

Cross-Branch Contrastive Learning Algorithm in Heterogeneous Networks

Runfeng Wang , Xingyang Zhao , Bingyu Zhang , and Zengxin Kang 

Abstract—Heterogeneous information networks (HINs) provide a flexible graph representation for relational data with different entity and relation types. Such heterogeneity enriches semantic modeling, but also increases the difficulty of representation learning. Existing heterogeneous graph neural networks have achieved promising results on HIN data; however, many of them rely on supervised or semi-supervised training and require sufficient labeled nodes. Since reliable annotations are often costly or incomplete in practical heterogeneous networks, reducing label dependence remains an important issue. To address this problem, this paper develops Cross-Branch Contrastive Learning (CBCL), a self-supervised framework for HIN representation learning. CBCL learns node embeddings through two complementary branches. The structural-view branch models heterogeneous neighborhood relations and path-induced structural dependencies, while the path-semantic branch captures meta-path-level semantic information. A cross-branch contrastive objective aligns the two representations and transfers complementary information between them. Moreover, meta-path-based relevance is used to expand positive samples, allowing semantically related nodes to participate in contrastive optimization. By integrating structural modeling, semantic fusion, and cross-branch supervision, CBCL learns discriminative node representations without extensive labeled data.

Link to graphical and video abstracts, and to code:
<https://latam.ieee9.org/index.php/transactions/article/view/10895>

Index Terms—Heterogeneous Information Network, Attention Mechanism, Graph Contrastive Learning, Network Embedding

I. INTRODUCTION

HETEROGENEOUS information networks (HINs) describe complex systems through multiple categories of nodes and relations, allowing different semantic associations to be represented within a unified graph structure [1]. Owing to their strong expressive capability, HINs have been widely used in recommendation systems, academic network analysis, social network mining, and knowledge graph research [2]–[5]. Compared with homogeneous graphs, HINs encode richer structural correlations and semantic dependencies derived from node types, relation types, and meta-path interactions, making node representation learning a key

problem in heterogeneous graph mining. In recent years, graph neural networks (GNNs) and their heterogeneous variants have achieved remarkable results on node classification, clustering, and recommendation [6], [7]. Nevertheless, most existing approaches rely on supervised or semi-supervised training paradigms that require annotated nodes [8]–[10]. In practical scenarios, label-scarce HIN problems are also relevant to Latin American applications, such as academic collaboration, public-service planning, mobility, health-resource coordination, and industrial innovation. Since these scenarios involve multiple entity and relation types with costly labels, self-supervised HIN representation learning can support sustainable-development-oriented regional data analytics.

To reduce the reliance on manual annotations, self-supervised learning has become an important research direction, and graph contrastive learning has received considerable attention [11]. Its main idea is to construct correlated and uncorrelated sample pairs from graph data so that learned representations preserve similarity between related samples while distinguishing unrelated ones [12], [13]. For HINs, recent studies have extended contrastive learning into multi-view and meta-path-aware settings. Existing methods usually generate semantic views through meta-paths, combine structural information with cross-view objectives, or perform representation alignment at node or semantic levels [14], [15]. These studies show that contrastive learning can utilize implicit supervisory information in graph structures without heavily depending on labeled data.

Although substantial progress has been achieved, current heterogeneous graph contrastive learning methods still have several limitations. First, many methods mainly use meta-paths as semantic view generators [16], [17]. Although meta-paths are useful for modeling high-order semantics, using them as the dominant source of representation learning may weaken the role of direct heterogeneous neighborhood relations and local topological information. Second, interactions among multiple views are often modeled through simple aggregation or alignment strategies, which may not sufficiently exploit the complementarity between structural information and path-level semantics [18], [19]. Third, most contrastive objectives regard embeddings of the same node under different views as positive pairs while treating other nodes as negatives [20]. This strategy may ignore semantically correlated nodes connected through meaningful meta-paths and even separate similar nodes during optimization.

To overcome these limitations, this paper presents a cross-branch contrastive learning framework for HINs, named

The associate editor coordinating the review of this manuscript and approving it for publication was Eduardo José da Luz (*Corresponding author: Runfeng Wang*).

Runfeng Wang, X. Zhao, B. Zhang, and Z. Kang are with the School of Intelligent Engineering, Shijiazhuang Posts and Telecommunications Technical College, Shijiazhuang, China, 050020 (e-mails: wang8222@sjzpc.edu.cn, xy.zhao@sjzpc.edu.cn, zhangby@sjzpc.edu.cn, and kzx@sjzpc.edu.cn).

Runfeng Wang is also with the School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, China.

CBCL. The model contains two collaborative branches for structural modeling and semantic modeling. During preprocessing, heterogeneous node features are projected into a shared latent space, and PathSim is employed to select informative meta-path neighbors. The structural-view branch incorporates direct heterogeneous neighbors and meta-path-induced multi-hop neighbors to preserve local interactions and high-order structural dependencies. Meanwhile, the path-semantic branch uses an attention mechanism and a Transformer-based semantic fusion module to capture interactions among meta-path-specific semantic representations. During contrastive training, CBCL treats representations of the same node across branches as positive pairs and further expands the positive set according to meta-path relevance. A bidirectional cross-branch contrastive objective is also introduced to strengthen collaboration between structural and semantic representations.

The major contributions of this work are summarized as follows:

- 1) A cross-branch contrastive learning framework, termed CBCL, is developed for self-supervised HIN representation learning, jointly optimizing structural dependencies and semantic information.
- 2) A collaborative dual-branch architecture is proposed to model complementary information from structural and path-semantic perspectives, thereby improving node representation quality.
- 3) A meta-path-relevance-guided positive sample expansion strategy is designed to incorporate semantically associated nodes into contrastive optimization and alleviate coarse positive-negative partitioning.
- 4) Extensive experiments on three benchmark datasets demonstrate that CBCL achieves competitive results in both node classification and clustering tasks.

II. RELATED WORK

With the development of graph neural networks (GNNs) on homogeneous graphs, many studies have extended them to heterogeneous graph scenarios, where structural heterogeneity and semantic diversity need to be jointly considered. Existing heterogeneous GNNs mainly differ in how they model type-specific interactions and semantic dependencies across meta-paths. For instance, HAN [7] introduces hierarchical attention to capture node-level and semantic-level importance, whereas MAGNN [8] further incorporates intermediate nodes along meta-path instances. Models such as GTN [21] and HGT [9] explore automatic relation composition and Transformer-based message passing, respectively, to encode complex heterogeneous structures. In general, these approaches rely on type-aware aggregation, relation-specific transformations, or semantic attention to learn node representations. However, most of them require labeled data, making self-supervised heterogeneous graph representation learning increasingly important in practical settings.

Contrastive learning provides an alternative paradigm by encouraging consistency between correlated samples

while separating unrelated ones [22]. This idea has been widely used in graph representation learning and gradually extended to heterogeneous graphs. DGI [23] learns node embeddings by maximizing the mutual information between local representations and global summaries. Based on this idea, several studies adapt contrastive learning to HINs through multi-view or meta-path-based strategies. DMGI [24] constructs multiple relation-specific views and introduces consensus regularization. HeCo [25] jointly considers meta-path-based and schema-level views to capture semantic and structural information. STENCIL [15] integrates multi-view aggregation with explicit structural encoding. GTC [26] combines GNNs and Transformers to model local and global dependencies through cross-view contrastive learning. Other methods, such as M2HGCL [27] and MiCo [28], further improve representation quality through multi-scale meta-path integration or topology-aware sampling. These methods show that contrastive objectives are effective for learning heterogeneous graph representations without extensive annotations. Nevertheless, most of them construct views mainly from meta-paths or predefined transformations, leaving the interaction between structural patterns and semantic information insufficiently explored.

