ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889497

Homogeneous Graph Extraction: An Approach to Learning Heterogeneous Graph Embedding

Shihao Gao[1,4], Xiaoyan Yu[2*], Yu Cai[3], Xulong Zhang[4], Jianzong Wang[4], Taisong Jin[1]
[1]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, School of Informatics, Xiamen University, China
[2]School of Computer Science and Technology, Beijing Institute of Technology, China
[3]School of Informatics, Xiamen University, China
[4]Ping An Technology Co., Ltd., Shenzhen, China

***Abstract*—Heterogeneous Graph Neural Networks (HGNNs) aim to embed rich structural and semantic information of heterogeneous graphs into low-dimensional node representations. While HGNNs extend the foundational work of homogeneous Graph Neural Networks, the methodology for effectively transforming heterogeneous graphs into homogeneous graphs and then learning node representations remains under-explored. In this paper, we propose a novel heterogeneous graph embedding method via the Homogeneous Graph Extraction strategy, termed HGE. Specifically, the proposed method ingeniously harnesses information clusters and metapaths to extract tailored homogeneous graphs from the complex heterogeneous graph. Subsequently, these distilled homogeneous graphs are fed into a weight-shared homogeneous graph encoder to obtain embeddings with diverse semantic information. Finally, we employ an attention mechanism, which adeptly fuses embeddings derived from distinct homogeneous graphs, resulting in the more expressive capability of the nodes. The effectiveness of the proposed architecture was demonstrated through experiments on three real heterogeneous graph datasets.**

***Index Terms*—Heterogeneous Information Network, Graph Representation Learning, Attention Mechanism**

I. Introduction

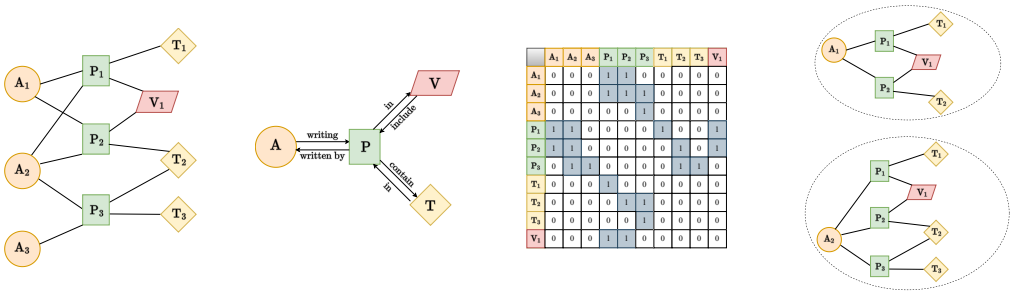
In real-world scenarios, data predominantly exists in the form of heterogeneous graphs. A heterogeneous graph [1, 2, 3] is composed of multiple types of nodes and edges, containing rich structural and semantic information. For example, as shown in Fig. 1 (a), the citation network DBLP comprises four types of nodes (authors, papers, venues, and terms) and multiple types of edges (author-write-paper, term-in-paper, paper-contain-term, etc.). Learning node representations in a heterogeneous environment poses a challenge.

To tackle the challenges posed by the heterogeneous environment, various heterogeneous graph neural networks (HGNNs) [4, 5, 6, 7, 8] have been proposed. Existing HGNNs can be primarily divided into two kinds of methods, namely metapath-based and relation-based methods. Metapath-based methods [9, 10, 11, 12, 13] leverage hand-crafted metapaths to capture the semantic relationships and structural information between nodes. Each metapath represents distinct semantic information, and the resulting embedding vectors are a fusion of multiple semantic information. However, these HGNNs typically focus on capturing sequential information while disregarding other higher-order information. Relation-based methods [14, 15, 16, 17] do not require the manual specification of metapaths. These models can aggregate messages from the local neighbors of nodes, similar to traditional GNNs, but they require the design of intricate aggregation layers to handle different types of nodes or edges.

In response to the limitations demonstrated by the existing HGNNs mentioned above, we propose a heterogeneous graph representation learning method based on Homogeneous Graph Extraction (HGE), which eliminates the need for complex aggregation layers while effectively capturing rich heterogeneous information. We only need to build homogeneous graphs for target-type nodes that are used for downstream tasks. To construct new node features, we introduce the concept of information cluster, the new features of nodes are extracted from their corresponding information clusters. The adjacency matrix, containing solely the target type nodes, is obtained by traversing the meta adjacency matrix, and different adjacency matrices can be obtained according to different metapaths. These distinct adjacency matrices are capable of capturing various sequential information. By combining the newly extracted node features and structures, multiple distinct homogeneous graphs are generated. These graphs are then fed into a homogeneous graph encoder to learn node embeddings under different structural contexts. Finally, an attention mechanism is employed to further fuse node embeddings. Our contributions are summarised as follows:

- We re-examine heterogeneous graph embedding learning from the perspective of homogeneous graph extraction and study effective conversion methods from heterogeneous graphs to homogeneous graphs.
- We propose a simple and effective new heterogeneous graph neural network that can capture both sequential information and high-order local information.

