Harnessing Language Model for Cross-Heterogeneity Graph Knowledge Transfer

Jinyu Yang[1], Ruijia Wang[2], Cheng Yang[1]*, Bo Yan[1], Qimin Zhou[1], Juan Yang[1], Chuan Shi[1]*

[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]China Telecom Cloud Computing Research Institute, Beijing, China
{jinyu.yang, yangcheng, mjkbyb, zhouqimin, yangjuan, shichuan}@bupt.edu.cn, wangrj12@chinatelecom.cn

Abstract

Heterogeneous graphs (HGs) that contain various node and edge types are ubiquitous in real-world scenarios. Considering the common label sparsity problem in HGs, some researchers propose to pretrain on source HGs to extract general knowledge and then fine-tune on a target HG for knowledge transfer. However, existing methods often assume that source and target HGs share a single heterogeneity, meaning that they have the same types of nodes and edges, which contradicts the real-world scenarios requiring cross-heterogeneity transfer. Although a recent study has made some preliminary attempts in cross-heterogeneity learning, its definition of general knowledge heavily rely on human knowledge, which lacks flexibility and further leads to a suboptimal transfer. To address the problem, we propose a novel Language Model-enhanced Cross-Heterogeneity learning model, namely LMCH. Specifically, we first design a metapath-based corpus construction method to unify HG representations as languages. The corpora of source HGs are then used to fine-tune a pretrained Language Model (LM), enabling the LM to autonomously extract general knowledge across different HGs. Furthermore, to fully utilize the extensive unlabeled nodes in a few-labeled target HG, we propose an iterative training pipeline with the help of an extra Graph Neural Network (GNN) predictor, enhanced by LM-GNN contrastive alignment at the end of each iteration. Extensive experiments on four real-world datasets have demonstrated the superior performance of LMCH over state-of-the-art methods.

Code — https://github.com/BUPT-GAMMA/LMCH

1 Introduction

Heterogeneous graphs that contain a diverse range of node and edge types are ubiquitous in real-world scenarios, such as social networks (Dong et al. 2012), recommendation systems (Yan et al. 2024) and biological networks (Ma et al. 2023). Embedding the structure and attribute information of HGs into a low-dimensional vector space, Heterogeneous Graph Neural Networks (HGNNs) that typically trained in an end-to-end manner have demonstrated promising performance in plenty of graph tasks (Dong et al. 2012; Schlichtkrull et al. 2018). Recently, due to label sparsity and the high cost of data annotation, researchers have shifted to a new paradigm that initially pretrains the model on source HGs and then fine-tunes it on a target HG to handle few-shot scenarios (Zhang et al. 2022b, 2024b). However, these methods usually assume that source and target HGs share a same heterogeneity (Zhuang et al. 2021), meaning that they have the same types of nodes and edges. Such approaches frequently lead to poor outcomes in cross-heterogeneity scenarios that are closer to real-world applications.

To achieve cross-heterogeneity graph knowledge transfer, a recent research (Ding, Wang, and Liu 2023) has made some preliminary attempts. Considering the structural characteristics of HG relations, they distinguish HG relations into two categories: Affiliation Relations (ARs) with one-centered-by-another structures and Interaction Relations (IRs) with peer-to-peer structures. Subsequently, ARs and IRs are viewed as the general knowledge shared across different HGs and are used to facilitate knowledge transfer. However, the selection and definition of general knowledge heavily rely on predefined patterns from human knowledge, which lacks flexibility and harms generality. Besides, ARs and IRs are extracted based on node degree discrepancies, which solely considers local structures and ignores the long-range semantic relationships inherent in HGs, thus leading to suboptimal performance.

To address the problems, we propose a novel Language Model-enhanced Cross-Heterogeneity learning model, namely LMCH. The core idea of LMCH is to first unify HG representations as languages for automatically extracting general knowledge from source HGs, and then transfer this knowledge to the target HG. Specifically, firstly, given that metapaths can capture both local structures and long-range semantic relationships, we design a metapath-based corpus construction method to unify HG representations using natural language as shown in Figure 1 and fine-tune a language model on source HG corpora, enabling the LM to autonomously extract the general knowledge of source HGs. Secondly, to fully leverage the rich information of abundant unlabeled nodes in the target HG, we propose an iterative training pipeline with the help of an extra GNN predictor. The LM initially generates enhanced features to train the GNN, and then the GNN produces soft labels for the unlabeled nodes to fine-tune the LM. Lastly, to further narrow the performance gap between the LM and the

*Corresponding authors.