Despite these advances, several challenges remain. First, existing methods often emphasize meta-path-induced high-order semantics, while direct heterogeneous neighborhood relations and local topological structures are not fully utilized. Second, multi-view interactions are usually handled by aggregation, consensus regularization, or view-level alignment, which may not fully exploit the complementarity between structural patterns and path-level semantics. Third, positive samples are commonly built from node-level correspondence or structural proximity, and may overlook semantically related nodes connected by meaningful meta-paths.

To address these issues, CBCL introduces a cross-branch framework that separates structural modeling and path-semantic modeling, and strengthens their interaction through bidirectional contrastive learning. Compared with representative multi-view and co-contrastive HIN methods such as HeCo, GTC, and STHGE, CBCL is not intended to replace their view-construction principles, but to provide a complementary design. In terms of view construction, CBCL explicitly couples a relation-aware structural branch with a path-semantic branch, rather than only aligning predefined semantic or encoder-based views. In terms of positive sample selection, CBCL further expands positives according to meta-path relevance, instead of relying only on same-node correspondence across views. In terms of branch interaction and semantic fusion, CBCL adopts bidirectional cross-branch contrastive optimization and Transformer-based fusion among meta-path-specific semantic representations. These designs clarify the novelty of CBCL in structural-semantic collaboration, positive sample construction, and semantic interaction modeling. The conceptual advantage of CBCL lies in organizing these components into a collaborative structural-semantic framework, where the two branches provide complementary information and are mutually

enhanced through bidirectional contrastive optimization.

III. PRELIMINARY

Before presenting the proposed framework, several basic notions used in this paper are introduced.

Definition 1: Heterogeneous Information Network. A heterogeneous information network (HIN) can be represented as a graph $G = (V, E)$ equipped with node-type and edge-type mapping functions. Here, V and E are the node and edge sets, while $\phi : V \rightarrow \mathcal{A}$ assigns each node to a type in $\mathcal{A}$, and $\psi : E \rightarrow \mathcal{R}$ maps each edge to a relation type in $\mathcal{R}$. The graph is regarded as heterogeneous when $|\mathcal{A}| + |\mathcal{R}| > 2$.

Unlike homogeneous graphs, HINs preserve type-aware entities and relations in a unified structure, which enables them to describe complex semantic associations in real-world relational data.

Definition 2: Meta-path. A meta-path specifies an ordered relational pattern on the schema of a HIN. It links a series of node types through corresponding relation types and is written as

$$P : A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1} \quad (1)$$

where $A_i \in \mathcal{A}$ is a node type and $R_i \in \mathcal{R}$ indicates the relation type connecting two adjacent node types. Meta-paths with the same endpoints may still describe different semantics, thereby revealing indirect associations beyond immediate edges.

Definition 3: Meta-path-based neighbors. Given a node $v \in V$ and a meta-path $\mathcal{P}$, node u is treated as a meta-path-based neighbor of v if at least one path instance following $\mathcal{P}$ starts from v and reaches u . The corresponding neighbor set is denoted as $N_{\mathcal{P}}(v)$.

Such neighbors are selected by path constraints rather than direct adjacency. Therefore, they provide path-specific semantic contexts, and the induced neighbor sets usually vary across different meta-paths.

Definition 4: Heterogeneous graph contrastive learning. Heterogeneous graph contrastive learning trains graph encoders with supervisory signals derived from heterogeneous graph structures. It builds graph views or sample pairs and optimizes their representations so that related instances remain consistent while unrelated instances are distinguishable.

For HINs, view generation and sample selection need to consider node types, relation types, and meta-path semantics, since these factors jointly determine the structural and semantic characteristics of the learned representations.

IV. THE PROPOSED METHOD

This section presents the proposed CBCL framework. CBCL mainly consists of four components: node feature transformation and neighbor selection, a structural-view branch, a path-semantic branch, and contrastive learning optimization. First, node features of different types are projected into a unified latent space, and informative meta-path-based neighbors are selected according to PathSim similarity. The structural-view branch then integrates direct heterogeneous neighbors with meta-path-induced connectivity to capture structural information from both local and high-order receptive ranges. Meanwhile, the

path-semantic branch employs attention-based aggregation and Transformer-based semantic fusion to model dependencies among different meta-paths. Finally, cross-branch contrastive learning is performed to enhance the consistency and complementarity between structural and semantic representations. In this way, CBCL explicitly preserves local structural patterns, high-order path-based connectivity, and meta-path-level semantic dependencies within a unified framework. The overall framework of CBCL is illustrated in Fig. 1.

For clarity, we summarize the main symbols used in this section. Let $\Phi = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_m\}$ denote the set of predefined meta-paths used by CBCL. For each meta-path $\mathcal{P}_t \in \Phi$, $A_{\mathcal{P}_t}$ denotes the filtered adjacency matrix obtained after PathSim-based top- K neighbor selection, and $\tilde{A}_{\mathcal{P}_t}$ denotes its normalized form. The fused meta-path adjacency used in the structural-view branch is denoted by $\tilde{A}_{\mathcal{P}}$. For each target node i , S_i^+ and S_i^- denote its positive and negative sample sets in cross-branch contrastive learning, respectively.

A. Node Feature Transformation and Neighbor Selection

In HINs, nodes of different types may have feature vectors with inconsistent dimensions and different feature spaces. Therefore, CBCL first maps heterogeneous node features into a shared latent space through type-specific transformation matrices.

For node i with type ϕ_i , its original feature vector is transformed as

$$h_i = \sigma(W_{\phi_i} x_i^\top + b_{\phi_i}), \quad (2)$$

where $x_i \in \mathbb{R}^{1 \times d_{\phi_i}}$ denotes the original feature vector of node i , and d_{ϕ_i} is the feature dimensionality of type ϕ_i . $W_{\phi_i} \in \mathbb{R}^{d \times d_{\phi_i}}$ and b_{ϕ_i} are the type-specific transformation matrix and bias vector, respectively. d is the dimensionality of the unified latent space, and $\sigma(\cdot)$ denotes the activation function. The transformed representation is used as the initial node feature for both the structural-view branch and the path-semantic branch, which ensures that different node types are compared and aggregated in the same latent space.

Meta-paths can capture multi-hop semantic contexts for a target node. However, long meta-paths may introduce many reachable nodes, increasing computational cost and bringing weakly relevant or noisy neighbors into representation learning. Therefore, CBCL filters meta-path-based neighbors before constructing path-induced views, rather than directly using all nodes reachable along a meta-path.

In CBCL, meta-paths are used as auxiliary semantic constraints rather than the only source of graph views. To this end, PathSim [29] is adopted to measure the relevance between nodes under a given meta-path. In this work, PathSim is applied to symmetric meta-paths whose starting and ending node types are the same as the target node type, which is consistent with the meta-path settings of the adopted HIN benchmarks. For two nodes of the same type, their similarity along meta-path $\mathcal{P}$ is computed as

$$s(v_i, v_j) = \frac{2 |\{p_{v_i \rightsquigarrow v_j} | p_{v_i \rightsquigarrow v_j} \in \mathcal{P}\}|}{|\{p_{v_i \rightsquigarrow v_i} | p_{v_i \rightsquigarrow v_i} \in \mathcal{P}\}| + |\{p_{v_j \rightsquigarrow v_j} | p_{v_j \rightsquigarrow v_j} \in \mathcal{P}\}|}, \quad (3)$$