*The corresponding author. xiaoyan.yu@bit.edu.cn Supported by the National Natural Science Foundation of China (No. 62072386), Yunnan Provincial Major S&T Special Plan Project (No. 202402AD080001), Henan Key R&D Project (No. 231111212000), Open Foundation of Henan Key Lab of General Aviation Technology (No. ZHKF-230212), Key Lab of Oracle Information Processing of MOE (No. OIP2024E002), and Guangdong Key R&D Program (No. 2021B0101400003).



(a) Heterogeneous Graph (b) Meta Relation (c) Meta Adjacency Matrix (d) Information Cluster

Fig. 1. Heterogeneous information network graph composed of DBLP dataset.

- Experiments on three real-world datasets demonstrate the superiority of the proposed model.

II. Preliminaries

In this section, we give formal definitions of some key terminologies. Graphical illustrations for some of the definitions are provided in Fig. 1.

Heterogeneous Graph. A heterogeneous graph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ represents the set of nodes in the graph, and $\mathcal{E}$ denotes the set of links. $\mathcal{A}$ and $\mathcal{R}$ denote the set of node types and links types, and $|\mathcal{A}|+|\mathcal{R}| > 2$.

Metapath. A metapath defines the composite relationships among different nodes. Formally, $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ can be abbreviated as the path $A_1 A_2 \cdots A_{l+1}$, describing the intrinsic association between nodes A_1 and A_{l+1}. Different composite patterns can represent distinct semantics.

Meta Relation. Meta relations are utilized to depict the linking relationships between nodes of diverse types. Specifically, for the edge $e = (s, t)$ between node s and node t in the graph, the meta relation between the two nodes is denoted as $\langle \phi(s), \psi(e), \phi(t) \rangle$ and the inverse meta relation is denoted as $\left\langle \phi(t), \psi(e)^{-1}, \phi(s) \right\rangle$.

Meta Adjacency Matrix. The Meta Adjacency Matrix represents whether there is a corresponding meta-relationship between the nodes, where each element signifies the existence or absence of an edge between the corresponding nodes.

III. Method

In this section, we formally propose heterogeneous graph embedding learning based on homogeneous graph extraction (HGE). The architecture of HGE is shown in Fig. 2, which contains three main parts: (1) Homogeneous graph extraction, (2) Encoding homogeneous graphs, and (3) Inter-homogeneous graphs aggregation.

A. Homogeneous Graphs Extraction

To construct homogeneous graphs from a given heterogeneous graph, it is only necessary to focus on the target type nodes for downstream tasks. The new features of the nodes in the same types are obtained by the information cluster corresponding to the nodes. Seeking to capture structural information from different perspectives, we generate diverse homogeneous graph structures based on various metapaths.

1) Obtain new features: The new features for each node are derived from the associated information cluster, which encompasses nodes of all types $\{a_1, a_2, \ldots, a_M\} \in \mathcal{A}$. The information cluster is defined as follows:

Definition 1 (Information Cluster)**.** *An information cluster is defined as consisting of a central node and other types of nodes that are closest to the central node. Other types of nodes in the information cluster can provide rich local information for the central node. Fig. 1 (d) shows the information clusters corresponding to nodes A_1 and A_2.*

We first obtain the features of nodes of the same type within the information cluster through mean aggregation, as shown below:

$$h_i^{a_m} = \frac{1}{\|C_i^{a_m}\|} \sum_{j \in C_i^{a_m}} X_j, \tag{1}$$

where $h_i^{a_m}$ represents the information about type $a_m \in \mathcal{A}$ in the information cluster of node i, $C_i^{a_m}$ is all nodes of type a_m in the information cluster corresponding to node i, and X is raw feature matrix.

Since the feature vectors obtained above may have unequal dimensions, or be situated in different feature spaces. So the resulting h_i^a is projected into the same data space:

$$h_i^{'a_m} = W_{a_m} \cdot h_i^{a_m}, \tag{2}$$

where W_{a_m} is the parametric weight matrix.

