mitigate **Challenge (3)**, we develop a multi-domain fusion module that leverages multiple pre-trained expert models, each specializing in a different domain. The experts perform independent prompt tuning to produce domain-specific predictions, which are then fused to capture complementary knowledge and enhance overall decision quality. Our major contributions can be summarized as follows:

- To the best of our knowledge, the proposed UniCom is the first unified framework for diverse community-level tasks, which jointly addresses CS and CD tasks for both disjoint and overlapping communities.
- We implement a task- and dataset-aware cohesive subgraph prompt and design a strategy to automatically generate multi-hop subgraphs, collectively ensuring local and global cohesiveness to support community-level tasks.
- We present a prompt-tuning framework with a task-aware fusion module for multi-domain knowledge integration. This framework enables knowledge integration by mitigating interference across domains and tasks.
- We conduct extensive experiments on 22 baselines and 16 real-world datasets, covering both disjoint and overlapping types, to comprehensively demonstrate the consistently superior effectiveness of UniCom for both CS and CD tasks.

2 RELATED WORKS

2.1 Community Search and Detection

Community search [8, 10, 21, 61, 66, 67] is a fundamental task that discovers query-based subgraphs with high cohesiveness. Traditional approaches [1, 15, 63, 65, 68] rely on predefined structural constraints such as k-core [14], k-truss [33], and k-clique [13], yet they fall short in effectively integrating multi-dimensional node attributes. Recently, growing efforts have been devoted to deep learning-based solutions. ICS-GNN [23] and SMN [52] search for query-dependent communities with user-defined size, whereas QD-GNN [34] and COCLEP [37] predict community scores via GNNs and select members through thresholding. Additionally, CGNP [18] and IACS [19] adopt meta-learning to enhance performance, and CommunityDF [11] utilizes diffusion-based modeling to strengthen community search capability further. Disjoint community detection [27, 39, 47, 58, 77] learns node representations to assign nodes into separate clusters. Specifically, CCGC [76] employs contrastive learning and pseudo-labels to train GNNs in a two-stage manner. In addition, DyFSS [89] fuses diverse pre-training strategies, and FPGC [72] integrates cluster-specific features with squeeze-and-excitation blocks to decouple feature representations. Overlapping community detection [53, 86] allows nodes to exist in multiple communities. NOCD [50] employs a Bernoulli-Poisson decoder with GNNs. DynaResGCN [45] subsequently introduces residual connections to improve performance, while UCoDe [44] designs a multi-task model for both disjoint and overlapping community

Table 2: The Summary of Notations.

Notation	Definition
G, C	Graph and community set.
$\mathcal{V}$, $\mathcal{E}$, $\overline{\mathcal{E}}$, $\boldsymbol{A}$	Node set, edge set, non-edge set, and adjacency matrix.
$\boldsymbol{X}$, $\boldsymbol{H}$	Input and output node feature matrices.
K	Number of communities.
d, $\hat{d}$	Input and output node feature dimensions.
m	Node token sequence length.
$\mathcal{Q}$, $\mathcal{S}$	Community search query set and label set.
C, $\overline{C}$	Community and complement of community.
$\mathcal{B}$	Set of nodes.
$\boldsymbol{P}^{feat}$, $\boldsymbol{P}^{strc}$	Feature-level prompt, and structure-level prompt.
$\boldsymbol{P}^{adp}$	Domain adaptation prompt.
$\text{Proj}(\cdot, \theta_p)$	Domain alignment projector.
$\text{GT}(\cdot, \theta_g)$	Graph transformer.

detection. SSGCAE [26] further incorporates semi-supervised learning to enhance overlapping community detection. Nevertheless, existing community search and detection methods primarily focus on the task-specific graph, overlooking the rich information in graphs from diverse domains, leading to suboptimal performance.

2.2 Graph Transfer Learning

Graph pre-training aims to equip graph neural networks (GNNs) with transferable structural and semantic knowledge via large-scale self-supervised learning, including generative [30, 31, 36] and contrastive methods [59, 70, 81]. To bridge the gap between pre-training and downstream tasks, recent studies explore two adaptation paradigms: fine-tuning [79, 91] and prompt learning [20, 41, 56, 57], both of which freeze the pre-trained model and introduce additional learnable parameters. However, these solutions mainly focus on single-domain pre-training, limiting the transferability to acquire knowledge from multiple domains. To solve the challenges in out-of-distribution generalization for cross-domain graph learning, novel foundation models have been built for textual graphs [28, 40, 90]. Moreover, text-free graph foundation models [55, 64, 84] have also been designed to train with multiple graphs from various domains, and then generalize to downstream tasks. Nevertheless, they are predominantly optimized for node or graph classifications, while neglecting important aspects such as graph cohesiveness and subgraph-level tasks.

3 PRELIMINARIES

In this paper, we focus on attributed graphs. Given a graph $G = (\mathcal{V}, \mathcal{E})$, its adjacency matrix $\boldsymbol{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ is defined such that $\boldsymbol{A}_{u,v} = 1$ if there is an edge between nodes u and v, and $\boldsymbol{A}_{u,v} = 0$ otherwise. The node feature matrix is denoted as $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}|\times d}$, where each row $\boldsymbol{X}_v$ corresponds to the d-dimensional feature vector of node $v \in \mathcal{V}$. For clarity, we summarize the main notations used in this paper in Table 2, including graph-related symbols, community and model variables, and internal mechanisms of the framework.

3.1 Problem Statement

In this work, we investigate the problem of cross-domain graph transfer learning, with the goal of developing a unified framework that jointly addresses both community search and detection tasks.

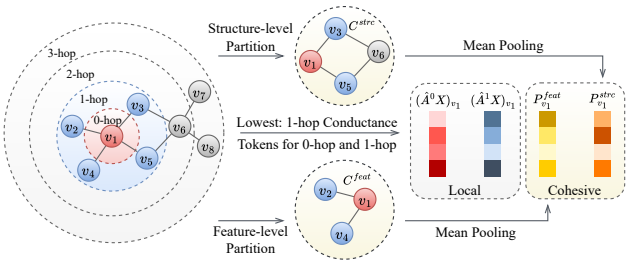


Figure 2: Dataset Pre-processing.

Definition 3.1 (Community Search [23, 52]). Given a query set $\mathcal{Q}$ containing one or more nodes per query, a graph $G = (\mathcal{V}, \mathcal{E})$, and a target size r, community search (CS) aims to identify a query-specific, cohesive subgraph of size r, as illustrated in Figure 1a.

Definition 3.2 (Community Detection [22, 44]). Given a graph $G = (\mathcal{V}, \mathcal{E})$ and a predefined community number K, the goal of community detection (CD) is to partition the node set into K subsets $\{C_1, C_2, \ldots, C_K\}$, where each $C_i \subseteq \mathcal{V}$ forms a densely connected subgraph. For the disjoint CD task, each node is assigned to exactly one community, i.e., $C_i \cap C_j = \emptyset$ for $i \neq j$, corresponding to Figure 1b. In contrast, the overlapping CD task allows a node to participate in multiple communities, i.e., $|\{i : v \in C_i\}| \geq 1$ for a node $v \in \mathcal{V}$, as shown in Figure 1c.

Based on the definitions above, CS identifies the top-r most relevant nodes for each query, where r denotes the user-specified size of the desired community. While for the CD task, a node belongs to a single community in disjoint datasets and may belong to multiple communities in overlapping datasets.

4 METHODOLOGY

In this section, we propose UniCom, a novel approach designed to address the community search (CS) and community detection (CD) tasks under a unified framework. As shown in Figure 3, the overall pipeline of UniCom consists of Domain-aware Specialization (DAS) and Universal Graph Learning (UGL), both regulated by a prompting mechanism that incorporates cohesiveness to provide task- and dataset-level guidance. The graph transformer in UGL is fully tunable, while kept frozen in DAS to prevent catastrophic forgetting. In the DAS stage, a domain adaptation prompt and a feature projector bridge the source and target domains, aided by an anchor node selector and a challenging node selector for effective alignment. In addition, a fusion module is adopted to aggregate multi-domain predictions, thereby supporting downstream tasks.

4.1 Domain-aware Specialization

Based on the pre-trained backbone neural networks, DAS aims to transfer both semantic and topological knowledge across multiple domains from source graphs $\{G_1^{src}, G_2^{src}, \ldots, G_n^{src}\}$, i.e., graphs that have been used for pre-training, to a target graph G^{tar}, i.e., a graph on which the downstream task is performed, thereby enabling effective CS or CD. To clarify the DAS procedure, we decompose it into the following five essential sections: Graph Augmentation and Alignment, Graph Transformer Architecture, Community Task Expert, Optimization, and Multi-domain Knowledge Fusion.

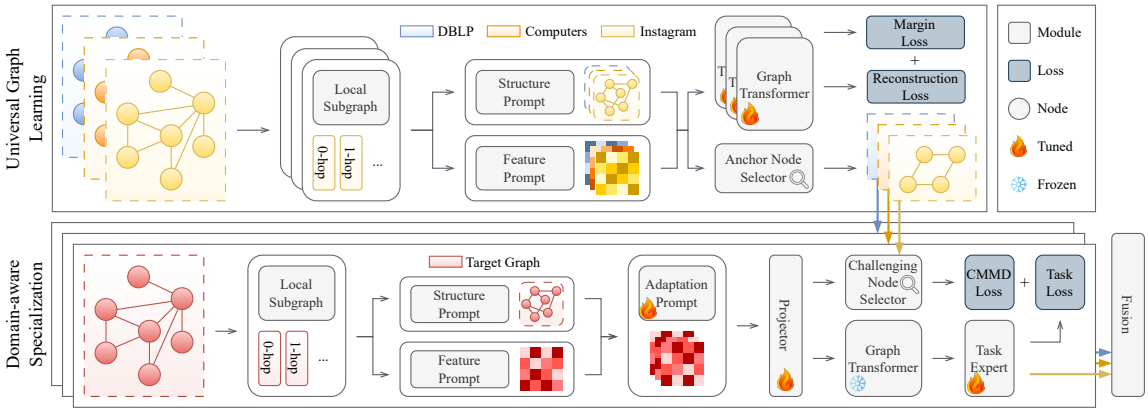


Figure 3: Overview of UniCom. 1. In Universal Graph Learning (UGL), we pre-train multiple models in parallel using datasets from different domains. 2. For Domain-aware Specialization (DAS), the pre-trained models are frozen, while anchor nodes selected from pre-training datasets are transferred. 3. Finally, outputs from multiple models are fused for the downstream task.

4.2 Graph Augmentation and Alignment

4.2.1 Local Subgraph Construction. The local subgraphs provide cohesive information around a node. However, the importance of different hops can vary significantly across nodes. To fully capture local subgraph information and adaptively select appropriate hop numbers, we aim to strike a balance by introducing a conductance-based subgraph extraction method.