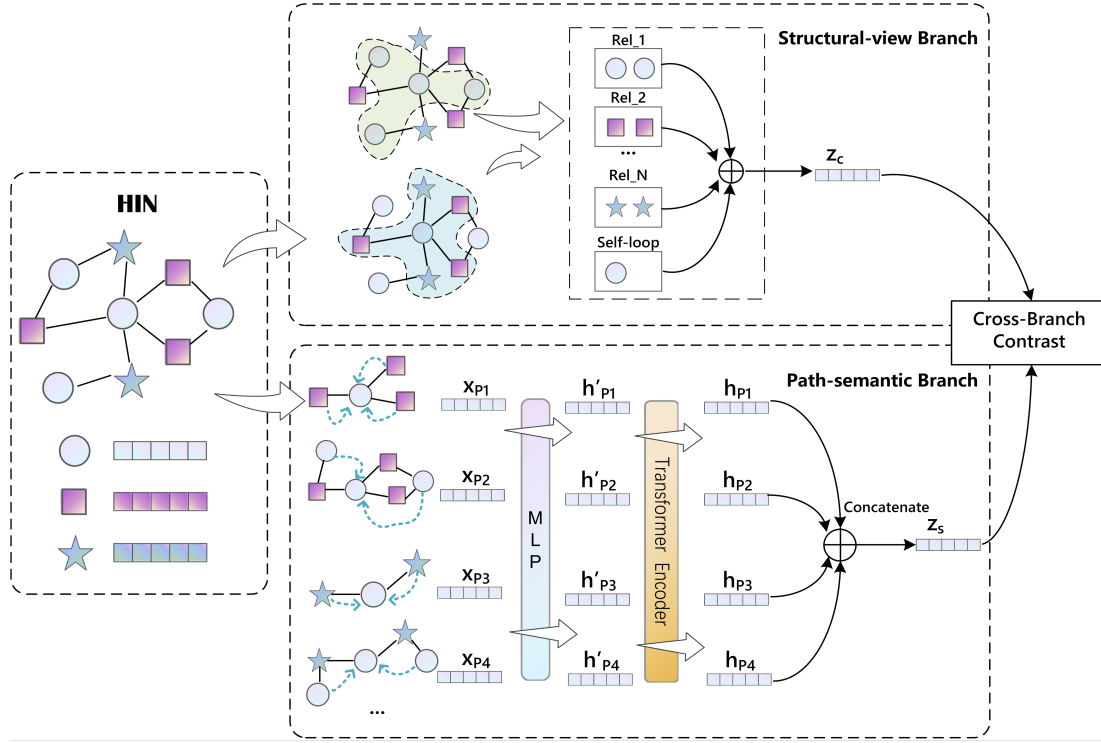


Fig. 1. Overall framework of CBCL. The structural-view branch captures heterogeneous structural information to obtain z_c , while the path-semantic branch models and fuses meta-path semantics to obtain z_s . The two representations are jointly optimized through cross-branch contrastive learning.

where $v_i \rightsquigarrow v_j$ indicates that v_i and v_j are connected by an instance of meta-path $\mathcal{P}$. For each node, CBCL retains the top- K most similar neighbors according to the PathSim scores. This strategy reduces computational overhead and weakens the influence of noisy or weakly related nodes. Only the selected neighbor pairs are retained to construct the filtered adjacency matrix induced by meta-path $\mathcal{P}$, denoted by $A_{\mathcal{P}}$.

B. Structural-view Branch

The structural-view branch aims to characterize the structural information of each target node by jointly considering direct heterogeneous relations and meta-path-induced connectivity. In this branch, the selected path-based neighbors are incorporated into relation-aware aggregation as supplementary structural information, so that both local interactions and high-order connectivity patterns can be modeled.

Given the predefined meta-path set $\Phi = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_m\}$, the neighbor-selection step produces a filtered adjacency matrix $A_{\mathcal{P}_t}$ for each meta-path $\mathcal{P}_t \in \Phi$. These adjacency matrices are normalized and fused into an overall structural adjacency matrix:

$$\tilde{A}_{\mathcal{P}} = \frac{1}{m} \sum_{t=1}^m g_{A_{\mathcal{P}_t}} \tilde{A}_{\mathcal{P}_t}, \quad (4)$$

where $g_{A_{\mathcal{P}_t}}$ is the coefficient associated with the t -th meta-path adjacency matrix, and $\tilde{A}_{\mathcal{P}_t} = D_{\mathcal{P}_t}^{-1/2} A_{\mathcal{P}_t} D_{\mathcal{P}_t}^{-1/2}$. Here, $D_{\mathcal{P}_t}$ is the degree matrix of $A_{\mathcal{P}_t}$, whose diagonal entries are computed by $D_{\mathcal{P}_t, kk} = \sum_j A_{\mathcal{P}_t, kj}$.

To model different node and relation types, CBCL adopts a parallel graph convolution-based aggregation scheme inspired by RGCN [30]. The structural representation is updated as

$$h_i^{(l+1)} = \text{ReLU} \left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + \sum_{j \in N_i^{\tilde{A}_{\mathcal{P}}}} \frac{1}{c_{i,\tilde{A}_{\mathcal{P}}}} W_{\tilde{A}_{\mathcal{P}}}^{(l)} h_j^{(l)} + b^{(l)} \right), \quad (5)$$

and the final output of this branch is

$$z_i^c = h_i^{(L)}. \quad (6)$$

Here, $\mathcal{R}$ denotes the set of relation types, N_i^r is the neighbor set of node i under relation r , and $W_r^{(l)}$ is the relation-specific weight matrix at layer l . $N_i^{\tilde{A}_{\mathcal{P}}}$ denotes the neighbor set determined by the fused structural adjacency matrix, and $W_{\tilde{A}_{\mathcal{P}}}^{(l)}$ is its corresponding weight matrix. $c_{i,r}$ and $c_{i,\tilde{A}_{\mathcal{P}}}$ are normalization factors, and L denotes the number of graph convolution layers. By adding the fused meta-path adjacency as a parallel aggregation branch, CBCL separately models direct heterogeneous relations and path-based structural dependencies. The relation-specific branch preserves original heterogeneous topology, whereas the fused meta-path branch captures selected high-order connectivity after similarity-based filtering. This design improves the expressiveness of structural representations without mixing all structural signals into a single adjacency matrix.

C. Path-semantic Branch

The path-semantic branch is designed to capture fine-grained semantics associated with different meta-paths and their interactions. Different from the structural-view branch, which emphasizes relation-aware connectivity, this branch focuses on the semantic representations generated under different meta-paths. Inspired by SeHGNN [31], it combines path-wise feature projection with a Transformer-based semantic fusion module [32].

Given the meta-path set $\Phi = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_m\}$, a GAT layer is first applied to each meta-path-induced graph to generate m groups of node embeddings with neighbor attention information, denoted by $\{H_{\mathcal{P}_1}, H_{\mathcal{P}_2}, \dots, H_{\mathcal{P}_m}\}$. Then, each group of embeddings is projected by a meta-path-specific MLP to reduce distribution gaps among different semantic views and enhance nonlinear representation capacity:

$$H'_{\mathcal{P}} = \text{MLP}_{\mathcal{P}}(H_{\mathcal{P}}), \quad (7)$$

where $H'_{\mathcal{P}}$ denotes the transformed semantic representation under meta-path $\mathcal{P}$. The MLP contains normalization, nonlinear activation, and dropout operations to stabilize feature distributions and alleviate overfitting.

Next, a Transformer-based semantic fusion module is used to model interactions among projected meta-path representations, allowing each meta-path representation to attend to other semantic contexts. For clarity, the formulation below uses a single attention head, while the implementation can be extended to multi-head attention. The overall process is illustrated in Fig. 2.

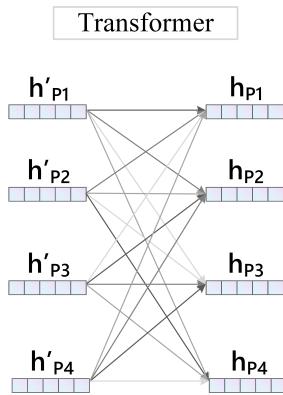


Fig. 2. Illustration of the Transformer-based semantic fusion module.