Next, we aggregate different types of information within the information cluster through the attention mechanism to derive new features for the nodes. First, we summarize the importance of each type by averaging the transformed vectors of different types across all information clusters:

$$s_{a_m} = \frac{1}{|\mathcal{V}a_m|} \sum_{i \in \mathcal{V}a_m} q_1^{\mathrm{T}} \cdot \tanh\left(W_1 \cdot h_i^{'a_m} + b_1\right), \tag{3}$$

where W_1 is the weight parameter matrix, b_1 is the learnable bias vector, q_1^{T} is the parameterized attention vector. We use the softmax function to normalize the importance of different types of nodes:

$$\beta_{a_m} = \frac{\exp(s_{a_m})}{\sum_{m=1}^{M} \exp(s_{a_m})}, \tag{4}$$

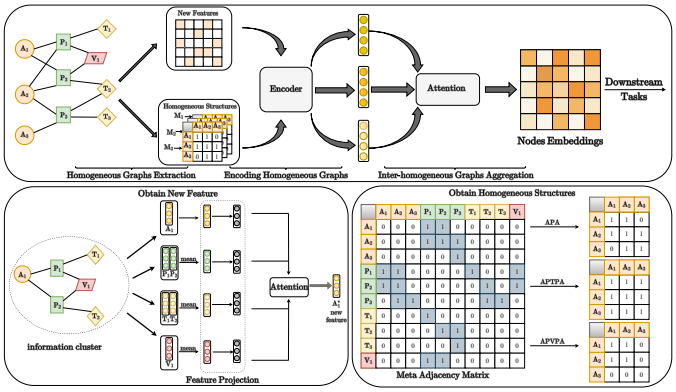


Fig. 2. The overall architecture of HGE. The example is based on DBLP dataset with node types author (A), paper (P), term (T), and venue (V). This figure exhibits the extraction of homogeneous graphs and the acquisition of node embeddings under three metapaths.

with the learned importance as weight coefficients, we can integrate these distinct embeddings of different types to derive embedding h_i that can serve as new feature for the node. The operation is shown as follows:

$$h_i = \sum_{m=1}^{M} \beta_{a_m} \cdot h_i^{'a_m}. \tag{5}$$

2) Obtaining homogeneous structures: In heterogeneous graphs, nodes of the same type often have few or even no direct edges. To construct homogeneous structures from a given heterogeneous graph, we employ a traversal method on the meta adjacency matrix. Different metapaths can be used to derive different Homogeneous adjacency matrices, which contain serialized semantic information. Formally, it can be expressed as follows:

$$\Gamma\left(P_k\right) : M \rightarrow M_k. \tag{6}$$

where M is an meta adjacency matrix, M_k is a homogeneous adjacency matrix, and Γ is a traversal function, which traverses M to generate M_k based on metapath.

B. Encoding Homogeneous Graphs

By combining the newly obtained node features with different homogeneous adjacency matrices, we create distinct homogeneous graphs. Subsequently, these graphs are fed into a homogeneous graph encoder to learn node embeddings under different structural contexts. The encoder uses the GCN [18] as the encoder, which can be formalized as

$$f\left(X, A\right) = \sigma\left(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}XW^{l}\right), \tag{7}$$

where $\hat{A} = A + I$ is the adjacency matrix with self-loops, $\hat{D} = \sum_i \hat{A}_i$ is the degree matrix, $\sigma\left(\cdot\right)$ is a nonlinear activation function, and W^l is the trainable matrix for the l-th layer. The embeddings obtained by the encoder from a homogeneous graph are represented as:

$$Z_{M_k} = f\left(H, M_k\right). \tag{8}$$

TABLE I
DATASET STATISTICS AND CHARACTERISTICS.

	Nodes	Target node (Classes)	Metapath
DBLP	Author, Paper Term, Venue	Paper 3	APA, APTPA APVPA
IMDB	Movie, Director Actor, Keyword	Author 4	MAM, MDM MKM
ACM	Author, Paper Subject, Trem	Movie 5	PP, PAP PSP, PTP

where H represents the newly extracted features matrix mentioned above, where h_i is the new feature of v_i and M_k is a homogeneous adjacency matrix.