Definition 4.1 (Conductance [2, 74]). Let $G = (\mathcal{V}, \mathcal{E})$ be a graph and $C \subseteq \mathcal{V}$ be a community, the conductance of C is defined as:

$$\text{Cond}\,(G, C) = \frac{\text{cut}(C, \overline{C})}{\min\left(\deg(C), \deg(\overline{C})\right)}, \tag{1}$$

where $\overline{C} = \mathcal{V} \setminus C$ is the complement of C, $\text{cut}\,(\cdot,\cdot)$ is the number of edges between nodes in C and nodes in $\overline{C}$, and $\deg(\cdot)$ denotes the sum of degrees of all nodes within a given community.

We choose the subgraph induced by the multi-hop neighbors of each given node that has the lowest conductance value as the augmented local cohesive subgraph. This conductance-based method achieves a balance between internal cohesion and external separability, thereby enabling the extraction of a meaningful local subgraph for each given node from the graph.

4.2.2 Cohesive Subgraph Prompt. Existing works [9, 88] have primarily focused on local neighborhood information for subgraph-level augmentation, thereby limiting awareness of global cohesiveness. To provide cohesive information beyond multi-hop neighbors for community members, we employ K-means [25] and Louvain [4] to extract semantic and structural information for generating the global cohesive subgraph prompt, composed of two parts: a feature prompt and a structure prompt. Specifically, K-means captures semantic similarity for the former, while Louvain detects densely connected patterns based on topology for the latter.

Let $G = (\mathcal{V}, \mathcal{E})$ be a graph with node features $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$. Applying K-means clustering to the node features yields $C^{feat} = \{C_1^{feat}, \ldots, C_{K_{feat}}^{feat}\}$, where node v belongs to C_n^{feat}. Then the feature prompt $\boldsymbol{P}_v^{feat} \in \mathbb{R}^d$ for node v is constructed as:

$$\boldsymbol{P}_v^{feat} = \frac{1}{|C_n^{feat}|} \sum_{u \in C_n^{feat}} \boldsymbol{X}_u. \tag{2}$$

Similarly, for structure-level information, we adopt Louvain to generate $C^{strc} = \{C_1^{strc}, \ldots, C_{K_{strc}}^{strc}\}$, where node v is assigned to C_n^{strc}. Then the structure prompt $\boldsymbol{P}_v^{strc} \in \mathbb{R}^d$ of node v is defined as:

$$\boldsymbol{P}_v^{strc} = \frac{1}{|C_n^{strc}|} \sum_{u \in C_n^{strc}} \boldsymbol{X}_u. \tag{3}$$

Subsequently, the conductance-augmented features are prompted by cohesive subgraph prompts through concatenation:

$$\boldsymbol{X}^{coh} = \text{Concat}\left(\boldsymbol{X}^{aug}, \boldsymbol{P}^{feat}, \boldsymbol{P}^{strc}\right) \in \mathbb{R}^{|\mathcal{V}| \times m \times d}, \tag{4}$$

where $\boldsymbol{X}^{aug}$ denotes the conductance-augmented feature matrix that incorporates features from each node and the corresponding multi-hop neighbors. This design is analogous to prompt-based language models [7], where external tokens are appended to inputs to provide task- and data-specific guidance. After prompting, each node is associated with m token vectors, including the node feature, representations from multi-hop neighbors, and two cohesive subgraph prompts. The prompted feature matrix is denoted as $\boldsymbol{X}^{coh}$, which incorporates both local neighborhood information and global cohesive prompts, thus facilitating downstream learning. The overall workflow for local subgraph construction and cohesive subgraph prompting is delineated in Algorithm 1.

Example 1. *Given the graph shown in Figure 2, for node v_1, its 1-hop neighborhood contains nodes v_2, v_3, v_4, and v_5. The conductance of the 1-hop local subgraph is thus $\frac{2}{4+1+1} = \frac{1}{3}$. Similarly, the conductance of the 2-hop local subgraph is 1. Since the conductance of the 1-hop neighborhood is the lowest, it is selected as the local subgraph. We then perform feature propagation to obtain the 0-hop and 1-hop feature tokens for node v_1, thus forming $\boldsymbol{X}_{v_1}^{aug}$. The feature- and structure-level partitions yield multiple communities, and C^{feat} and C^{strc} refer to the communities to which node v_1 belongs in each partition. Then $\boldsymbol{P}_{v_1}^{feat}$ and $\boldsymbol{P}_{v_1}^{strc}$ are constructed according to Equation 2 and Equation 3.*

Algorithm 1: Dataset Pre-processing.

Input : Graph $G = (\mathcal{V}, \mathcal{E})$, adjacency matrix $\boldsymbol{A}$, feature matrix $\boldsymbol{X}$.
Output: Prompted cohesive subgraph feature matrix $\boldsymbol{X}^{coh}$.

$\hat{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$;
$\mathcal{H} = \{\hat{\boldsymbol{A}}^0\boldsymbol{X}, \hat{\boldsymbol{A}}^1\boldsymbol{X}, \ldots, \hat{\boldsymbol{A}}^h\boldsymbol{X}\}$;
for $v \in \mathcal{V}$ **do**
  $\hat{h} \leftarrow 0$;
  $c_{min} \leftarrow \infty$;
  `// Select the hop depth with minimum conductance.`
  for $i = 0, \ldots, h$ **do**
    Determine the i-hop neighborhood C_i of v;
    $c \leftarrow \mathrm{Cond}(G, C_i)$;
    if $c < c_{min}$ **then** $\hat{h} \leftarrow i$; $c_{min} \leftarrow c$;
  $\boldsymbol{X}_v^{aug} \leftarrow \mathrm{Concat}\left(\mathcal{H}[0]_v, \ldots, \mathcal{H}[\hat{h}]_v\right)$;
  Compute $\boldsymbol{P}_v^{feat}$ (Eq. 2);
  Compute $\boldsymbol{P}_v^{strc}$ (Eq. 3);
  $\boldsymbol{X}_v^{coh} \leftarrow \mathrm{Concat}\left(\boldsymbol{X}_v^{aug}, \boldsymbol{P}_v^{feat}, \boldsymbol{P}_v^{strc}\right)$;
return $\boldsymbol{X}^{coh}$;

4.2.3 Domain Adaptation Prompt. To mitigate the distribution gap between source and target domains, we propose a lightweight yet effective graph prompt learning method inspired by the "pre-training and prompting" paradigm, which enables effective cross-domain adaptation with learnable prompts while keeping the pre-trained model frozen. Specifically, a set of shared and trainable basis prompt vectors $\{\boldsymbol{P}_i^{bas}\}_{i=1}^{N_p}$ is introduced, where $\boldsymbol{P}_i^{bas} \in \mathbb{R}^d$, and N_p denotes the number of basis prompts [20]. For each node v and token position t, an adaptation prompt $\boldsymbol{P}_{v,t}^{adp}$ is constructed as a weighted combination of all basis prompt vectors:

$$\boldsymbol{P}_{v,t}^{adp} = \sum_{i=1}^{N_p} \omega_{v,t,i}\boldsymbol{P}_i^{bas}, \tag{5}$$

where $\omega_{v,t,i}$ is a learnable attention weight indicating the importance of the i-th prompt for node v at position t. Then, the adaptation prompt is integrated with the cohesiveness-aware features via element-wise addition to obtain the prompted feature matrix $\boldsymbol{X}^{adp}$. Serving as a dual-channel bridge, the adaptation prompt aligns the feature distribution across domains. However, challenges arising from inconsistent feature dimensionality persist.

4.2.4 Feature Alignment. A projector is introduced between the source and target graphs to align feature dimensions across domains and enable effective knowledge transfer, formulated as:

$$\boldsymbol{Z} = \mathrm{Proj}\left(\boldsymbol{X}^{adp}, \theta_p\right), \tag{6}$$

where $\mathrm{Proj}\left(\cdot, \theta_p\right)$ is a learnable function parameterized by θ_p, for example, a fully-connected layer or a multi-layer perceptron (MLP). Compared with traditional dimensionality reduction techniques such as Singular Value Decomposition (SVD), which overlook semantic consistency across dimensions, the learnable projector is designed to preserve and align informative features during transfer.

4.3 Graph Transformer Architecture

Following prior works [60, 80], we adopt a graph transformer to efficiently capture long-range dependencies and obtain expressive representations of graph-structured data. Unlike *NAGphormer* [9], which solely relies on multi-hop node features, our method incorporates conductance-augmented features and cohesive subgraph prompts as input. Moreover, the adaptation prompt and projector are additionally introduced during the DAS stage prior to the graph transformer. Given the projected feature matrix $\boldsymbol{Z}$, we then obtain the encoded node features as:

$$\boldsymbol{H} = \mathrm{GT}\left(\boldsymbol{Z}, \theta_g\right), \tag{7}$$

where $\mathrm{GT}(\cdot, \theta_g)$ denotes the graph transformer encoder with frozen parameters θ_g pre-trained in UGL. This graph encoder for UniCom contains an initial layer that projects input to a hidden dimension, followed by L transformer encoder layers. The transformer encoder contains three major sub-modules: positional encoding, multi-head attention, and feed-forward network. Specifically, given an input embedding $\boldsymbol{H}_v^l$ for node v at layer l, position encoding is first added to the input embedding, following the approach of NAGphormer [9]. A single attention block is defined as:

$$\mathrm{Attention}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d^{(l+1)}}}\right)\boldsymbol{V}, \tag{8}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ denote the query, key, and value matrices derived from the input embeddings, and $d^{(l+1)}$ is the dimensionality of the attention space at layer $l+1$. The multi-head attention mechanism consists of many parallel attention blocks. For each attention head Head_i, $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$, and $\boldsymbol{V}_i$ are obtained via linear projections, and attention is computed independently. The outputs from all heads are then concatenated and passed through a learnable weight matrix for aggregation, and the multi-head attention module is $\mathrm{MHA}(\cdot)$.

In our implementation, we apply layer normalization (LN) [3] to the input embeddings before the multi-head attention module, and then apply another layer normalization followed by a feed-forward network (FFN), both with residual connections:

$$\begin{aligned} \boldsymbol{H}_v^{(l+1)} &= \mathrm{MHA}\left(\mathrm{LN}\left(\boldsymbol{H}_v^{(l)}\right)\right) + \boldsymbol{H}_v^{(l)} \\ \boldsymbol{H}_v^{(l+1)} &= \mathrm{FFN}\left(\mathrm{LN}\left(\boldsymbol{H}_v^{(l+1)}\right)\right) + \boldsymbol{H}_v^{(l+1)}, \end{aligned} \tag{9}$$

where FFN is composed of two linear layers with a GELU [29] nonlinearity in between. After passing through L layers of transformer blocks, we obtain the final node and community embeddings from the output representation $\boldsymbol{H}_v^L$. Specifically, the embedding at the first token position is used as the node-level representation, while the remaining tokens correspond to the community-level representations. To ensure both representations have consistent dimensions for computations, we apply a mean pooling operation over the community-level tokens. The output embeddings are denoted as $\boldsymbol{H}^{node}, \boldsymbol{H}^{com} \in \mathbb{R}^{|\mathcal{V}| \times \hat{d}}$, where $\hat{d}$ is the model output dimension.