For node i , let $h'_{i, \mathcal{P}_x}$ denote its transformed representation under meta-path $\mathcal{P}_x$. The corresponding query, key, and value vectors are obtained by

$$q_{i, \mathcal{P}_x} = W_Q h'_{i, \mathcal{P}_x}, \quad (8)$$

$$k_{i, \mathcal{P}_x} = W_K h'_{i, \mathcal{P}_x}, \quad (9)$$

$$v_{i, \mathcal{P}_x} = W_V h'_{i, \mathcal{P}_x}, \quad (10)$$

where W_Q , W_K , and W_V are trainable projection matrices. The attention weight from meta-path $\mathcal{P}_x$ to $\mathcal{P}_y$ is computed as

$$\alpha_i(\mathcal{P}_x, \mathcal{P}_y) = \frac{\exp(q_{i, \mathcal{P}_x} \cdot k_{i, \mathcal{P}_y}^\top)}{\sum_{\mathcal{P}_t \in \Phi} \exp(q_{i, \mathcal{P}_x} \cdot k_{i, \mathcal{P}_t}^\top)}, \quad (11)$$

which reflects the semantic contribution of meta-path $\mathcal{P}_y$ to $\mathcal{P}_x$ for node i .

The fused representation under meta-path $\mathcal{P}_x$ is obtained by aggregating value vectors according to the attention weights and adding a residual connection:

$$h_{i, \mathcal{P}_x} = \beta \sum_{\mathcal{P}_y \in \Phi} \alpha_i(\mathcal{P}_x, \mathcal{P}_y) v_{i, \mathcal{P}_y} + h'_{i, \mathcal{P}_x}, \quad (12)$$

where β is a scaling coefficient. Finally, the representations under all meta-paths are concatenated to obtain the path-semantic representation of node i :

$$z_i^s = [h_{i, \mathcal{P}_1} \parallel h_{i, \mathcal{P}_2} \parallel \dots \parallel h_{i, \mathcal{P}_m}]. \quad (13)$$

D. Contrastive Learning Optimization

After obtaining the structural-view representation z_i^c and the path-semantic representation z_i^s , CBCL constructs positive and negative samples for cross-branch contrastive learning. A common strategy is to treat the same node under two views as a positive pair and regard all other nodes as negatives. However, this ignores semantically related nodes in HINs. Therefore, CBCL expands positive samples by considering meta-path-induced relevance, so that nodes sharing meaningful path-based semantics can also contribute to contrastive optimization.

Specifically, the relevance between nodes i and j is measured by aggregating their connectivity over all filtered meta-path adjacency matrices:

$$\rho_i(j) = \sum_{\mathcal{P}_t \in \Phi} A_{\mathcal{P}_t}(i, j), \quad (14)$$

where $\rho_i(j)$ denotes the meta-path-based relevance score. Here, $A_{\mathcal{P}_t}(i, j) = 1$ indicates that nodes i and j are connected in the filtered adjacency matrix induced by meta-path $\mathcal{P}_t$, and $A_{\mathcal{P}_t}(i, j) = 0$ otherwise. Based on this score, the positive and negative sample sets of node i are defined as $\mathcal{S}_i^+ = \{i\} \cup \{j \in V \mid \rho_i(j) > \delta\}$ and $\mathcal{S}_i^- = V \setminus \mathcal{S}_i^+$, respectively, where δ is a predefined threshold. It should be noted that the construction of $\mathcal{S}_i^+$ does not use node labels. The positive samples are determined only by filtered meta-path-induced connectivity. Therefore, the training process remains self-supervised. However, in benchmark HIN datasets, meta-path-based relevance may be correlated with class labels because labels often reflect research topics or semantic communities. Thus, δ is introduced to control the expansion strength of the positive set and avoid overly broad semantic alignment.

Since the structural representation and the concatenated path-semantic representation may have different dimensionalities, we employ two lightweight projection heads before contrastive learning: $\bar{z}_i^c = f_c(z_i^c)$, $\bar{z}_i^s = f_s(z_i^s)$,

where $f_c(\cdot)$ and $f_s(\cdot)$ are MLP-based projection functions that map the two representations into the same latent space. The projected representations are used only for contrastive optimization.

Taking $\bar{z}_i^c$ as the anchor, the contrastive loss from the structural-view branch to the path-semantic branch is defined as

$$\mathcal{L}_i^{c \rightarrow s} = -\log \frac{\sum_{j \in S_i^+} \exp(\text{sim}(\bar{z}_i^c, \bar{z}_j^s)/\tau)}{\sum_{k \in (S_i^+ \cup S_i^-)} \exp(\text{sim}(\bar{z}_i^c, \bar{z}_k^s)/\tau)}, \quad (15)$$

where τ is the temperature parameter and $\text{sim}(\cdot, \cdot)$ denotes cosine similarity. Similarly, taking $\bar{z}_i^s$ as the anchor, the reverse contrastive loss is defined as

$$\mathcal{L}_i^{s \rightarrow c} = -\log \frac{\sum_{j \in S_i^+} \exp(\text{sim}(\bar{z}_i^s, \bar{z}_j^c)/\tau)}{\sum_{k \in (S_i^+ \cup S_i^-)} \exp(\text{sim}(\bar{z}_i^s, \bar{z}_k^c)/\tau)}. \quad (16)$$

The final objective is

$$\mathcal{L} = \frac{1}{|V|} \sum_{i \in V} [\lambda \mathcal{L}_i^{c \rightarrow s} + (1 - \lambda) \mathcal{L}_i^{s \rightarrow c}], \quad (17)$$

where λ balances the two contrastive directions. Through bidirectional optimization, the structural-view branch and the path-semantic branch transfer complementary information to each other. During inference, following the strategy adopted in GTC [26], CBCL uses z_i^s as the final node representation, since it captures meta-path-level semantic information and is further enhanced by structural knowledge through cross-branch contrastive learning.

V. EXPERIMENTS AND ANALYSIS

The experimental study is organized around the following three questions:

- **RQ1:** How competitive is CBCL on node classification when compared with representative baselines?
- **RQ2:** Are the embeddings learned by CBCL suitable for unsupervised node clustering?
- **RQ3:** What roles do the key components of CBCL play in the final performance?

A. Experimental Settings

1) *Datasets:* Three widely used real-world HIN benchmark datasets are selected for evaluation, including ACM, DBLP, and AMiner. Consistent with existing heterogeneous graph representation learning studies [25], these datasets involve different node types, relation types, and manually defined meta-paths.

For ACM, papers are regarded as the target nodes and are categorized into three research topics: database, wireless communication, and data mining. For DBLP, authors serve as the target nodes, and their labels correspond to four research fields, namely database, data mining, artificial intelligence, and information retrieval. For AMiner, papers are used as target nodes and are assigned to four categories. On average, each paper in AMiner is associated with 2.74 authors and 8.96 references. The meta-paths used in this paper follow common benchmark settings for comparability, while different meta-path selections may affect both path-semantic representations and meta-path-relevance-based

positive samples. The detailed statistics of the datasets are listed in Table I, and their heterogeneous network schemas are illustrated in Fig. 3. These datasets are used to evaluate the representation-learning ability of CBCL under standard heterogeneous graph settings. The same label-efficient learning principle can be transferred to regional HIN scenarios in Latin America, where annotations are often limited and multi-source relational data are common.

TABLE I

STATISTICS OF THE BENCHMARK HIN DATASETS USED IN THE EXPERIMENTS, INCLUDING NODE TYPES, RELATION TYPES, AND PREDEFINED META-PATHS

Dataset	Nodes	Relations	Meta-paths
ACM	paper (P): 4019	P-A: 13407	PAP
	author (A): 7167	P-S: 4019	PSP
	subject (S): 60		
DBLP	author (A): 4057	P-A: 19645	APA
	paper (P): 14328	P-C: 14328	APCPA
	conference (C): 20	P-T: 85810	APTPA
	term (T): 7723		
AMiner	paper (P): 6564	P-A: 18007	PAP
	author (A): 13329	P-R: 58831	PRP
	reference (R): 35890		

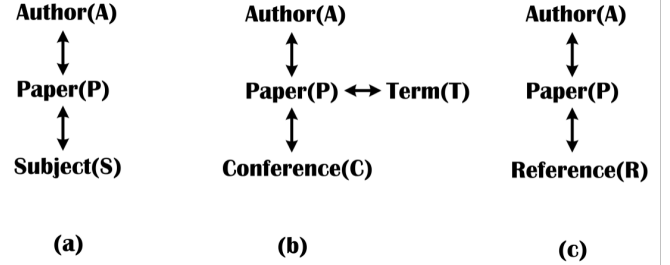


Fig. 3. Heterogeneous network schemas of the benchmark datasets: (a) ACM with paper, author, and subject nodes; (b) DBLP with author, paper, conference, and term nodes; (c) AMiner with paper, author, and reference nodes.