C. Inter-homogeneous Graphs Aggregation

Given k homogeneous graphs, after feeding into the encoder, we can obtain the corresponding embedding matrices $\{Z_{M_1}, \ldots, Z_{M_K}\}$. The attention mechanism is used to integrate the embeddings from different homogeneous graphs, and we generate the final embedding vectors for each node. First, we consider each homogeneous graph by averaging the node-specific vectors produced by all target nodes $v \in \mathcal{V}_A$ passing through different homogeneous graphs:

$$s_{M_k} = \frac{1}{|\mathcal{V}_A|} \sum_{v \in \mathcal{V}_A} \tanh\left(W_2 \cdot z_v^{M_k} + b_2\right), \tag{9}$$

where W_2 and b_2 are learnable parameters. We integrate various information into the target node through the attention mechanism as follows:

$$e_{M_k} = q_2^{\mathrm{T}} \cdot s_{M_k}, \tag{10}$$

$$\beta_{M_k} = \frac{\exp\left(e_{M_k}\right)}{\sum_{k=1}^{K} \exp\left(e_{M_k}\right)}, \tag{11}$$

$$h_v = \sum_{k=1}^{K} \beta_{M_k} \cdot z_v^{M_k}, \tag{12}$$

where q_2^{T} is the parameterized attention vector.

The finally obtained node representations will be used in different downstream tasks, such as node classification and node clustering. In these tasks, the model is optimized by minimizing the cross-entropy:

$$\mathcal{L} = -\sum_{v \in \mathcal{V}_L} \sum_{c=1}^{C} y_v\left[c\right] \cdot \log \mathrm{h}_v\left[c\right]. \tag{13}$$

where $\mathcal{V}_L$ is the set of nodes involved in the computation, C is the number of categories in the dataset, y_v is the category vector of node v, and h_v is the probability vector output of node v.

IV. Experiments

A. Datasets and Baselines

The datasets required for the experiments are from DBLP, ACM, and IMDB of the HGB benchmark. We follow the setting of [13, 17] and the simple statistics of the dataset are shown in Table I.

We compare HGE with 10 baselines in three categories: homogeneous graph Neural Networks: GCN [18], GAT [19]. relation-based heterogeneous graph Neural Networks: RSHN [14], HetSANN [15], HGT [16], HGB [17]. and metapath-based heterogeneous graph Neural Networks: HAN [11], MAGNN [12], GNT [10], SeHGNN [13].

B. Evaluation Metrics

For all datasets, the same strategy is adopted: 24% for training, 6% for validation, and 70% for testing, with all edges available during training. Micro-F1 and Macro-F1 are used as evaluation metrics for the node classification task, while NMI and ARI are used for the node clustering task.

C. Node Classification

In the semi-supervised node classification task, the experimental results of our HGE model compared to the baseline are shown in Table II. From the performance comparison, we can observe that the proposed HGE outperforms all baselines in most cases. Particularly, on the DBLP and ACM datasets, our method further improves performance even when the existing baselines have already achieved fairly good performance. Moreover, The performance of simple GAT is comparable to the performance of many meticulously designed HGNNs.

TABLE II
Node classification performance comparison of different methods on three datasets.

	Macro-F1			Micro-F1		
Method	DBLP	IMDB	ACM	DBLP	IMDB	ACM
GCN	90.84	57.88	92.17	91.47	64.82	92.12
GAT	93.83	58.94	92.26	93.39	64.86	92.19
RSHN	93.34	59.85	90.50	93.81	64.22	90.32
HetSANN	78.55	49.47	90.02	80.56	57.68	89.91
HGT	93.01	63.00	91.12	93.49	67.20	91.00
HGB	94.01	63.53	93.42	94.46	67.36	93.35
HAN	91.67	57.74	90.89	92.05	64.63	90.79
MAGNN	93.28	56.49	90.88	93.76	64.67	90.77
GTN	93.52	60.47	91.31	93.97	65.14	91.20
SeHGNN	94.86	**66.63**	93.95	95.24	68.21	93.87
HGE (ours)	**94.91**	65.49	**94.37**	**95.33**	**68.57**	**94.32**

D. Node Clustering

The comparison of performance between the HGE and baseline models is shown in Table III. We can observe that in most cases, HGE outperforms all other baselines in node clustering. Furthermore, it is noticeable that all models exhibit poor performance on the IMDB dataset. This phenomenon could be attributed to the fact that the target nodes (movie nodes) in IMDB have multiple labels, but during our experiments, we only selected one of them as its corresponding true label. Additionally, combining the information from Table II, we can deduce that the node classification results and clustering results exhibit a positive correlation overall.

TABLE III
Node clustering performance comparison of different methods on three datasets.

	NMI			ARI		
Method	DBLP	IMDB	ACM	DBLP	IMDB	ACM
HGT	78.58	14.39	64.90	84.49	11.62	67.40
HGB	81.48	15.25	75.38	86.85	12.69	80.65
HAN	78.54	15.46	64.82	84.41	11.19	68.81
MAGNN	80.47	15.73	73.34	86.32	12.76	77.99
GTN	79.82	11.70	70.05	85.49	7.76	74.85
SeHGNN	83.82	16.09	77.00	88.82	**13.91**	82.34
HGE (ours)	**84.23**	**16.22**	**78.76**	**88.98**	13.09	**83.67**

E. Ablation Study

To evaluate the influence of new node features and homogeneous structures guided by different metapaths on model performance, a series of ablation experiments are conducted in this section. "w/o New F" represents a model variant that does not use the extracted new node features and only uses the original node features. Fig. 3 illustrates the results of the ablation study on the DBLP and ACM datasets.