4.4 Community Task Expert

For both CS and CD tasks, a combination of $\boldsymbol{H}^{node}$ and $\boldsymbol{H}^{com}$ is used as the representation of all nodes, incorporating both node- and community-level information. To identify the most relevant community of size r for each query in the CS task, we introduce a

lightweight cross-attention mechanism to capture the interaction between the query and graph nodes. Specifically, attention scores between representations of nodes in the query set and all potential nodes in the graph are computed. Then, top-r nodes from the graph with the highest scores form the predicted community.

For the disjoint CD task, K-means clustering is employed as the task expert on the encoded node representations to partition the graph into communities. In contrast, the overlapping CD task intends to produce a soft community affiliation matrix $\hat{\boldsymbol{Y}} \in \mathbb{R}^{|\mathcal{V}| \times K}$, where $\hat{\boldsymbol{Y}}_{ij} \in [0, 1]$ denotes the probability that node v_i belongs to community C_j. In practice, we employ a single MLP with an activation function (e.g., ReLU) as the decoder to project the embeddings to a target space matching the number of communities, thereby producing community assignment scores for each node.

4.5 Optimization

4.5.1 Cross-domain Alignment. Previous methods [64, 84] typically optimize for a single task-driven objective across both domain adaptation and task learning, which limits the quality of cross-domain alignment. Moreover, others [79, 88] tend to bridge all source and target samples jointly, which is computationally inefficient. To guide the learning and refinement of the adaptation prompt and projector while preserving efficiency, we propose representative node selection strategies to emphasize key node features in both source and target graphs. Specifically, we first perform clustering (e.g., K-means) on source graphs to form pseudo-communities. The centroids of these communities are regarded as a set of anchor nodes with the most critical feature information, denoted as $\mathcal{B}^{src}$. For each target graph, motivated by hard sampling strategy [35], we extract the most challenging nodes for alignment, i.e., those whose projected features exhibit the lowest cosine similarity to the augmented features of nodes in $\mathcal{B}^{src}$, denoted as $\mathcal{B}^{tar}$.

Based on the selected representative nodes from both domains, we optimize the adaptation prompt and the projector using Community Maximum Mean Discrepancy (CMMD), an MMD-inspired objective [6] that enables effective domain alignment without requiring labeled data. Specifically, the CMMD loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{cmmd}} = & \frac{1}{|\mathcal{B}^{tar}|^2} \sum_{u,u' \in \mathcal{B}^{tar}} \kappa(\boldsymbol{Z}_u^{tar}, \boldsymbol{Z}_{u'}^{tar}) \\ & - \frac{2}{|\mathcal{B}^{src}||\mathcal{B}^{tar}|} \sum_{u \in \mathcal{B}^{tar}} \sum_{v \in \mathcal{B}^{src}} \kappa(\boldsymbol{Z}_u^{tar}, \boldsymbol{X}_v^{src}) \\ & + \frac{1}{|\mathcal{B}^{src}|^2} \sum_{v,v' \in \mathcal{B}^{src}} \kappa(\boldsymbol{X}_v^{src}, \boldsymbol{X}_{v'}^{src}),\end{aligned} \tag{10}$$

where $\kappa(\cdot, \cdot)$ denotes the Gaussian kernel function. The source representation $\boldsymbol{X}^{src}$ is derived by applying mean pooling to the node features in $\mathcal{B}^{src}$ after cohesive subgraph prompting. Similarly, the target representation $\boldsymbol{Z}^{tar}$ is computed by applying mean pooling to the projected node features in $\mathcal{B}^{tar}$.

4.5.2 Multi-task Learning. Beyond cross-domain alignment, we introduce training objectives for multiple downstream tasks. For the CS task, given a training dataset consisting of a query set $\mathcal{Q}$ and the corresponding labels $\mathcal{S}$, we optimize the trainable parameters with binary cross-entropy loss, guided by per-query supervision. The task-specific binary cross-entropy loss function is:

$$\mathcal{L}_{\text{cs}} = -\frac{1}{|\mathcal{Q}|} \sum_{q=1}^{|\mathcal{Q}|} \frac{1}{|\mathcal{S}_q|} \sum_{v \in \mathcal{S}_q} y_{q,v} \log\left(\hat{y}_{q,v}\right) + \left(1 - y_{q,v}\right) \log\left(1 - \hat{y}_{q,v}\right), \tag{11}$$

where $\hat{y}_{q,v}$ is the prediction score for node v given the q-th query in $\mathcal{Q}$, and $y_{q,v}$ is the corresponding ground-truth label from $\mathcal{S}$.

Disjoint CD aims to learn node embeddings that capture both feature and structural variations, enabling effective clustering. To encourage similar embeddings among intra-community nodes while reducing similarity across communities, we retain the margin loss and reconstruction loss from the pre-training stage for domain adaptation purposes, which will be formally introduced in Section 4.7. Moreover, a pseudo-label-based refinement loss [76] is employed, where high-quality labels are generated from nodes whose node- and community-level embeddings are most similar to each other. Specifically, the refinement loss is formulated as:

$$\mathcal{L}_{\text{dcd}} = \frac{1}{K} \sum_{i=1}^{K} \sum_{v \in \mathcal{B}_i^{pos}} \left(2 - 2 \cdot \text{sim}\left(\boldsymbol{H}_v^{node}, \boldsymbol{H}_v^{com}\right)\right) + \|\boldsymbol{S} - \text{diag}(\boldsymbol{S})\|_F^2, \tag{12}$$

where $\mathcal{B}_i^{pos}$ denotes the subset of high-confidence nodes assigned to the cluster i, selected based on the embedding cosine similarity between $\boldsymbol{H}^{node}$ and $\boldsymbol{H}^{com}$. $\boldsymbol{S} \in \mathbb{R}^{K \times K}$ is the inter-class similarity matrix computed between the mean embeddings (centers) of each cluster in the two views. The term $\|\boldsymbol{S} - \text{diag}(\boldsymbol{S})\|_F^2$ removes the diagonal elements and penalizes the off-diagonal similarities, encouraging better separation between different communities.

For the overlapping CD task, we follow [45, 50] by minimizing the Bernoulli–Poisson negative log-likelihood, aligning community memberships of connected nodes and separating unconnected pairs based on the soft assignment matrix $\hat{\boldsymbol{Y}}$. The loss is defined as:

$$\mathcal{L}_{\text{ocd}} = \frac{1}{|\mathcal{E}|} \sum_{(u,v) \in \mathcal{E}} -\log\left(1 - \exp\left(-\epsilon - \hat{\boldsymbol{Y}}_u^\top \hat{\boldsymbol{Y}}_v\right)\right) + \frac{1}{|\overline{\mathcal{E}}|} \sum_{(u,v) \in \overline{\mathcal{E}}} \hat{\boldsymbol{Y}}_u^\top \hat{\boldsymbol{Y}}_v, \tag{13}$$

where ϵ serves as a small non-negative offset to stabilize the likelihood function. Specifically, the first term increases the probability that connected node pairs belong to the same community, while the second term reduces the probability that unconnected node pairs are assigned to the same community, based on the assignment matrix $\hat{\boldsymbol{Y}}$, thereby enhancing structural consistency.

4.5.3 Overall Objective Function. Building upon the pre-trained models, we propose a framework that incorporates prompting and projection mechanisms to adapt the target dataset features. Furthermore, a task-specific decoder is employed to facilitate interaction between the learned knowledge and the downstream task. The overall learning objective jointly optimizes both feature alignment and task performance, which is formally defined as:

$$\mathcal{L}_{\text{das}} = \mathcal{L}_{\text{task}} + \alpha \mathcal{L}_{\text{cmmd}}, \tag{14}$$

where $\alpha > 0$ is a trade-off parameter, and $\mathcal{L}_{\text{task}}$ denotes the corresponding task-specific objective (i.e., $\mathcal{L}_{\text{cs}}$, $\mathcal{L}_{\text{dcd}}$, or $\mathcal{L}_{\text{ocd}}$).

Algorithm 2: Domain-aware Specialization.

Input : Source graphs $\{G_1^{src}, \ldots, G_n^{src}\}$, target graph G^{tar}, pre-trained models $\{\mathrm{GT}_1, \ldots, \mathrm{GT}_n\}$.
Output: Task-specific prediction π_{out}.

$\boldsymbol{X}^{tar} \leftarrow$ Algorithm 1(G^{tar});
for $i = 1, \ldots, n$ **do**
  $\boldsymbol{X}_i^{src} \leftarrow$ Algorithm 1(G_i^{src});
  $\mathcal{B}_i^{src} \leftarrow \mathrm{AnchorNodeSelection}(\boldsymbol{X}_i^{src})$;
  while *not converged* **do**
    $\boldsymbol{X}_i^{adp} \leftarrow \boldsymbol{X}^{tar} + \boldsymbol{P}_i^{adp}$;
    $\boldsymbol{Z}_i \leftarrow \mathrm{Proj}_i(\boldsymbol{X}_i^{adp}, \theta_{p,i})$;
    $\mathcal{B}_i^{tar} \leftarrow \mathrm{ChallengingNodeSelection}(\mathcal{B}_i^{src}, \boldsymbol{X}_i^{src}, \boldsymbol{Z}_i)$;
    Compute $\mathcal{L}_{\mathrm{cmmd}}$ (Eq. 10);
    $\boldsymbol{H}_i \leftarrow \mathrm{GT}_i(\boldsymbol{Z}_i, \theta_{g,i})$;
    $\pi_i \leftarrow \mathrm{CommunityTaskExpert}(\boldsymbol{H}_i)$;
    Compute $\mathcal{L}_{\mathrm{task}}$ (Eqs. 11, 12, 13);
    $\mathcal{L}_{\mathrm{das}} \leftarrow \mathcal{L}_{\mathrm{task}} + \alpha\, \mathcal{L}_{\mathrm{cmmd}}$;
    Update $\boldsymbol{P}_i^{adp}$ and $\theta_{p,i}$ with $\mathcal{L}_{\mathrm{das}}$;
  $\boldsymbol{X}_i^{adp} \leftarrow \boldsymbol{X}^{tar} + \boldsymbol{P}_i^{adp}$;
  $\boldsymbol{Z}_i \leftarrow \mathrm{Proj}_i(\boldsymbol{X}_i^{adp}, \theta_{p,i})$;
  $\boldsymbol{H}_i \leftarrow \mathrm{GT}_i(\boldsymbol{Z}_i, \theta_{g,i})$;
  $\pi_i \leftarrow \mathrm{CommunityTaskExpert}(\boldsymbol{H}_i)$;
$\pi_{out} \leftarrow \mathrm{Fusion}(\pi_1, \ldots, \pi_n)$;
return π_{out};

4.6 Multi-domain Knowledge Fusion

Finally, domain-specific structures and feature distributions in each source dataset bias the model through transfer learning, causing predictions to focus on different node regions. As a consequence, relying on a single prediction may introduce bias and limit generalization to the target task. To alleviate biases and balance the complementary inductive patterns captured by distinct pre-training datasets, we apply a task-aware fusion approach under a unified framework, combining the advantages of multiple predictions.