2) *Experimental Methods and Metrics:* The quality of the learned node representations is examined through two downstream tasks: node classification and node clustering. In all datasets, evaluation is performed only on the nodes belonging to the target type.

For node classification, Macro-F1, Micro-F1, and AUC are used as evaluation criteria. Macro-F1 treats each class equally and reflects the average classification ability across different categories. Micro-F1 aggregates predictions over all test samples and measures the overall classification performance. AUC further evaluates whether the learned embeddings can distinguish nodes from different classes under different decision thresholds.

For node clustering, K-means is applied to the learned node embeddings. The clustering quality is measured by Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). NMI evaluates the information overlap between predicted clusters and ground-truth labels, while ARI measures

clustering consistency after correcting the effect of random assignments.

During model training, CBCL is optimized with Adam. The embedding dimension is fixed to 64, and the learning rate is selected from $[10^{-4}, 10^{-3}]$. The temperature parameter τ and the loss trade-off parameter λ are searched within $\{0.1, 0.2, \dots, 0.9\}$. For each meta-path, the neighbor number K is set according to the average size of the corresponding meta-path-based neighbor set. A smaller K may remove useful high-order semantic neighbors, while a larger K may introduce noisy or weakly related nodes and increase computational cost. For the meta-path-relevance-based positive expansion, the threshold δ controls the size of the expanded positive set. A smaller δ includes more meta-path-related nodes as positives, while a larger δ makes the positive set closer to the same-node-only contrastive strategy. Thus, K and δ jointly control the trade-off between semantic coverage and noise, and neither is determined by node labels.

For node classification, the baseline results in Table II are collected from published repeated-run evaluations under the corresponding benchmark settings when such results are available. The CBCL results are reported as the mean $\pm$ standard deviation over five independent runs using the same data splits and evaluation protocol. Since repeated-run statistics for STHGE under the corresponding settings are unavailable, its originally reported point estimates are retained and marked with $\dagger$. The clustering results in Table III are reported as point estimates under the adopted benchmark protocol. Because some baseline statistics in Table II are collected from previously published studies and the per-run results under identical random seeds are unavailable, the comparisons are interpreted descriptively rather than as formal cross-paper statistical significance tests.

3) *Baselines*: CBCL is compared with a group of representative graph representation learning methods, including semi-supervised, unsupervised, and self-supervised approaches. The compared baselines include both classical HIN representation learning methods and more recent heterogeneous graph contrastive learning methods. In addition to HAN, DMGI, and HeCo, we include STHGE and GTC, which are closely related to meta-path-aware semantic modeling and cross-view contrastive learning.

GraphSAGE [33] generates node embeddings through sampled neighborhood aggregation, where information from local neighbors is collected and combined to update node representations.

DGI [23] is a self-supervised graph representation model that optimizes the agreement between node-level embeddings and a graph-level summary through mutual information maximization.

HAN [7] is a heterogeneous graph neural network based on hierarchical attention, which assigns importance weights at both the node level and the semantic level over different meta-paths.

metapath2vec [34] learns heterogeneous network embeddings by conducting random walks constrained by predefined meta-paths and optimizing the generated node sequences with a skip-gram objective.

DMGI [24] extends deep graph infomax to heterogeneous settings by learning from multiple relation-specific views and introducing a consensus-based regularization term.

HeCo [25] constructs two heterogeneous graph views, namely the network-schema view and the meta-path view, and performs contrastive learning between them to integrate local structural information and high-order semantics.

STHGE [35] considers both independence and correlation among different meta-path representation spaces, so as to describe semantic relationships from multiple path-based perspectives.

GTC [26] combines GNN-based and Transformer-based encoders to obtain complementary graph views, and aligns the two branches through a cross-view contrastive objective.

B. Node Classification

The node classification results on the three datasets are shown in Table II. The learned node embeddings are evaluated by a linear classifier under different training settings. Macro-F1, Micro-F1, and AUC are used as evaluation metrics. As shown in Table II, CBCL achieves favorable node classification performance on the three benchmark datasets. On ACM, CBCL obtains the highest mean values in eight of the nine evaluation settings, including all Micro-F1 and AUC settings. The relatively small standard deviations of CBCL also indicate stable performance across independent runs. These results are consistent with the intended complementary design of the structural-view and path-semantic branches: the former captures relation-aware neighborhood information, while the latter models high-order semantic dependencies associated with different meta-paths.

On DBLP, CBCL also remains competitive with the strongest baselines. Under the 60-label training setting, CBCL obtains the highest mean Macro-F1, Micro-F1, and AUC values. Under the 20- and 40-label settings, its mean performance is close to that of STHGE and GTC. This result suggests that CBCL can effectively integrate the heterogeneous relations among authors, papers, conferences, and terms.

On AMiner, CBCL achieves the highest mean AUC values under the 20- and 40-label settings and remains nearly tied with the best result under the 60-label setting. For Macro-F1 and Micro-F1, CBCL remains competitive, although HeCo and GTC obtain higher mean values in several settings. This may be related to the more complex citation structure of AMiner, where classification performance is more sensitive to variations in the learned representations.

Overall, CBCL consistently outperforms several classical baselines and remains competitive with recent heterogeneous graph contrastive learning methods, including HeCo, STHGE, and GTC. Since some baseline statistics are taken from previously published studies and the repeated-run statistics of STHGE are unavailable, these results are interpreted as evidence of overall competitiveness and stability rather than formal cross-paper statistical significance.

C. Node Clustering

K-means clustering is performed to further evaluate the quality of the learned node embeddings. NMI and ARI are

TABLE II

NODE CLASSIFICATION RESULTS (%) ON THE THREE BENCHMARK DATASETS USING MACRO-F1, MICRO-F1, AND AUC UNDER DIFFERENT TRAINING-LABEL SETTINGS