From the Fig. 3, it can be observed that each component contributes positively to the model's performance. Notably, when only the original node features are utilized, there is a significant performance drop, indicating that the new features captured by information clusters contribute richer information to the nodes. Additionally, we observe that metapaths vary in importance and tend to be more important the shorter they are.

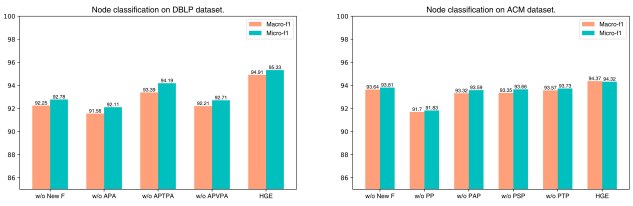


Fig. 3. Effects of the components in HGE on node classification task.

V. Conclusion

In this paper, we have proposed a novel method, termed HGE, for heterogeneous graph representation learning based on homogeneous graph extraction. HGE is designed to capture high-order local information of a heterogeneous graph through information clusters and captures sequential information through metapaths. The rich semantic information is encapsulated within different homogeneous graphs. By leveraging a homogeneous graph encoder and attention mechanism, different semantic information is effectively fused, resulting in significant performance improvement.

REFERENCES

[1] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip, "A survey of heterogeneous information network analysis," *IEEE Trans.Knowl.Data Eng.*, vol. 29, no. 1, pp. 17–37, 2016.

[2] Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and S Yu Philip, "A survey on heterogeneous graph embedding: methods, techniques, applications and sources," *IEEE Transactions on Big Data*, vol. 9, no. 2, pp. 415–436, 2022.

[3] Rui Bing, Guan Yuan, Mu Zhu, Fanrong Meng, Huifang Ma, and Shaojie Qiao, "Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8003–8042, 2023.

[4] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang, "Heterogeneous network representation learning.," in *IJCAI*, 2020, pp. 4861–4867.

[5] Ming-Yi Hong, Shih-Yen Chang, Hao-Wei Hsu, Yi-Hsiang Huang, Chih-Yu Wang, and Che Lin, "Treexgnn: can gradient-boosted decision trees help boost heterogeneous graph neural networks?," in *ICASSP*. IEEE, 2023, pp. 1–5.

[6] Costas Mavromatis and George Karypis, "Global and nodal mutual information maximization in heterogeneous graphs," in *ICASSP*. IEEE, 2023, pp. 1–5.

[7] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu, "Graph neural networks for graphs with heterophily: A survey," *arXiv preprint arXiv:2202.07082*, 2022.

[8] Mingyu Yan, Mo Zou, Xiaocheng Yang, Wenming Li, Xiaochun Ye, Dongrui Fan, and Yuan Xie, "Characterizing and understanding hgnns on gpus," *IEEE Computer Architecture Letters*, vol. 21, no. 2, pp. 69–72, 2022.

[9] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling, "Modeling relational data with graph convolutional networks," in *ESWC*. Springer, 2018, pp. 593–607.

[10] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim, "Graph transformer networks," *NeurIPS*, vol. 32, 2019.

[11] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu, "Heterogeneous graph attention network," in *WWW*, 2019, pp. 2022–2032.

[12] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *WWW*, 2020, pp. 2331–2341.

[13] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan, "Simple and efficient heterogeneous graph neural network," in *AAAI*, 2023, pp. 10816–10824.

[14] Shichao Zhu, Chuan Zhou, Shirui Pan, Xingquan Zhu, and Bin Wang, "Relation structure-aware heterogeneous graph neural network," in *ICDM*, 2019, pp. 1534–1539.

[15] Huiting Hong, Hantao Guo, Yucheng Lin, Xiaoqing Yang, Zang Li, and Jieping Ye, "An attention-based graph neural network for heterogeneous structural learning," in *AAAI*, 2020, pp. 4132–4139.

[16] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun, "Heterogeneous graph transformer," in *WWW*, 2020, pp. 2704–2710.

[17] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang, "Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks," in *SIGKDD*, 2021, pp. 1150–1160.

[18] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[19] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, "Graph attention networks," in *ICML*, 2018.