For the CS task, we compute the average prediction score from all models for each node, and then select the top-r nodes with the highest scores as the predicted community. To leverage transferred knowledge for disjoint CD, we concatenate embeddings from all domains and apply a task-specific expert for the final prediction. For the overlapping CD task, we first align the multi-model predictions via consistent label mapping to avoid mismatched community indices across models, where the mapping is obtained by matching each model's probability matrix to that of the first model using cosine similarity and the Hungarian algorithm. Next, we average these matrices and apply a thresholding operation to transform soft assignments into discrete community memberships. We further demonstrate the full DAS pipeline as presented in Algorithm 2.

Lemma 1 (Risk Reduction with Fusion). *For the multi-domain knowledge fusion, the CS classification risk never exceeds that of each individual prediction. The same monotonicity property applies to the DCD joint distortion and the OCD self-supervised loss.*

Lemma 1 provides theoretical support for the effectiveness of multi-domain knowledge fusion, and the proof is given in Section 5.

4.7 Universal Graph Learning

Multi-domain transfer learning enables the acquisition of extensive knowledge from diverse pre-training datasets. However, due to the significant discrepancies in both feature distributions and structural properties of graphs across domains, a single unified model is prone to harmful cross-domain interference. Thus, we train an expert model for each pre-training dataset to minimize the information loss and preserve the independence of domain-specific information. Practically, for each source dataset, we first generate conductance-based subgraph features, which are subsequently augmented with cohesive subgraph prompts, and then encoded with the graph transformer to generate $\boldsymbol{H}^{node}$ and $\boldsymbol{H}^{com}$. Moreover, we aim to maximize the similarity between each node and its local and global contexts, while minimizing that of unrelated pairs. Thus, we introduce the margin triplet loss [49]:

$$\mathcal{L}_{\mathrm{mar}} = \frac{1}{|\mathcal{V}|^2} \sum_{u,v \in \mathcal{V}} -\max\left(\sigma\left(\boldsymbol{H}_v^{node} \boldsymbol{H}_v^{com}\right) - \sigma\left(\boldsymbol{H}_u^{node} \boldsymbol{H}_v^{com}\right) + \epsilon,\, 0\right), \tag{15}$$

where $\sigma(\cdot)$ is the activation function and ϵ is the margin value.

In addition, relying on the adjacent matrix $\boldsymbol{A}$ from the edge set $\mathcal{E}$, the reconstruction loss is formulated as follows to encourage similarity between connected node pairs while enforcing dissimilarity between unconnected ones:

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{|\mathcal{V}|^2} \sum_{u,v \in \mathcal{V}} (1 - \boldsymbol{A}_{u,v}) \left(\boldsymbol{H}_u^{node} \boldsymbol{H}_v^{node}\right) - \boldsymbol{A}_{u,v} \left(\boldsymbol{H}_u^{node} \boldsymbol{H}_v^{node}\right). \tag{16}$$

The overall training objective of the UGL phase comprises both the margin loss and the reconstruction loss:

$$\mathcal{L}_{\mathrm{ugl}} = \mathcal{L}_{\mathrm{mar}} + \beta \mathcal{L}_{\mathrm{rec}}, \tag{17}$$

where $\beta > 0$ is the coefficient to balance the importance of each objective function. These pre-trained models serve as the foundation for the DAS stage, which leverages multi-domain knowledge to support robust and transferable CS and CD across diverse graphs.

5 ANALYSIS

5.1 Effectiveness of Adaptation Prompt

We explain the effectiveness of prompt-based alignment by analyzing the geometry of the frozen encoder $\mathrm{GT}(\cdot, \theta_g)$ and establishing the existence of a bridge graph G^{bri} that connects the source and downstream representations.

Following the existing work [62], the pre-trained GNN encoder $\mathrm{GT}(\cdot, \theta_g)$ is regarded as a surjective mapping from the graph set $\mathcal{G}$ to the feature space $\mathbb{R}^{\hat{d}}$. This property ensures that every valid representation in $\mathbb{R}^{\hat{d}}$ can be obtained from at least one graph instance. Given an input graph G^{tar} and its desired downstream representation $C(G^{tar})$, we consider $C(G^{tar}) \in \mathbb{R}^{\hat{d}}$ as the embedding vector produced by the optimal downstream model for its corresponding task. According to the surjectivity of $\mathrm{GT}(\cdot, \theta_g)$, for this particular $C(G^{tar})$, there must exist a graph $\hat{G}^{bri}$ satisfying:

$$\mathrm{GT}(\hat{G}^{bri}, \theta_g) = C(G^{tar}).$$

By definition, this $\hat{G}^{bri}$ is an instance of the bridge graph G^{bri}. Hence, the existence of G^{bri} is guaranteed, demonstrating that the downstream representation $C(G^{tar})$ lies within the reachable space of the frozen encoder $\mathrm{GT}(\cdot, \theta_g)$. Therefore, for any given graph G^{tar}, there always exists a bridge graph G^{bri}, satisfying:

$$\mathrm{GT}(G^{bri}, \theta_g) = C(G^{tar}).$$

The adaptation prompt $\boldsymbol{P}^{adp}$ aims to apply the transformation to shift the G^{tar} towards the bridge region:

$$\mathrm{GT}(\boldsymbol{P}^{adp}(G^{tar}, \theta_a), \theta_g) \approx C(G^{tar}).$$

Recent work [62] proves that for a multi-layer full-rank GNN, the prompted output always takes the form:

$$\mathrm{GT}(\boldsymbol{P}^{adp}(G^{tar}, \theta_a), \theta_g) = \boldsymbol{R}^{(L)} + \boldsymbol{c}\boldsymbol{p}^{\top},$$

where $\boldsymbol{R}^{(L)}$ is the transformed base embedding after L layers, $\boldsymbol{p} \in \mathbb{R}^{\hat{d}}$ denotes the prompt-induced direction that spans the entire output space $\mathbb{R}^{\hat{d}}$, and $\boldsymbol{c}$ is a non-negative coefficient vector (with $c_i \geq 0$ and $\|\boldsymbol{c}\| > 0$) that accumulates the prompt effect across layers.

Given this rank-one structure, we next show that the adaptation prompt can reach the downstream target representation. Since $\boldsymbol{R}^{(L)}$ is fixed and the direction vector $\boldsymbol{p}$ spans $\mathbb{R}^{\hat{d}}$, modifying the prompt parameters θ_a effectively adjusts the coefficient vector $\boldsymbol{c}$, which in turn determines the rank-one update $\boldsymbol{c}\boldsymbol{p}^{\top}$ in the encoder's output space. Consequently, the term $\boldsymbol{c}\boldsymbol{p}^{\top}$ enables shifting the embedding along any direction in $\mathbb{R}^{\hat{d}}$, making the mapping

$$\theta_a \mapsto \mathrm{GT}(\boldsymbol{P}^{adp}(G^{tar}, \theta_a), \theta_g)$$

surjective onto $\mathbb{R}^{\hat{d}}$. Since $C(G^{tar})$ lies in this space and, in addition, $\mathrm{GT}(G^{bri}, \theta_g) = C(G^{tar})$, there exists some $\theta_a^{\star}$ such that

$$\mathrm{GT}(\boldsymbol{P}^{adp}(G^{tar}, \theta_a^{\star}), \theta_g) = \mathrm{GT}(G^{bri}, \theta_g) = C(G^{tar}).$$

Hence, the adaptation prompt is capable of shifting G^{tar} to its optimal bridge representation, completing the alignment.

5.2 Effectiveness of Multi-domain Fusion

5.2.1 Proof of Lemma 1. Across the CS task, DAS produces N task-specific predictions $\{\pi_n\}_{n=1}^{N}$ from different source-domain pre-training paths. We adopt a simple averaging ensemble operator:

$$\pi_{out} = \frac{1}{N}\sum_{n=1}^{N}\pi_n.$$

This formulation aggregates multiple domain-specific predictions into a single fused output. Each model $n \in \{1, \ldots, N\}$ outputs a conditional probability distribution $\pi_n(\cdot \mid G, q)$. The task loss is:

$$\mathcal{R}(\pi) = \mathbb{E}_{(G,q,y)}\left(-\log \pi(y \mid G, q)\right),$$

where $\pi(y \mid G, q)$ is the predicted probability of the true label y, with π representing any prediction under consideration, including the individual prediction π_n and the fused prediction π_{out}. Since $-\log(\cdot)$ is a strictly convex function, Jensen's inequality applied to the uniform average $\pi_{out} = \frac{1}{N}\sum_{n=1}^{N}\pi_n$ yields:

$$-\log \pi_{out}(y \mid G, q) \leq \frac{1}{N}\sum_{n=1}^{N}\left(-\log \pi_n(y \mid G, q)\right).$$

Taking expectation over (G, q, y) gives:

$$\mathcal{R}(\pi_{out}) \leq \frac{1}{N}\sum_{n=1}^{N}\mathcal{R}(\pi_n),$$

with strict inequality whenever the individual predictions $\{\pi_n(y \mid G, q)\}_{n=1}^{N}$ are not almost surely identical. Thus, the fusion in DAS incurs no higher expected risk than the average single-domain model and becomes better when the models provide complementary probability estimates based on diverse pre-training domains.

The proof for the DCD task follows from the additive decomposition of the squared Euclidean distance across views, which replaces convexity in the CS task and ensures that minimizing distortion in the concatenated space reduces the joint distortion over all views.

The proof for the OCD task follows from Jensen's inequality in the same way as in the CS task, since the OCD loss is convex in the similarity calculation $\hat{\boldsymbol{Y}}_i^{\top}\hat{\boldsymbol{Y}}_j$, exhibiting strict convexity on positive pairs and reducing to a linear form on negative pairs.

5.3 Time Complexity Analysis

5.3.1 Pre-training. The time complexity of the 3 projection operations in the graph transformer is $O\left(m \times d^2\right)$ each, and this should result in a total of $O\left(m \times d^2\right)$. The dot products between the query and key, and between the attention weights and value, should both take $O\left(m^2 \times d\right)$. Hence, training the graph transformer with L layers and for t epochs over $|\mathcal{V}|$ nodes yields a total time complexity of $O\left(t \times L \times |\mathcal{V}| \times \left(m \times d^2 + m^2 \times d\right)\right)$. Considering that t, L, and m are all hyperparameters and d is a fixed integer representing the dimension, the overall time complexity of the pre-training stage grows linearly with respect to the number of nodes $|\mathcal{V}|$.

5.3.2 Domain Adaptation. The time complexity of both weight score calculation and adaptation prompt generation for each node is $O\left(m \times N_p \times d\right)$. Then, we have the adaptation prompting time complexity being $O\left(|\mathcal{V}| \times m \times N_p \times d\right)$. The graph alignment process with an L-layer MLP projector takes $O\left(L \times |\mathcal{V}| \times m \times d^2\right)$ time complexity. Then the total time for domain adaptation by running t epochs is the sum of prompt adaptation and projection, which is shown as $O\left(t \times |\mathcal{V}| \times m \times d \times \left(N_p + L \times d\right)\right)$. Owing to its linear time complexity depending on the number of graph nodes $|\mathcal{V}|$, this design supports efficient and scalable deployment on large graphs.