Dataset	Metric	Train	Graph SAGE	Mp2vec	HAN	DGI	DMGI	HeCo	STHGE†	GTC	CBCL
ACM	Ma-F1	20	47.13 ± 4.7	51.91 ± 0.9	85.66 ± 2.1	79.27 ± 3.8	87.86 ± 0.2	88.56 ± 0.8	90.47	90.20 ± 0.7	91.72 ± 0.6
		40	55.96 ± 6.8	62.41 ± 0.6	87.47 ± 1.1	80.23 ± 3.3	86.23 ± 0.8	87.61 ± 0.5	91.51	88.92 ± 0.6	91.21 ± 0.7
		60	56.59 ± 5.7	61.13 ± 0.4	88.41 ± 1.1	80.03 ± 3.3	87.97 ± 0.4	89.04 ± 0.5	92.04	89.91 ± 0.2	92.46 ± 0.5
	Mi-F1	20	49.72 ± 5.5	53.13 ± 0.9	85.11 ± 2.2	79.63 ± 3.5	87.60 ± 0.8	88.13 ± 0.8	90.40	90.64 ± 0.2	91.65 ± 0.4
		40	60.98 ± 3.5	64.43 ± 0.6	87.21 ± 1.2	80.41 ± 3.0	86.02 ± 0.9	87.45 ± 0.5	90.37	88.55 ± 0.3	90.44 ± 0.3
		60	60.72 ± 4.3	62.72 ± 0.3	88.10 ± 1.2	80.15 ± 3.2	87.82 ± 0.5	88.71 ± 0.5	91.04	89.45 ± 0.4	91.43 ± 0.3
	AUC	20	65.88 ± 3.7	71.66 ± 0.7	93.47 ± 1.5	91.47 ± 2.3	96.72 ± 0.3	96.49 ± 0.3	97.61	97.58 ± 0.1	98.01 ± 0.2
		40	71.06 ± 5.2	80.48 ± 0.4	94.84 ± 0.9	91.52 ± 2.3	96.35 ± 0.3	96.40 ± 0.4	97.68	97.54 ± 0.2	97.79 ± 0.2
		60	70.45 ± 6.2	79.33 ± 0.4	94.68 ± 1.4	91.41 ± 1.9	96.79 ± 0.2	96.55 ± 0.3	97.80	97.82 ± 0.1	98.26 ± 0.2
DBLP	Ma-F1	20	71.97 ± 8.4	88.98 ± 0.2	89.31 ± 0.9	87.93 ± 2.4	89.94 ± 0.4	91.28 ± 0.2	92.92	93.12 ± 0.3	93.05 ± 0.3
		40	73.69 ± 8.4	88.68 ± 0.2	88.87 ± 1.0	88.62 ± 0.6	89.25 ± 0.4	90.34 ± 0.3	93.10	91.62 ± 0.3	93.08 ± 0.3
		60	73.86 ± 8.1	90.25 ± 0.1	89.20 ± 0.8	89.19 ± 0.9	89.46 ± 0.6	90.64 ± 0.3	93.87	92.95 ± 0.2	93.90 ± 0.2
	Mi-F1	20	71.44 ± 8.7	89.67 ± 0.1	90.16 ± 0.9	88.72 ± 2.6	90.78 ± 0.3	91.97 ± 0.2	93.85	93.67 ± 0.3	93.68 ± 0.3
		40	73.61 ± 8.6	89.14 ± 0.2	89.47 ± 0.7	89.22 ± 0.5	89.92 ± 0.4	90.76 ± 0.3	93.89	92.02 ± 0.3	93.41 ± 0.2
		60	74.05 ± 8.3	91.17 ± 0.1	90.34 ± 0.8	90.35 ± 0.8	90.66 ± 0.5	91.59 ± 0.2	94.68	93.61 ± 0.2	94.97 ± 0.3
	AUC	20	90.59 ± 4.3	97.69 ± 0.0	98.07 ± 0.6	96.99 ± 1.4	97.75 ± 0.3	98.32 ± 0.1	98.71	98.96 ± 0.1	98.97 ± 0.1
		40	91.42 ± 4.0	97.08 ± 0.0	97.48 ± 0.6	97.12 ± 0.4	97.23 ± 0.2	98.06 ± 0.1	98.81	98.46 ± 0.1	98.73 ± 0.1
		60	91.73 ± 3.8	98.00 ± 0.0	97.96 ± 0.5	97.76 ± 0.5	97.72 ± 0.4	98.59 ± 0.1	98.99	98.89 ± 0.1	99.34 ± 0.1
AMiner	Ma-F1	20	42.46 ± 2.5	54.78 ± 0.5	56.07 ± 3.2	51.61 ± 3.2	59.50 ± 2.1	71.38 ± 1.1	70.15	70.75 ± 1.2	71.42 ± 1.5
		40	45.77 ± 1.5	64.77 ± 0.5	63.85 ± 1.5	54.72 ± 2.6	61.92 ± 2.1	73.75 ± 0.5	73.65	72.92 ± 0.7	73.64 ± 0.9
		60	44.91 ± 2.0	60.65 ± 0.3	62.02 ± 1.2	55.45 ± 2.4	61.15 ± 2.5	75.80 ± 1.8	74.00	75.03 ± 0.6	74.28 ± 1.0
	Mi-F1	20	49.68 ± 3.1	60.82 ± 0.4	68.86 ± 4.6	62.39 ± 3.9	63.93 ± 3.3	78.81 ± 1.3	75.65	78.34 ± 1.1	78.32 ± 1.6
		40	52.10 ± 2.2	69.66 ± 0.6	76.89 ± 1.6	63.87 ± 2.9	63.60 ± 2.5	80.53 ± 0.7	76.52	79.76 ± 0.8	78.21 ± 0.9
		60	51.36 ± 2.2	63.92 ± 0.5	74.73 ± 1.4	63.10 ± 3.0	62.51 ± 2.6	82.46 ± 1.4	78.12	81.85 ± 0.5	80.27 ± 0.8
	AUC	20	70.86 ± 2.5	81.22 ± 0.3	78.92 ± 2.3	75.89 ± 2.2	85.34 ± 0.9	90.82 ± 0.6	91.11	90.04 ± 0.4	91.41 ± 0.4
		40	74.44 ± 1.3	88.82 ± 0.2	80.72 ± 2.1	77.86 ± 2.1	88.02 ± 1.3	92.11 ± 0.6	92.21	91.97 ± 0.5	92.91 ± 0.6
		60	74.16 ± 1.3	85.57 ± 0.2	80.39 ± 1.5	77.21 ± 1.4	86.20 ± 1.7	92.40 ± 0.7	92.77	92.23 ± 0.6	92.76 ± 0.6

Note: Train indicates the number of labeled nodes used for classifier training. Higher values indicate better performance. Results are reported as mean ± standard deviation when repeated-run statistics are available; † denotes literature-reported point estimates.

adopted as evaluation metrics, where higher values indicate better clustering performance. Since HAN relies on labeled nodes during training, it is not included in this unsupervised comparison. The results are presented in Table III.

TABLE III

NODE CLUSTERING PERFORMANCE (%) ON ACM, DBLP, AND AMINER EVALUATED BY NMI AND ARI

Method	ACM		DBLP		AMiner	
	NMI	ARI	NMI	ARI	NMI	ARI
GraphSAGE	29.20	27.72	51.50	36.40	15.74	10.10
Mp2vec	48.43	34.65	73.55	77.70	30.80	25.26
DGI	51.73	41.16	59.23	61.85	22.06	15.93
DMGI	51.66	46.64	70.06	75.46	19.24	20.09
HeCo	56.87	56.94	74.51	80.19	32.26	28.64
STHGE	65.61	68.92	77.56	83.23	30.11	29.08
GTC	63.26	69.30	76.90	82.89	30.24	29.22
CBCL	64.18	71.54	77.51	83.20	30.28	29.25

Note: NMI and ARI denote Normalized Mutual Information and Adjusted Rand Index, respectively. Higher values indicate better clustering performance.

As shown in Table III, CBCL achieves competitive clustering performance on all three datasets. On ACM, CBCL obtains the best ARI score of 71.54 and the second-best NMI score of 64.18, indicating that the learned embeddings can form clusters that are highly consistent with the ground-truth labels. This benefits from the joint modeling of relation-aware structural information and meta-path-based semantic dependencies.

On DBLP, CBCL achieves an NMI of 77.51 and an ARI of 83.20, both of which are very close to the best results obtained by STHGE. The small performance gap suggests that CBCL can effectively capture the rich semantic relations among authors, papers, conferences, and terms. This result further indicates that the proposed cross-branch strategy remains effective on a dataset with complex heterogeneous relations.

On AMiner, CBCL obtains the best ARI score of 29.25, while its NMI score is comparable to those of STHGE and GTC but lower than those of HeCo and metapath2vec. This shows that CBCL can still preserve useful clustering consistency on AMiner, although the semantic grouping of nodes remains more challenging. Overall, the clustering results demonstrate that CBCL can learn effective unsupervised node embeddings by integrating structural dependencies and semantic correlations in heterogeneous information networks.