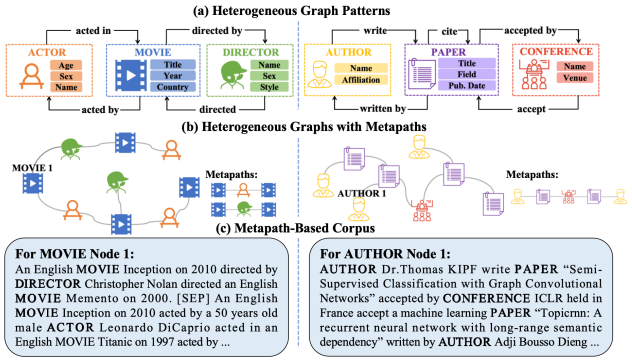

Figure 1: Heterogeneous graphs with distinct heterogeneity. (a) Heterogeneous graph patterns exhibit completely different heterogeneity across HGs. (b) Heterogeneous graphs with metapaths that contain both local structures and rich semantic relationship information. (c) Converting each node's metapaths into sequences of language tokens to standardize their representations across various HGs.

GNN, we incorporate an LM-GNN contrastive alignment approach at the end of each iteration.

Our contributions in this work are summarized as follows:

- To the best of our knowledge, we are the first to unify HG representations in metapath-based language and apply LM to autonomously extract general knowledge, which is a new paradigm for cross-heterogeneity learning and paves a new path for future research.
- We propose a novel model called LMCH, which innovatively integrates the LM and the GNN via an iterative learning process, enabling the LM to transfer source HGs' general knowledge to a target HG with abundant soft labels generated by GNN. A LM-GNN contrastive alignment method is also proposed to further enhance the iterative learning process.
- We conduct extensive experiments on real-world datasets from academic and recommendation scenarios. The results demonstrate that LMCH outperforms the best-performing baselines by an average of 5.16% in accuracy and 6.22% in Macro-F1 score.

2 Related Work

Heterogeneous Graph Neural Networks (HGNNs). Embedding the structure and attribute information of HGs into a low-dimensional vector space, Heterogeneous Graph Neural Networks (HGNNs) have demonstrated superior performance in various graph tasks. HGNNs (Wang et al. 2022; Liu et al. 2022) can be classified into two categories depending on whether they model relations or metapaths. Given the rich diversity of node and edge information inherent in HGs, many HGNNs commence their modeling approach directly from the intricate relations encapsulated in HGs (Schlichtkrull et al. 2018; Lu et al. 2019). Relations merely represent an intuitive expression of HG structure, and semantic information is the hidden connotation deep within it. To better explore the rich semantic information contained in HGs, numerous metapath based methods have been developed (Wang et al. 2019; Fu et al. 2020). However, existing HGNNs typically adopt an end-to-end paradigm and do not facilitate cross-heterogeneity transfer across HGs.

Graph Few-shot Learning (GFL). GFL has garnered significant research interest due to the prevalent issue of sparse labels on graphs and meta-learning is one of the mainstream approaches in this field. Most methods are designed for homogeneous graphs (Huang and Zitnik 2020). For example, Meta-GNN (Zhou et al. 2019) leverages MAML (Finn, Abbeel, and Levine 2017) for gradient updates during meta-training, realizing quick adaptation to new tasks in few-shot setting. There are also few studies focusing on HGs (Zhang et al. 2022a). For example, HG-Meta (Zhang et al. 2022b) achieves metapath knowledge transfer in a specific HG via a task feature scaling module and a degree-based task attention module. These methods fail to achieve cross-heterogeneity learning when the source HGs exhibit markedly different heterogeneity with the target HG.

Recently, CGFL (Ding, Wang, and Liu 2023) views ARs and IRs as the general knowledge shared across different HGs and uses them to achieve knowledge transfer. Nevertheless, the definition of general knowledge heavily relies on human knowledge and primarily considers local structures, which would result in suboptimal performance.

Combination of GNN and LM. While GNNs excel at capturing structural information in a graph, pretrained LMs are better at capturing semantic information. A prevalent solution is initially employing a pretrained LM to generate semantically richer embeddings for the graph nodes. These embeddings are then fed into a cascaded GNN to extract the graph's structural information. During this process, the parameters of the LM can either be frozen (Chien et al. 2021; Yasunaga, Leskovec, and Liang 2022) or jointly trained with the GNN (Zhu et al. 2021; Xie et al. 2023). Another approach is to integrate the training of GNN and LM in an iterative process (Zhao et al. 2022; Cai et al. 2024), where they supervise each other. GNN assists LM in better understanding graph structure information, while LM enhances GNN in extracting richer textual attributes. However, most of these methods focus on homogeneous graphs and cannot achieve cross-heterogeneity learning.

3 Preliminaries

In this section, we first introduce some basic concepts and then formalize the problem of cross-heterogeneity graph knowledge transfer.