5.3.3 Task Expert. For the CS task, the task expert is a lightweight cross-attention layer, where we remove the $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ matrices and directly use the query-level embedding and global node embeddings for calculation, and the query-level embedding is derived from the mean pooling of all node embeddings in a query. The calculation of attention score is $O\left(|\mathcal{V}| \times d\right)$ and the calculation of score–value multiplication is also $O\left(|\mathcal{V}| \times d\right)$, given the time complexity for a single search be $O\left(|\mathcal{V}| \times d\right)$, thus the total search time of CS task expert is $O\left(|\mathcal{Q}| \times |\mathcal{V}| \times d\right)$.

The disjoint CD employs K-means as the task expert for community membership assignment, which has a time complexity of $O\left(|\mathcal{V}| \times |C| \times d \times t\right)$ when iterated for t steps.

The overlapping CD employs a 2-layer MLP for calculating community scores, which projects node embeddings from the hidden dimension d to the number of communities $|C|$. The time complexity of this operation is $O\left(|\mathcal{V}| \times \left(d^2 + d \times |C|\right)\right)$.

Table 3: Statistics of Datasets.

Type	Dataset	# N	# E	# C	# F	OR	MLA
Pre-training	DBLP	17,716	105,734	4	1,639	-	-
	Computers	13,752	491,722	10	767	-	-
	Instagram	11,339	155,349	2	500	-	-
	WikiCS	11,701	431,726	10	300	-	-
	CoCS	18,333	163,788	15	6,805	-	-
Disjoint	Cora	2,708	10,556	7	1,433	-	-
	CiteSeer	3,327	9,104	6	3,703	-	-
	Photo	7,650	238,162	8	745	-	-
	PubMed	19,717	88,648	3	500	-	-
	Reddit	232,965	114M	41	602	-	-
	Products	2,449,029	124M	47	100	-	-
Overlapping	FB107	1,046	54,543	9	576	1.82	3
	FB1684	793	28,840	17	319	0.76	3
	FB1912	756	60,805	46	480	34.52	6
	CS	21,957	193,500	18	7,793	27.54	13
	ENG	14,927	98,610	16	4,839	27.20	12

The CS task expert has time complexity proportional to the query set size $|Q|$ and the number of nodes $|\mathcal{V}|$, while the two CD task experts witness the complexity grows linearly with respect to both the number of nodes $|\mathcal{V}|$ and the number of communities $|C|$, verifying the overall efficiency and practicality.

6 EXPERIMENTS

In this section, we report the results of comprehensive experiments using 16 datasets from diverse domains and 22 competitive baselines, covering both CS and CD tasks, to demonstrate the effectiveness and efficiency of our proposed methods.

6.1 Experimental Setup

6.1.1 Datasets. 16 datasets are used in the experiments. Evaluation of UniCom is conducted on 11 real-world datasets, covering citation networks [51, 78], social networks [24, 42], e-commerce networks [12, 51], and academic networks [50]. Specifically, Cora, CiteSeer, Photo, PubMed, Reddit, and Products are graphs with disjoint communities, while FB107, FB1684, FB1912, CS, and ENG are graphs with overlapping communities. Additionally, 5 datasets from diverse domains [5, 32, 43, 51] are adopted as pre-training sources. Detailed statistics, including the numbers of nodes, edges, and communities, along with the node feature dimensions, for all datasets are provided in Table 3. We further report Overlap Rates (OR) and Max Label Affiliations (MLA) [52] on overlapping datasets to quantify the extent of multi-community membership. Given the ground-truth label assignment Y, where Y_v denotes the set of communities to which node v belongs, the OR measures the proportion of nodes associated with more than one community:

$$\text{OR} = \frac{1}{|\mathcal{V}|}\sum_{v=1}^{|\mathcal{V}|}\mathbb{I}\big(|Y_v| > 1\big). \tag{18}$$

The MLA captures the maximum number of community affiliations held by any node within the graph:

$$\text{MLA} = \max_{v\in\mathcal{V}} |Y_v|. \tag{19}$$

6.1.2 Data Splitting. For the CS task, UniCom handles single- and multi-node queries (1–3 nodes). For training set generation, we sample 20 queries per community on disjoint datasets, 30 random queries on Facebook datasets, and use 5% of nodes as queries on MAG datasets. For each training query, following COCLEP [37], we have 3 positive nodes and 3 negative nodes randomly sampled from ground-truth communities as weak supervision signals. We set 100 test queries per dataset. For the CD task, following the settings of CCGC [76] for disjoint datasets and NOCD [50] for overlapping datasets, UniCom is trained in an unsupervised manner, with ground-truth labels used solely for evaluation.

6.1.3 Baselines. To comprehensively evaluate the performance of UniCom, we adopt different baseline methods for each task. For the **CS** task, we compare UniCom against 4 **algorithm-based methods**: CST [14], CTC [33], OCS [13], and MkECS [1], and 4 state-of-the-art **GNN-based baselines**: ICS-GNN [23], QD-GNN [34], COCLEP [37], and SMN [52]. By adapting the task output modules, we further include 3 **graph foundation models (GFMs)**: GCOPE [88], SAMGPT [84], and MDGFM [64].

The evaluation of the **disjoint CD** task includes 2 **traditional methods**: AutoEncoder (AE) [73] and DeepWalk [48], as well as 4 **GNN-based models**: CCGC [76], UCoDe [44], DyFSS [89], and FPGC [72]. In addition, for the **overlapping CD** task, we adopted 2 **traditional baselines**, W-CPM [86] and BigCLAM [75], along with 4 **GNN-based models**, which are NOCD [50], DynaResGCN [45], UCoDe [44], and SSGCAE [26]. Since the GFMs in the CS task are tailored for supervised learning, we exclude them from the CD task.

6.1.4 Evaluation Metrics. Evaluations of CS and CD are conducted with different metrics. For the CS task, we adopt 3 representative metrics: F1-score (F1), Normalized Mutual Information (NMI), and Jaccard Similarity (JAC). For the disjoint CD task, we adopt the F1 and NMI to evaluate the accuracy and community quality. Moreover, the overlapping CD is evaluated with Overlapping Normalized Mutual Information (ONMI) due to the inherent overlapping nature of the communities, which makes traditional metrics inapplicable.

To ensure fair and robust evaluation, we run each experiment 5 times and report the average performance, which mitigates the effects of randomness in model initialization and data sampling.

6.1.5 Implementation Details. For the DAS phase, we train UniCom with a learning rate selected from the range $[0.0001, 0.01]$. The maximum number of hops for conductance augmentation is 5 by default. The loss balancing coefficient α is searched within the range $[0.01, 1.00]$. We use the pre-trained model with a 3-dimensional positional encoding on most datasets, except for Reddit, Products, CS, and ENG, which use a 10-dimensional positional encoding.

Within the UGL phase, UniCom is pre-trained for 100 epochs with early stopping. The maximum number of hops used in conductance augmentation is set to 5 by default. The loss balancing coefficient β is set to 0.1. The batch size equals the number of nodes and is set to 4000 when out-of-memory (OOM) issues arise. By default, we use a single-layer graph transformer with a 3-dimensional positional encoding on all pre-training datasets, configured with a 512-dimensional hidden size, 8 attention heads, and a dropout rate of 0.1. However, the dimensionality of positional encoding can be increased for larger datasets to enhance model capacity.

Table 4: Performance of CS on Disjoint Datasets.

Method	Cora			CiteSeer			PubMed			Photo			Reddit			Products			Avg
	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	JAC	+/-
CST	43.39	31.86	27.70	38.61	28.69	23.93	69.46	59.38	53.21	70.94	59.02	54.97	44.33	36.96	28.48	67.39	60.57	50.82	-35.13%
OCS	46.92	36.61	30.65	35.33	27.15	21.46	56.55	47.42	39.42	23.09	17.92	13.05	-	-	-	-	-	-	-51.98%
CTC	33.62	24.90	20.21	32.22	24.23	19.20	56.42	47.22	39.30	63.60	51.88	46.63	15.52	12.84	8.41	61.36	54.67	44.25	-45.87%
MkECS	44.50	32.84	28.62	38.32	28.62	23.70	69.21	59.13	52.92	71.07	59.15	55.12	41.72	34.70	26.36	57.76	51.24	40.61	-37.03%
ICS-GNN	60.33	45.98	43.31	52.88	40.19	36.00	89.61	82.44	81.22	93.52	87.02	87.85	73.87	64.75	58.56	57.74	51.23	40.61	-18.61%
QD-GNN	65.95	51.10	49.19	52.11	39.43	35.24	60.41	50.83	43.77	25.11	19.47	14.36	-	-	-	-	-	-	-42.69%
COCLEP	80.31	66.79	67.53	44.98	33.76	29.40	90.23	83.25	82.20	**94.88**	**89.18**	**90.27**	-	-	-	-	-	-	-13.88%
SMN	84.75	72.01	73.59	77.27	63.55	62.99	87.02	79.03	77.05	91.33	83.62	84.10	84.97	77.22	73.91	71.42	64.68	55.73	-6.55%
GCOPE	83.49	70.31	71.66	75.34	61.48	60.56	90.93	84.23	83.39	94.32	88.24	89.25	-	-	-	-	-	-	-5.51%
SAMGPT	80.32	66.50	67.23	77.06	63.31	62.71	91.13	84.49	83.71	93.76	87.30	88.25	-	-	-	-	-	-	-6.13%
MDGFM	75.66	61.04	60.85	71.14	56.97	55.24	80.23	72.11	69.04	93.51	86.91	87.81	-	-	-	-	-	-	-12.40%
UniCom	**93.83**	**85.73**	**88.39**	**77.71**	**64.04**	**63.57**	**94.51**	**89.51**	**89.61**	94.45	88.44	89.49	**89.72**	**83.17**	**81.35**	**76.57**	**69.98**	**62.04**	-

Table 5: Performance of CS on Overlapping Datasets.