D. Ablation Study

To investigate the contributions of different components in CBCL, an ablation study is conducted on the node classification task. Three variants are constructed for comparison. CBCL_c removes the structural-to-semantic contrastive direction, where the structural-view representation is used as the anchor. CBCL_s removes the semantic-to-structural contrastive direction, where the path-semantic representation is used as the anchor. CBCL_nt removes the Transformer-based semantic fusion module

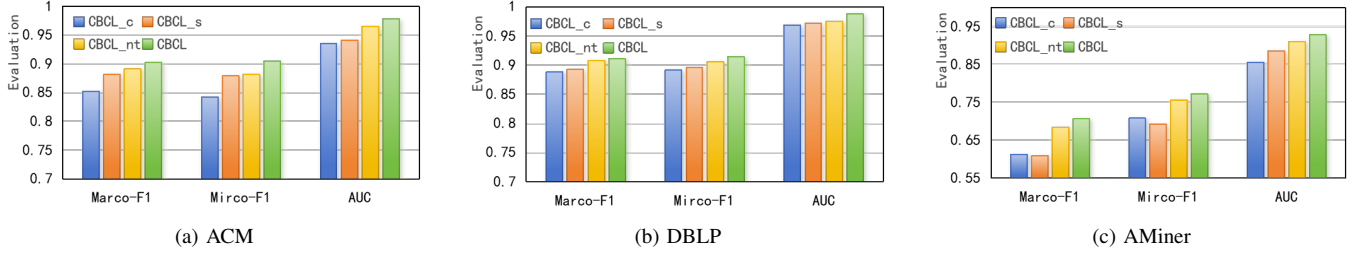


Fig. 4. Node classification ablation results of CBCL and its variants in terms of Macro-F1, Micro-F1, and AUC on (a) ACM, (b) DBLP, and (c) AMiner.

and directly concatenates meta-path-specific representations for contrastive learning. The results are shown in Fig. 4. These variants are designed to empirically isolate the main differences between CBCL and existing multi-view or co-contrastive HIN methods. Specifically, CBCL_c and CBCL_s weaken the bidirectional cross-branch contrastive interaction, which is one of the key distinctions between CBCL and methods that mainly align predefined views. CBCL_nt removes the Transformer-based semantic fusion module, thereby evaluating whether explicitly modeling interactions among meta-path semantics is beneficial beyond simple multi-view aggregation.

As shown in Fig. 4, the full CBCL model outperforms its variants on the three datasets, which verifies the effectiveness of the overall framework. Removing either branch-related contrastive component weakens the performance, indicating that the structural-view branch and the path-semantic branch provide complementary information. Their cross-branch interaction helps align heterogeneous structural features with high-order semantic information, thereby improving the discriminative ability of node embeddings.

Compared with CBCL_nt, the full CBCL model achieves better performance, demonstrating the importance of the Transformer-based semantic fusion module. This module models the interactions among different meta-path contexts and adaptively integrates semantic information from multiple views.

Overall, the ablation results show that both the cross-branch contrastive learning mechanism and the Transformer-based semantic fusion module contribute positively to CBCL. These components enable the model to jointly capture structural dependencies, semantic correlations, and complementary information between different branches, which is essential for learning effective node representations in heterogeneous information networks. Therefore, the ablation results not only support the effectiveness of individual modules, but also empirically support the main novelty of CBCL over conventional multi-view contrastive HIN frameworks.

These results also indicate that the additional components in CBCL are not redundant. Together with PathSim-based top- K filtering, the added complexity mainly improves structural-semantic interaction rather than simply increasing model size.

E. Computational and Memory Complexity

We analyze the computational and memory costs of CBCL to clarify the overhead introduced by PathSim-based neighbor filtering, multi-meta-path modeling, and Transformer-based semantic fusion. Let $n = |V|$ be the number of nodes, d be the hidden dimensionality, m be the number of meta-paths, K be the retained number of meta-path-based neighbors for each node, L be the number of graph convolution layers, and $|E_R|$ be the number of original heterogeneous relation edges.

Before model training, CBCL performs PathSim-based meta-path neighbor filtering. For each meta-path P_t , the path-induced connectivity or path-count matrix needs to be constructed according to the corresponding relation sequence. Let $\mathcal{C}_{P_t}$ denote the sparse computation cost of constructing this meta-path-induced matrix, and let M_{P_t} denote the set of nonzero node pairs reachable under P_t . Computing PathSim scores over these reachable pairs and retaining the top- K neighbors for each node requires approximately $O(|M_{P_t}| \log K)$ time when heap-based top- K selection is used. Therefore, the overall preprocessing cost over all meta-paths can be written as $O(\sum_{t=1}^m (\mathcal{C}_{P_t} + |M_{P_t}| \log K))$. In the worst case, $|M_{P_t}|$ may approach $O(n^2)$, but in practical HIN benchmarks the meta-path-induced matrices are usually sparse. Moreover, this PathSim-based filtering step is performed before training and can be reused across epochs. After filtering, each meta-path-induced adjacency matrix keeps at most nK edges, so the total number of retained meta-path edges is bounded by mnK .

In the structural-view branch, sparse message passing over original heterogeneous relations and filtered meta-path edges has an approximate per-epoch cost of $O(L(n(|\mathcal{R}| + 1)d^2 + (|E_R| + mnK)d))$. In the path-semantic branch, path-wise aggregation over the filtered meta-path graphs costs about $O(mnKd)$. The Transformer-based semantic fusion performs attention among m meta-path representations for each node, with a cost of $O(nm^2d)$. Since m is usually much smaller than n and is fixed by the predefined HIN schema, this term is manageable in typical benchmark HIN settings.

The main runtime bottleneck during full-batch training comes from the contrastive objective, which requires pairwise similarity computation between the two branches and has a cost of $O(n^2d)$. For larger HINs, mini-batch training or negative sampling can be used to reduce this cost, resulting in an approximate cost of $O(B^2d)$ for batch size B . The

memory cost is mainly determined by node embeddings, original heterogeneous edges, and filtered meta-path edges, which is approximately $O(nd + |E_R| + mnK)$ during training. If intermediate unfiltered meta-path-induced matrices are stored during preprocessing, the temporary preprocessing memory additionally depends on $\sum_{t=1}^m |M_{P_t}|$; however, these intermediate matrices can be discarded after top- K filtering.

VI. CONCLUSION

This work introduces CBCL, a self-supervised framework for representation learning in heterogeneous information networks. CBCL learns node embeddings through the collaboration of a structural-view branch and a path-semantic branch, which respectively capture heterogeneous structural dependencies and meta-path-level semantic information. Bidirectional cross-branch contrastive optimization and meta-path-relevance-based positive sample expansion further improve node representation quality.

Experiments on ACM, DBLP, and AMiner show that CBCL achieves favorable or competitive performance on node classification and clustering tasks. The ablation results further support the contributions of the cross-branch contrastive objective and the Transformer-based semantic fusion module.

CBCL may be transferable to regional label-scarce HIN scenarios, such as academic collaboration, public-service, mobility, health, and industrial networks in Latin America. However, its effectiveness on real regional data has not yet been empirically validated. Future work will investigate this direction.