Method	FB107			FB1684			FB1912			CS			ENG			Avg
	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	JAC	+/-
CST	54.01	42.23	37.00	73.15	59.58	57.67	55.31	42.51	38.22	38.90	28.78	24.15	33.62	23.95	20.21	-31.23%
OCS	30.35	23.04	17.89	40.71	30.60	25.56	49.49	38.09	32.88	26.99	20.42	15.60	50.60	40.02	33.87	-41.44%
CTC	36.66	27.95	22.44	57.93	44.95	40.78	50.75	38.63	34.00	32.81	24.14	19.62	29.94	21.24	17.61	-33.30%
MkECS	53.90	42.13	36.89	73.24	59.68	57.78	55.07	42.31	38.00	35.29	26.02	21.43	31.69	22.51	18.83	-32.20%
ICS-GNN	70.69	57.82	54.67	90.03	80.41	81.89	78.65	65.49	64.82	40.95	30.38	25.75	42.77	30.78	27.22	-17.03%
QD-GNN	77.44	65.03	63.20	81.05	73.26	73.82	80.16	67.24	66.89	52.54	39.78	35.67	61.73	46.48	44.64	-11.25%
COCLEP	71.30	58.46	55.42	91.54	82.73	84.42	81.42	68.80	68.71	66.24	52.06	49.53	69.88	54.25	53.70	-5.95%
SMN	70.07	57.21	53.96	83.87	71.95	72.24	76.98	63.61	62.60	72.66	58.51	57.09	76.91	61.74	62.51	-6.39%
GCOPE	71.62	58.78	55.80	90.36	80.91	82.44	78.84	65.72	65.09	65.94	51.77	49.20	69.33	53.70	53.06	-7.01%
SAMGPT	72.71	59.91	57.13	90.72	81.43	83.02	79.65	66.64	66.18	71.08	56.86	55.15	70.99	55.38	55.03	-5.05%
MDGFM	72.58	59.78	56.98	85.91	74.59	75.31	78.11	64.89	64.11	65.37	51.22	48.56	64.56	49.10	47.68	-9.26%
UniCom	**81.00**	**69.14**	**68.09**	**92.21**	**83.77**	**85.56**	**81.63**	**69.02**	**68.98**	**74.85**	**60.82**	**59.81**	**77.40**	**62.27**	**63.13**	-

Table 6: Performance of CD on Disjoint Datasets.

Method	Cora		Photo		Reddit		Avg
	F1	NMI	F1	NMI	F1	NMI	+/-
AE	47.58	28.36	27.63	20.55	16.30	27.91	-44.16%
DeepWalk	62.49	44.02	72.21	69.42	73.13	84.47	-4.59%
CCGC	58.56	52.27	65.68	64.62	-	-	-8.24%
UCoDe	62.39	56.24	47.72	50.20	-	-	-14.39%
DyFSS	67.34	54.30	73.19	**71.22**	-	-	-2.01%
FPGC	73.10	**57.76**	69.45	62.83	-	-	-2.74%
UniCom	**73.43**	56.29	**74.38**	70.00	**74.57**	**84.59**	-

Table 7: Performance of CD on Overlapping Datasets.

Method	FB107	FB1684	FB1912	CS	ENG	Avg
	ONMI	ONMI	ONMI	ONMI	ONMI	+/-
W-CPM	8.43	22.22	12.37	-	-	-19.21%
BigCLAM	14.23	33.99	33.71	-	-	-6.24%
NOCD	11.85	33.09	39.32	46.09	39.75	-3.08%
DynaResGCN	15.69	40.84	40.60	41.60	37.51	-1.85%
UCoDe	12.41	36.83	35.64	**46.16**	33.63	-4.17%
SSGCAE	15.07	37.05	25.56	31.23	30.43	-9.23%
UniCom	**17.34**	**42.21**	**41.11**	44.26	**40.59**	-

All experiments for UniCom are conducted on a server equipped with an Intel Xeon 6248R CPU (512 GB RAM) and NVIDIA RTX A5000 GPUs, providing a consistent computational setup for reproducible performance comparison across all methods.

6.2 Effectiveness Evaluation

The main effectiveness evaluation for CS and CD tasks is conducted on 11 downstream datasets, using DBLP, Computers, and Instagram for pre-training. The results are illustrated in Table 4 and Table 5 for CS, and in Table 6 and Table 7 for CD, respectively. The best results are highlighted in **bold**, while the second-best ones are underlined. Missing values denote Out-of-Memory (OOM) or runs that failed to finish training or evaluation within 7 days.

Community Search. The performance comparison of UniCom against 11 baselines is reported in Table 4 and Table 5 for disjoint and overlapping CS, respectively. The results show that UniCom outperforms all baselines on most datasets and ranks second on the rest. Moreover, the lightweight design enables UniCom to scale to large datasets without incurring OOM issues. Compared with 4 **algorithm-based methods** that overlook the pivotal role of node features, UniCom incorporates both structure- and feature-level information and outperforms CST (the best algorithm-based baseline) by 35.13% for disjoint CS and 31.23% for overlapping CS. Compared with **GNN-based baselines**, UniCom achieves an average improvement of at least 6.55% for disjoint CS, and more than 5.95% for overlapping CS. The result evidences the effectiveness of aggregating knowledge from multiple pre-training domains. Additionally, UniCom outperforms all **GFMs** on the CS task across all datasets, achieving an average improvement of over 5.51% on disjoint datasets and 5.05% on overlapping datasets, demonstrating the importance of incorporating cohesive information.

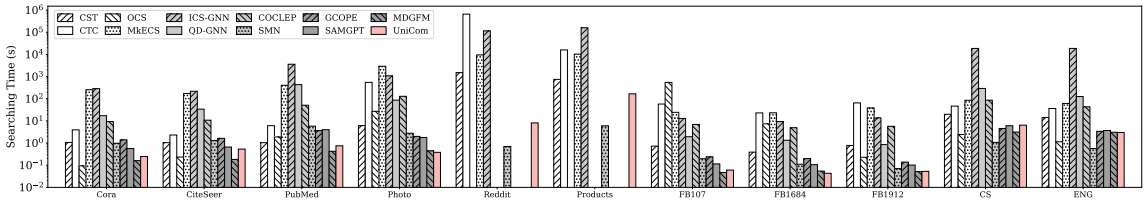
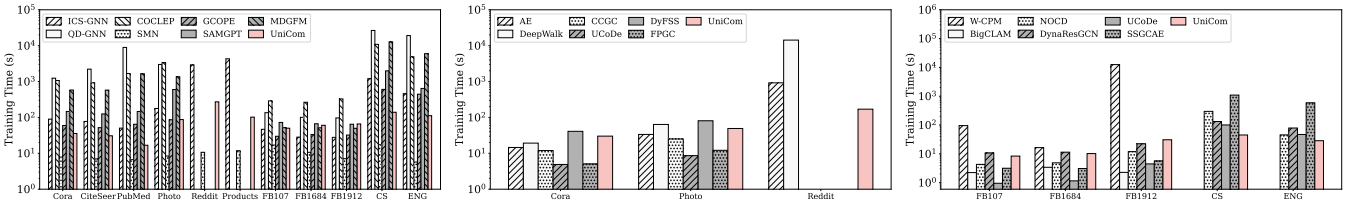


Figure 4: Total Online Searching Time of Community Search (100 Queries).



(a) CS Training Time. **(b) Disjoint CD Training Time.** **(c) Overlapping CD Training Time.**

Figure 5: Training Time of Community Search and Detection.

Community Detection. We report the disjoint CD performance of UniCom in Table 6. Across 3 benchmark datasets, UniCom consistently achieves the highest F1-score. Notably, existing state-of-the-art **GNN-based models** often encounter OOM issues when being applied to large-scale datasets such as Reddit. In contrast, UniCom is explicitly designed for scalability, enabling adaptation on large graphs without compromising performance. Table 7 demonstrates the performance of UniCom on the overlapping CD task, evaluated on 5 benchmark datasets. UniCom surpasses all baseline methods on 4 out of 5 datasets, with an average improvement ranging from 1.85% to 9.23% over all **GNN-based baselines**.

Overall, UniCom delivers consistently superior performance across all tasks compared with existing methods, which typically lack either multi-dataset knowledge or cohesive information. This demonstrates the advantages of a cohesiveness-aware unified framework with minimal task-specific designs while enabling consistent representation learning across diverse domains.

6.3 Efficiency Evaluation

Figure 4 shows the total online searching time of 100 queries for all baselines and UniCom. Traditional methods generally exhibit lower search time due to their non-learning nature. However, on graphs with high average degrees (e.g., Reddit), they suffer from a dramatic escalation in search time. Compared with GNN-based methods and GFMs, UniCom has the top-3 fastest online searching time on most datasets. Notably, UniCom achieves significantly better searching efficiency than ICS-GNN across all datasets. Moreover, it is on average 86.55× and 30.24× faster than QD-GNN and COCLEP, respectively. Compared with SMN and GFMs, UniCom also demonstrates comparable or even higher efficiency.

Figure 5a compares the CS training time of UniCom with all GNN-based baselines. Since traditional community search methods do not require offline training, we only report the offline training time of the GNN-based approaches and GFMs. UniCom may require multiple offline training runs depending on the number of pre-training datasets used. However, because these training runs can be executed in parallel on multiple GPUs, we report the longest single training run as the offline training cost of UniCom. In general, UniCom achieves substantially shorter training time than most selected baselines, including ICS-GNN, QD-GNN, COCLEP, and all GFMs, while being slightly slower than SMN.

Based on the results in Figure 5b and Figure 5c, UniCom demonstrates highly efficient training in both disjoint and overlapping community detection tasks. Notably, on the Reddit dataset, UniCom is the only GNN-based method that completes community detection without encountering OOM errors, all while keeping the training time within a practical range. Total training time for overlapping community detection supports that UniCom exhibits stable efficiency across 5 overlapping datasets, with particularly minimal training time for large MAG datasets.

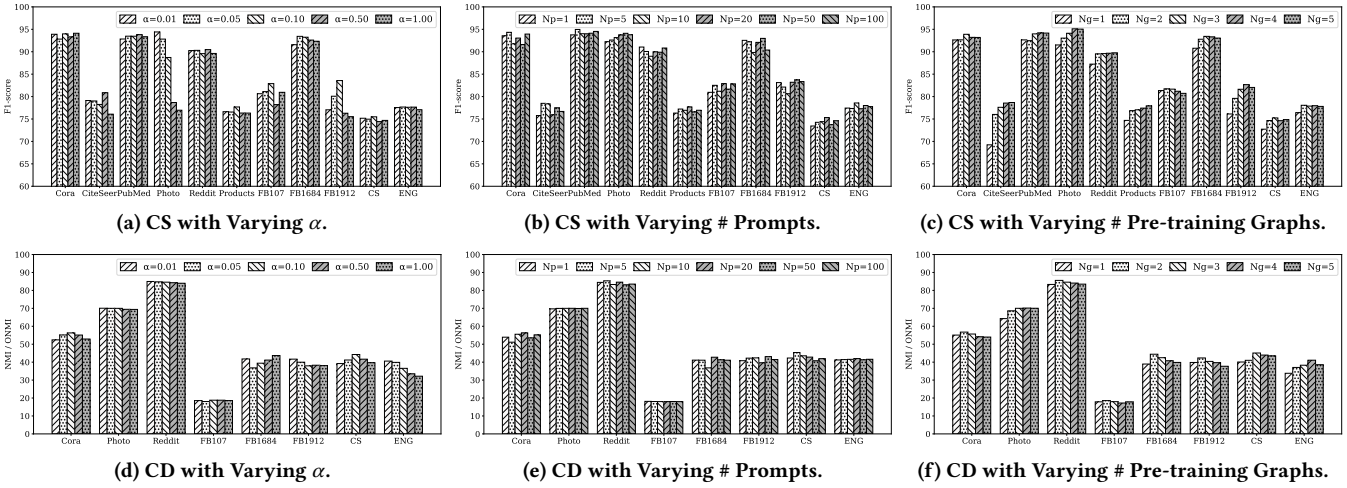
6.4 Ablation Studies

We conduct ablation studies by comparing the performance of UniCom and 4 variants on both CS and CD tasks across disjoint and overlapping datasets to evaluate the contribution of each component. Specifically, "w/o Pre-train" is a variant that replaces the *multi-domain pre-training* with task-specific fine-tuning. "w/o Fusion" is a variant that removes the *multi-domain knowledge fusion* module and is pre-trained solely on the DBLP dataset. Additionally, "w/o Coh Prompt" and "w/o Adp Prompt" indicate the removal of the *cohesive subgraph prompt* and *adaptation prompt*, respectively. The experiments are conducted on 6 datasets covering different tasks to assess the effectiveness of each component comprehensively.