REFERENCES

- [1] R. Jiang, Y. Li, H. Liu, and Y. Yu, "Incorporating attributes and multi-scale structures for heterogeneous graph contrastive learning", *Information Fusion*, vol. 122, Art. no. 103220, 2025. DOI: 10.1016/j.inffus.2025.103220
- [2] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, "Heterogeneous graph contrastive learning for recommendation", in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 544–552, 2023. DOI: 10.1145/3539597.3570484
- [3] L. Sang, Y. Wang, Y. Zhang, Y. Zhang, and X. Wu, "Intent-guided heterogeneous graph contrastive learning for recommendation", *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 4, pp. 1915–1929, 2025. DOI: 10.1109/TKDE.2025.3536096
- [4] C. Huo, D. He, Y. Li, D. Jin, J. Dang, W. Zhang, W. Pedrycz, and L. Wu, "Heterogeneous graph neural networks using self-supervised reciprocally contrastive learning", *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 1, Art. no. 24, pp. 1–21, 2025. DOI: 10.1145/3706115
- [5] H. Wang, R. Wang, C. Wen, S. Li, Y. Jia, W. Zhang, and X. Wang, "Author name disambiguation on heterogeneous information network with adversarial representation learning", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 238–245, 2020. DOI: 10.1609/aaai.v34i01.5356
- [6] D. Jin, C. Huo, C. Liang, and L. Yang, "Heterogeneous graph neural network via attribute completion", in *Proceedings of The Web Conference*, pp. 391–400, 2021. DOI: 10.1145/3442381.3449914
- [7] X. Wang, H. Ji, C. Shi, B. Wang, P. Cui, P. S. Yu, and Y. Ye, "Heterogeneous graph attention network", in *The World Wide Web Conference*, pp. 2022–2032, 2019. DOI: 10.1145/3308558.3313562
- [8] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding", in *Proceedings of The Web Conference*, pp. 2331–2341, 2020. DOI: 10.1145/3366423.3380297
- [9] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer", in *Proceedings of The Web Conference*, pp. 2704–2710, 2020. DOI: 10.1145/3366423.3380027
- [10] J. Zhao, X. Wang, C. Shi, Z. Liu, and Y. Ye, "Network schema preserving heterogeneous information network embedding", in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pp. 1366–1372, 2020. DOI: 10.24963/ijcai.2020/190
- [11] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: generative or contrastive", *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 857–876, 2023. DOI: 10.1109/TKDE.2021.3090866
- [12] Z. Wang, Q. Li, D. Yu, X. Han, X.-Z. Gao, and S. Shen, "Heterogeneous graph contrastive multi-view learning", in *Proceedings of the 2023 SIAM International Conference on Data Mining*, pp. 136–144, 2023. DOI: 10.1137/1.9781611977653.ch16
- [13] X. Chen, Y. Wang, J. Fang, Z. Meng, and S. Liang, "Heterogeneous graph contrastive learning with metapath-based augmentations", *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 1003–1014, 2024. DOI: 10.1109/TETCI.2023.3322341
- [14] Y. Mo, H. T. Shen, and X. Zhu, "Efficient self-supervised heterogeneous graph representation learning with reconstruction", *Information Fusion*, vol. 117, Art. no. 102846, 2025. DOI: 10.1016/j.inffus.2024.102846
- [15] Y. Zhu, Y. Xu, H. Cui, C. Yang, Q. Liu, and S. Wu, "Structure-enhanced heterogeneous graph contrastive learning", in *Proceedings of the 2022 SIAM International Conference on Data Mining*, pp. 82–90, 2022. DOI: 10.1137/1.9781611977172.10
- [16] J. Zhao, Q. Wen, S. Sun, Y. Ye, and C. Zhang, "Multi-view self-supervised heterogeneous graph embedding", in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 319–334, 2021. DOI: 10.1007/978-3-030-86520-7_20
- [17] J. Yu and X. Li, "Heterogeneous graph contrastive learning with meta-path contexts and weighted negative samples", in *Proceedings of the 2023 SIAM International Conference on Data Mining*, pp. 37–45, 2023. DOI: 10.1137/1.9781611977653.ch5
- [18] Y. Zhu, Y. Xu, H. Cui, C. Yang, Q. Liu, and S. Wu, "Structure-aware hard negative mining for heterogeneous graph contrastive learning", *arXiv preprint*, arXiv:2108.13886, 2021. DOI: 10.48550/arXiv.2108.13886
- [19] J. Zeng and P. Xie, "Contrastive self-supervised learning for graph classification", in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10824–10832, 2021. DOI: 10.1609/aaai.v35i12.17293
- [20] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, "Heterogeneous deep graph infomax", *arXiv preprint*, arXiv:1911.08538, 2019. DOI: 10.48550/arXiv.1911.08538
- [21] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks", in *Advances in Neural Information Processing Systems*, pp. 11983–11993, 2019. DOI: 10.48550/arXiv.1911.06455
- [22] K. Hassani and A. H. K. Ahmadi, "Contrastive multi-view representation learning on graphs", in *Proceedings of the International Conference on Machine Learning*, pp. 4116–4126, 2020. DOI: 10.48550/arXiv.2006.05582
- [23] P. Velickovic, W. Fedus, W. L. Hamilton, P. Lio, Y. Bengio, and R. D. Hjelm, "Deep graph infomax", in *Proceedings of the International Conference on Learning Representations*, pp. 1–17, 2019. DOI: 10.48550/arXiv.1809.10341
- [24] C. Park, D. Kim, J. Han, and H. Yu, "Unsupervised attributed multiplex network embedding", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 5371–5378, 2020. DOI: 10.1609/aaai.v34i04.5985
- [25] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning", in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1726–1736, 2021. DOI: 10.1145/3447548.3467415
- [26] Y. Sun, D. Zhu, Y. Wang, Y. Fu, and Z. Tian, "GTC: GNN-transformer co-contrastive learning for self-supervised heterogeneous graph representation", *Neural Networks*, vol. 181, Art. no. 106645, 2025. DOI: 10.1016/j.neunet.2024.106645
- [27] Y. Guo, Y. Xia, R. Wang, R. Duan, L. Li, and J. Li, "M2HGCL: multi-scale meta-path integrated heterogeneous graph contrastive learning", in *Proceedings of the International Conference on Advanced Data Mining and Applications*, pp. 352–367, 2023. DOI: 10.1007/978-3-031-46671-7_24
- [28] D. Sun, C. Sun, Z. Ding, C. Wang, and B. Luo, "Multi-view interactive heterogeneous graph contrastive learning", *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 1389–1405, 2026. DOI: 10.1109/TNSE.2025.3595875
- [29] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: meta path-based top-k similarity search in heterogeneous information networks",

- Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011. DOI: 10.14778/3402707.3402736
- [30] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks”, in *Proceedings of the European Semantic Web Conference*, pp. 593–607, 2018. DOI: 10.1007/978-3-319-93417-4_38
- [31] X. Yang, M. Yan, S. Pan, X. Ye, and D. Fan, “Simple and efficient heterogeneous graph neural network”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, pp. 10816–10824, 2023. DOI: 10.1609/aaai.v37i9.26283
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need”, in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017. DOI: 10.48550/arXiv.1706.03762
- [33] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs”, in *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017. DOI: 10.48550/arXiv.1706.02216
- [34] Y. Dong, N. V. Chawla, and A. Swami, “Metapath2Vec: scalable representation learning for heterogeneous networks”, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 135–144, 2017. DOI: 10.1145/3097983.3098036
- [35] Y. He, D. Yan, Y. Zhang, Q. He, and Y. Yang, “Semantic tradeoff for heterogeneous graph embedding”, *IEEE Transactions on Computational Social Systems*, vol. 10, no. 3, pp. 1263–1276, 2023. DOI: 10.1109/TCSS.2022.3163471



Runfeng Wang received his master's degree in communication and information systems from Hebei University of Science and Technology. He is currently a faculty member with the School of Intelligent Engineering, Shijiazhuang Posts and Telecommunications Technical College. His research interests include heterogeneous networks and UAV technology.



Xingyang Zhao received his master's degree in computer science from Beijing University of Posts and Telecommunications. He is currently a faculty member with the School of Intelligent Engineering, Shijiazhuang Posts and Telecommunications Technical College. His research interests include knowledge graphs and UAV technology.



Bingyu Zhang is currently the Deputy Dean of the School of Intelligent Engineering, Shijiazhuang Posts and Telecommunications Technical College. Her research interests include communication networks and UAV technology.



Zengxin Kang received the master's degree in control engineering from Northeastern University, China. He is currently a faculty member with the School of Intelligent Engineering, Shijiazhuang Posts and Telecommunications Technical College. His research interests include industrial big data processing.