The results are illustrated in Table 8. Overall, the full model consistently attains superior and robust performance across all tasks and evaluation metrics, with each component contributing complementarily. The pre-training and adaptation prompt modules yield the most significant performance gain, highlighting the crucial role of knowledge transfer in subgraph-level tasks. The effect is particularly evident in CD tasks, where transferred knowledge compensates for the absence of supervision. These observations further prove the necessity of building a unified model for CS and CD. Moreover, the results also indicate that the fusion module and cohesive subgraph prompt offer stable benefits across diverse tasks,

Table 8: Ablation Study on Community Search and Detection.

Method	PubMed-CS			CS-CS			Cora-CD		Photo-CD		FB1684-CD	FB1912-CD
	F1	NMI	JAC	F1	NMI	JAC	F1	NMI	F1	NMI	ONMI	ONMI
w/o Pre-train	92.30	86.17	85.71	71.98	57.78	56.22	65.31	50.81	57.98	43.75	28.71	37.31
w/o Fusion	92.86	87.07	86.74	73.32	59.19	57.88	72.53	55.09	71.66	63.53	40.36	40.30
w/o Coh Prompt	93.44	87.88	87.71	72.77	58.61	57.20	55.23	42.93	72.74	69.13	40.38	40.22
w/o Adp Prompt	91.81	85.46	84.86	72.86	58.70	57.31	37.59	26.66	59.97	52.99	31.96	40.75
Full Model	**94.51**	**89.51**	**89.61**	**74.85**	**60.82**	**59.81**	**73.43**	**56.29**	**74.38**	**70.00**	**42.21**	**41.11**



(a) CS with Varying α. **(b) CS with Varying # Prompts.** **(c) CS with Varying # Pre-training Graphs.**

(d) CD with Varying α. **(e) CD with Varying # Prompts.** **(f) CD with Varying # Pre-training Graphs.**

Figure 6: Hyper-parameter Analysis.

underscoring the value of cross-domain knowledge integration and the efficacy of cohesiveness guidance for community-level tasks.

6.5 Hyper-parameter Analysis

In this section, we systematically evaluate the performance of UniCom under various configurations, including the trade-off parameter α, the number of basis prompts for adaptation, and the number of pre-training graphs adopted in the UGL phase. These experiments aim to provide a comprehensive analysis of how each factor influences the model's effectiveness and to illuminate the robustness and applicability of the proposed approach.

6.5.1 Trade-off Parameter α. We investigate the impact of the trade-off parameter α during the model adaptation stage, and evaluate values of 0.01, 0.05, 0.1, 0.5, and 1, where α is used to balance the task loss and the CMMD loss. As illustrated in Figure 6a and Figure 6d, UniCom exhibits varying performance as α changes. In most cases, such as CS on the PubMed and FB1684 datasets, and CD on the Cora and CS datasets, the best results are achieved when α lies within a moderate range (0.05 to 0.5). Furthermore, the overall performance remains stable and competitive across the entire evaluated range.

6.5.2 Number of Basis Prompts for Adaptation. Figure 6b and Figure 6e present the performance of UniCom under different numbers of basis prompts (1, 5, 10, 20, 50, and 100). The results indicate that UniCom maintains strong and competitive performance across all configurations while introducing only a small number of learnable parameters. The best results are generally achieved with 5 to 50 prompts. These observations indicate that simply increasing the number of tunable parameters does not necessarily enhance performance, and that finding an appropriate balance is more important.

6.5.3 Number of Pre-training Datasets. To evaluate how the number of pre-training graphs affects the performance of UniCom, we progressively incorporate 1 to 5 pre-training datasets as listed in Table 3 and evaluate the results on all datasets and tasks. As shown in Figure 6c and Figure 6f, UniCom generally achieves the best performance when 2 to 4 pre-training datasets are utilized. As the number of pre-training datasets increases, the model performance gradually approaches a plateau, beyond which performance no longer improves. This validates the data efficiency of UniCom, indicating that using only a limited number of pre-training datasets is sufficient to achieve optimal performance.

7 CONCLUSION

In this paper, we address the long-standing gap between community search and detection by introducing UniCom, a unified framework for enhancing multi-domain knowledge transferability targeting both tasks across diverse downstream datasets. Our work leverages a Domain-aware Specialization (DAS) module, supported by Universal Graph Learning (UGL), both of which incorporate cohesiveness to capture structural and semantic information at local and global levels, while alignment is further strengthened through an adaptation prompt and a lightweight projector. UniCom demonstrates effectiveness and efficiency across multiple tasks, as validated by extensive experiments on 16 datasets and comparisons against 22 baselines. The generalizability reveals its potential to serve as a foundation model for a wide range of subgraph-level tasks.

REFERENCES

[1] Takuya Akiba, Yoichi Iwata, and Yuichi Yoshida. 2013. Linear-time enumeration of maximal K-edge-connected subgraphs in large networks by random contraction. In *22nd ACM International Conference on Information and Knowledge Management, CIKM 2013*. 909–918.

[2] Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. 2006. Local Graph Partitioning using PageRank Vectors. In *47th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2006*. 475–486.

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

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[5] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *6th International Conference on Learning Representations, ICLR 2018*.

[6] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alexander J. Smola. 2006. Integrating structured biological data by Kernel Maximum Mean Discrepancy. In *Proceedings 14th International Conference on Intelligent Systems for Molecular Biology 2006*. 49–57.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[8] Jiazun Chen, Jun Gao, and Bin Cui. 2023. ICS-GNN$^+$: lightweight interactive community search via graph neural network. *VLDB J.* 32, 2 (2023), 447–467.

[9] Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. 2023. NAGphormer: A Tokenized Graph Transformer for Node Classification in Large Graphs. In *The Eleventh International Conference on Learning Representations, ICLR 2023*.

[10] Jiazun Chen, Yikuan Xia, and Jun Gao. 2023. CommunityAF: An Example-based Community Search Method via Autoregressive Flow. *Proc. VLDB Endow.* 16, 10 (2023), 2565–2577.

[11] Jiazun Chen, Yikuan Xia, Jun Gao, Zhao Li, and Hongyang Chen. 2025. CommunityDF: A Guided Denoising Diffusion Approach for Community Search. In *41th IEEE International Conference on Data Engineering, ICDE 2025*. 460–473.

[12] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 257–266.

[13] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yiqi Lu, and Wei Wang. 2013. Online search of overlapping communities. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013*. 277–288.

[14] Wanyun Cui, Yanghua Xiao, Haixun Wang, and Wei Wang. 2014. Local search of communities in large graphs. In *International Conference on Management of Data, SIGMOD 2014*. 991–1002.

[15] Yizhou Dai, Miao Qiao, and Rong-Hua Li. 2024. On Density-based Local Community Search. *Proceedings of the ACM on Management of Data* 2, 2 (2024), 1–25.

[16] Anjali de Silva, Gang Chen, Hui Ma, Seyed Mohammad Nekooei, and Xingquan Zuo. 2025. Advancing Community Detection with Graph Convolutional Neural Networks: Bridging Topological and Attributive Cohesion. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025*. ijcai.org, 7383–7391.

[17] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference, WWW 2019*. 417–426.

[18] Shuheng Fang, Kangfei Zhao, Guanghua Li, and Jeffrey Xu Yu. 2023. Community Search: A Meta-Learning Approach. In *39th IEEE International Conference on Data Engineering, ICDE 2023*. 2358–2371.

[19] Shuheng Fang, Kangfei Zhao, Yu Rong, Zhixun Li, and Jeffrey Xu Yu. 2024. Inductive Attributed Community Search: to Learn Communities across Graphs. *Proc. VLDB Endow.* 17, 10 (2024), 2576–2589.

[20] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. 2023. Universal Prompt Tuning for Graph Neural Networks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

[21] Yixiang Fang, Xin Huang, Lu Qin, Ying Zhang, Wenjie Zhang, Reynold Cheng, and Xuemin Lin. 2020. A survey of community search over big graphs. *VLDB J.* 29, 1 (2020), 353–392.

[22] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.

[23] Jun Gao, Jiazun Chen, Zhao Li, and Ji Zhang. 2021. ICS-GNN: Lightweight Interactive Community Search via Graph Neural Network. *Proc. VLDB Endow.* 14, 6 (2021), 1006–1018.

[24] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*.

[25] John A. Hartigan and M. Anthony Wong. 1979. A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society* (1979).

[26] Chaobo He, Yulong Zheng, Junwei Cheng, Yong Tang, Guohua Chen, and Hai Liu. 2022. Semi-supervised overlapping community detection in attributed graph with graph convolutional autoencoder. *Inf. Sci.* 608 (2022), 1464–1479.

[27] Dongxiao He, Yue Song, Di Jin, Zhiyong Feng, Binbin Zhang, Zhizhi Yu, and Weixiong Zhang. 2020. Community-Centric Graph Convolutional Network for Unsupervised Community Detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Christian Bessiere (Ed.). 3515–3521.

[28] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. 2025. UniGraph: Learning a Unified Cross-Domain Foundation Model for Text-Attributed Graphs. In *SIGKDD*. ACM, 448–459.

[29] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[30] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022*. ACM, 594–604.

[31] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. GPT-GNN: Generative Pre-Training of Graph Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2020*. 1857–1867.

[32] Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024. Can GNN be Good Adapter for LLMs?. In *Proceedings of the ACM on Web Conference 2024, WWW 2024*. 893–904.

[33] Xin Huang, Laks V. S. Lakshmanan, Jeffrey Xu Yu, and Hong Cheng. 2015. Approximate Closest Community Search in Networks. *Proc. VLDB Endow.* 9, 4 (2015), 276–287.

[34] Yuli Jiang, Yu Rong, Hong Cheng, Xin Huang, Kangfei Zhao, and Junzhou Huang. 2022. Query Driven-Graph Neural Networks for Community Search: From Non-Attributed, Attributed, to Interactive Attributed. *Proc. VLDB Endow.* 15, 6 (2022), 1243–1255.

[35] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. 9694–9705.

[36] Jintang Li, Ruofan Wu, Wangbin Sun, Liang Chen, Sheng Tian, Liang Zhu, Changhua Meng, Zibin Zheng, and Weiqiang Wang. 2023. What's Behind the Mask: Understanding Masked Graph Modeling for Graph Autoencoders. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*. 1268–1279.

[37] Ling Li, Siqiang Luo, Yuhai Zhao, Caihua Shan, Zhengkui Wang, and Lu Qin. 2023. COCLEP: Contrastive Learning-based Semi-Supervised Community Search. In *39th IEEE International Conference on Data Engineering, ICDE 2023*. 2483–2495.

[38] Zhao Li, Pengrui Hui, Peng Zhang, Jiaming Huang, Biao Wang, Ling Tian, Ji Zhang, Jianliang Gao, and Xing Tang. 2021. What Happens Behind the Scene? Towards Fraud Community Detection in E-Commerce from Online to Offline. In *Companion of The Web Conference 2021*. 105–113.

[39] Chang Liu, Yuwen Yang, Yue Ding, Hongtao Lu, Wenqing Lin, Ziming Wu, and Wendong Bi. 2024. DAG: Deep Adaptive and Generative *K*-Free Community Detection on Attributed Graphs. In *SIGKDD*. ACM, 5454–5465.

[40] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024. One For All: Towards Training One Graph Model For All Classification Tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024*.

[41] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *Proceedings of the ACM Web Conference 2023, WWW 2023*. 417–428.

[42] Julian J. McAuley and Jure Leskovec. 2014. Discovering social circles in ego networks. *ACM Trans. Knowl. Discov. Data* 8, 1 (2014), 4:1–4:28.

[43] Péter Mernyei and Cătălina Cangea. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901* (2020).

[44] Atefeh Moradan, Andrew Draganov, Davide Mottin, and Ira Assent. 2023. UCoDe: unified community detection with graph convolutional networks. *Mach. Learn.* 112, 12 (2023), 5057–5080.

[45] Md. Nurul Muttakin, Md. Iqbal Hossain, and Md. Saidur Rahman. 2025. Overlapping Community Detection Using Dynamic Residual Deep GCN. In *Applied Algorithms - Second International Conference, ICAA 2025 (Lecture Notes in Computer Science)*, Vol. 15505. 105–116.

[46] Li Ni, Rui Ye, Wenjian Luo, Yiwen Zhang, Lei Zhang, and Victor S. Sheng. 2025. SLRL: Semi-Supervised Local Community Detection Based on Reinforcement Learning. In *AAAI*. 631–639.

[47] Erlin Pan and Zhao Kang. 2023. Beyond Homophily: Reconstructing Structure for Graph-agnostic Clustering. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research)*, Vol. 202. 26868–26877.

[48] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014*. 701–710.

[49] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*. 815–823.

[50] Oleksandr Shchur and Stephan Günnemann. 2019. Overlapping community detection with graph neural networks. *arXiv preprint arXiv:1909.12201* (2019).

[51] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[52] Qing Sima, Jianke Yu, Xiaoyang Wang, Wenjie Zhang, Ying Zhang, and Xuemin Lin. 2025. Deep Overlapping Community Search via Subspace Embedding. *Proc. ACM Manag. Data* 3, 1 (2025), 28:1–28:26.

[53] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cécile Paris, Surya Nepal, Di Jin, Quan Z. Sheng, and Philip S. Yu. 2024. A Comprehensive Survey on Community Detection With Deep Learning. *IEEE Trans. Neural Networks Learn. Syst.* 35, 4 (2024), 4682–4702.

[54] Longxu Sun, Xin Huang, Jiannan Wang, and Jianliang Xu. 2025. A Flexible Framework for Query-oriented Interactive Community Search. *Proc. VLDB Endow.* 18, 6 (2025), 1977–1990.

[55] Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip S. Yu. 2025. RiemannGFM: Learning a Graph Foundation Model from Riemannian Geometry. In *Proceedings of the ACM on Web Conference 2025, WWW 2025*. 1154–1165.

[56] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 2022. GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022*. 1717–1727.

[57] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2023. All in One: Multi-Task Prompting for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*. 2120–2131.

[58] Cunchao Tu, Xiangkai Zeng, Hao Wang, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun, Bo Zhang, and Leyu Lin. 2019. A Unified Framework for Community Detection and Network Representation Learning. *IEEE Trans. Knowl. Data Eng.* 31, 6 (2019), 1051–1065.

[59] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019*.

[60] Jianwei Wang, Kai Wang, Xuemin Lin, Wenjie Zhang, and Ying Zhang. 2024. Efficient Unsupervised Community Search with Pre-trained Graph Transformer. *Proc. VLDB Endow.* 17, 9 (2024), 2227–2240.

[61] Jianwei Wang, Kai Wang, Xuemin Lin, Wenjie Zhang, and Ying Zhang. 2024. Neural attributed community search at billion scale. *Proceedings of the ACM on Management of Data* 1, 4 (2024), 1–25.

[62] Qunzhong Wang, Xiangguo Sun, and Hong Cheng. 2025. Does Graph Prompt Work? A Data Operation Perspective with Theoretical Analysis. In *Proceedings of the 42th International Conference on Machine Learning, ICML 2025*.

[63] Shu Wang, Yixiang Fang, and Wensheng Luo. 2025. Searching and Detecting Structurally Similar Communities in Large Heterogeneous Information Networks. *Proceedings of the VLDB Endowment* 18, 5 (2025), 1425–1438.

[64] Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. 2025. Multi-Domain Graph Foundation Models: Robust Knowledge Transfer via Topology Alignment. In *Proceedings of the 42th International Conference on Machine Learning, ICML 2025*.

[65] Xinrui Wang, Zilong Liu, Shixin Ye, Xin Huang, Hong Gao, Xiuzhen Cheng, and Dongxiao Yu. 2025. With Anchors or Not: Fairness-Aware Truss-Based Community Search on Attributed Graphs. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE, 3971–3983.

[66] Yuxiang Wang, Xiaoxuan Gou, Xiaoliang Xu, Yuxia Geng, Xiangyu Ke, Tianxing Wu, Zhiyuan Yu, Runhuai Chen, and Xiangying Wu. 2024. Scalable Community Search over Large-scale Graphs based on Graph Transformer. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024*. ACM, 1680–1690.

[67] Yuxiang Wang, Zhangyang Peng, Xiangyu Ke, Xiaoliang Xu, Tianxing Wu, and Yuan Gao. 2025. Cohesiveness-aware Hierarchical Compressed Index for Community Search on Attributed Graphs. *Proc. ACM Manag. Data* 3, 1 (2025), 22:1–22:27.

[68] Yuxiang Wang, Shuzhan Ye, Xiaoliang Xu, Yuxia Geng, Zhenghe Zhao, Xiangyu Ke, and Tianxing Wu. 2024. Scalable community search with accuracy guarantee on attributed graphs. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2737–2750.

[69] Xixi Wu, Kaiyu Xiong, Yun Xiong, Xiaoxin He, Yao Zhang, Yizhu Jiao, and Jiawei Zhang. 2024. ProCom: A Few-shot Targeted Community Detection Algorithm. In *SIGKDD*. 3414–3424.

[70] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. In *WWW '22: The ACM Web Conference 2022*. ACM, 1070–1079.

[71] Yantuan Xian, Pu Li, Hao Peng, Zhengtao Yu, Yan Xiang, and Philip S. Yu. 2025. Community Detection in Large-Scale Complex Networks via Structural Entropy Game. In *Proceedings of the ACM on Web Conference*. 3930–3941.

[72] Xuanting Xie, Bingheng Li, Erlin Pan, Zhaochen Guo, Zhao Kang, and Wenyu Chen. 2025. One Node One Model: Featuring the Missing-Half for Graph Clustering. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2025)*. 21688–21696.

[73] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. 2017. Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017 (Proceedings of Machine Learning Research)*, Vol. 70. 3861–3870.

[74] Jaewon Yang and Jure Leskovec. 2012. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD workshop on mining data semantics*. 1–8.

[75] Jaewon Yang and Jure Leskovec. 2013. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013*. 587–596.

[76] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. 2023. Cluster-Guided Contrastive Graph Clustering Network. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023)*. 10834–10842.

[77] Xihong Yang, Cheng Tan, Yue Liu, Ke Liang, Siwei Wang, Sihang Zhou, Jun Xia, Stan Z. Li, Xinwang Liu, and En Zhu. 2023. CONVERT: Contrastive Graph Clustering with Reliable Augmentation. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023*. 319–327.

[78] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016 (JMLR Workshop and Conference Proceedings)*, Vol. 48. 40–48.

[79] Zhe-Rui Yang, Jindong Han, Chang-Dong Wang, and Hao Liu. 2025. GraphLoRA: Structure-Aware Contrastive Low-Rank Adaptation for Cross-Graph Transfer Learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025*. 1785–1796.

[80] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Badly for Graph Representation?. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*. 28877–28888.

[81] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

[82] Jianke Yu, Hanchen Wang, Xiaoyang Wang, Zhao Li, Lu Qin, Wenjie Zhang, Jian Liao, and Ying Zhang. 2023. Group-based fraud detection network on e-commerce platforms. In *SIGKDD*. 5463–5475.

[83] Jianke Yu, Hanchen Wang, Xiaoyang Wang, Zhao Li, Lu Qin, Wenjie Zhang, Jian Liao, Ying Zhang, and Bailin Yang. 2024. Temporal insights for group-based fraud detection on e-commerce platforms. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[84] Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. 2025. SAMGPT: Text-free Graph Foundation Model for Multi-domain Pre-training and Cross-domain Adaptation. In *Proceedings of the ACM on Web Conference 2025, WWW 2025*. 1142–1153.

[85] Mengting Zhang and Weihong Bi. 2025. A Local Community Detection Method Based on Folded Subgraph. *IEEE Trans. Knowl. Data Eng.* 37, 7 (2025), 3869–3880.

[86] Xingyi Zhang, Congtao Wang, Yansen Su, Linqiang Pan, and Hai-Feng Zhang. 2017. A Fast Overlapping Community Detection Algorithm Based on Weak Cliques for Large-Scale Networks. *IEEE Trans. Comput. Soc. Syst.* 4, 4 (2017), 218–230.

[87] Xingyi Zhang, Shuliang Xu, Wenqing Lin, and Sibo Wang. 2023. Constrained Social Community Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*.

[88] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. 2024. All in One and One for All: A Simple yet Effective Method towards Cross-domain Graph Pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024*. 4443–4454.

[89] Pengfei Zhu, Qian Wang, Yu Wang, Jialu Li, and Qinghua Hu. 2024. Every Node Is Different: Dynamically Fusing Self-Supervised Tasks for Attributed Graph Clustering. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2024)*. 17184–17192.

[90] Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. 2025. GraphCLIP: Enhancing Transferability in Graph Foundation Models for Text-Attributed Graphs. In *Proceedings of the ACM on Web Conference 2025, WWW 2025*. 2183–2197.

[91] Yun Zhu, Yaoke Wang, Haizhou Shi, Zhenshuo Zhang, Dian Jiao, and Siliang Tang. 2024. GraphControl: Adding Conditional Control to Universal Graph Pre-trained Models for Graph Domain Transfer Learning. In *Proceedings of the ACM on Web Conference 2024, WWW 2024*. 539–550.