Definition 1. **Heterogeneous Graph (HG)**. A heterogeneous graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \mathcal{X})$, in which $\mathcal{V}$ represents the set of nodes with a node type mapping function $\varphi : \mathcal{V} \mapsto \mathcal{A}$, and $\mathcal{E}$ denotes the set of edges with an edge type mapping function $\psi : \mathcal{E} \mapsto \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ respectively denote the sets of node and edge types, where $|\mathcal{A}|+|\mathcal{R}| > 2$. In addition, $\mathcal{X}$ represents a textual attribute set of nodes.

Definition 2. **Heterogeneity**. The heterogeneity of $\mathcal{G}_i$, denoted as $\mathcal{H}_i = (\mathcal{A}_i, \mathcal{R}_i)$, contains the node type set $\mathcal{A}_i$ and the edge type set $\mathcal{R}_i$. $\mathcal{G}_i$ and $\mathcal{G}_j$ have different heterogeneity if $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$.

Definition 3. **Metapath**. A metapath P is defined as a path template in the form of $a_1 \xrightarrow{r_1} a_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} a_{l+1}$, which describes a composite relation $r = r_1 \circ r_2 \circ \cdots \circ r_l$ between node types a_1 and a_{l+1}, where $\circ$ denotes the composition operator on relations.

Problem Formulation. Following previous work on cross-heterogeneity graph learning, we adopt exactly the same benchmark setting as CGFL (Ding, Wang, and Liu 2023) based on node classification. Specifically, multiple source HGs are available with abundant labels, whereas a target HG suffers from label sparsity. There are distinct heterogeneity among the source HGs and the target HG. In a target HG, only a few nodes are labeled, which can be abstracted as an *N*-way *K*-shot setting: *N*-way denotes the number of classes and *K*-shot specifies that there are *K* labeled nodes per class. Our objective is to extract general knowledge from the rich-labeled source HGs and subsequently achieve knowledge transfer to a target HG using only $N \times K$ labeled instances.

4 Methodology

In this section, we present our proposed LM-enhanced Cross-Heterogeneity learning model, namely LMCH. Figure 2 presents the overall framework. Firstly, to unify the representations of heterogeneous graphs and preserve more comprehensive information inherent in HGs, we design a metapath-based corpus construction method in Figure 2 (a) to convert different HGs into corpora. Source HG corpora are then used to fine-tune a language model in Figure 2 (b), enabling the LM to acquire general knowledge from various source HGs. Secondly, to leverage the rich information of abundant unlabeled nodes in the few-labeled target HG, we propose an iterative training pipeline in Figure 2 (c), in which the GNN generates soft labels for the LM fine-tuning, and the LM-encoded node embeddings are used as GNN's input. Lastly, to narrow the performance gap and align GNN and LM at the representation level, we employ a LM-GNN contrastive alignment paradigm at the end of each iteration in Figure 2 (d). Specific implementation details are provided in the following subsections.

4.1 Metapath-Based Corpus Construction

Note that metapaths can help extract both local structures and long-range semantic relationship information. In this subsection, we will extract metapath instances from each HG, and convert them into sequences of language tokens as a metapath-based corpus. The construction consists of two steps: node/edge textualization and metapath textualization.

Node/Edge Textualization. We design a rule-based program function $T(\cdot)$ as a template to textualize nodes and edges. Specifically, we concatenate node/edge types and attributes (such as *name*, *age*, etc.). Taking the node type ACTOR with attribute (*age*, *sex*, *name*) in Figure 1 (a) as example, a node with type $\varphi(v) =$ ACTOR and attributes (*age* : 34, *sex* : male, *name* : Guy Edward Pearce) can be transformed into $T(v) =$ $<$ A 34 years old male ACTOR Guy Edward Pearce $>$.

Metapath Textualization. For each HG $\mathcal{G}$ and each node v in the graph, we first initialize the relevant metapaths $P_v = [P_v^1, P_v^2, ..., P_v^n]$, where n represents the number of metapaths associated with node type $\varphi(v)$. For each metapath P_v^k, we will sample metapath instances in the form of $v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \cdots \xrightarrow{e_l} v_{l+1}$. Then the nodes $(v_1, v_2, ..., v_{l+1})$ and edges $(e_1, e_2, ..., e_l)$ in the sampled metapath instance will be processed by node/edge textualization. Subsequently, the texts of each instance are concatenated into the form of $[T(v_1), T(e_1), T(v_2), ..., T(v_l), T(e_l), T(v_{l+1})]$. Finally, all metapath instances regarding of node v will be assembled into sequences of language tokens W_v, with each sequence separated by a [SEP] token. In the HG shown in Figure 1 (b), taking the MOVIE node 1 as an example, there are metapaths MDM and MAM surrounding this node. After metapath textualization, the resulting corpus is depicted as shown in Figure 1 (c).

4.2 Cross-Heterogeneity LM Fine-Tuning

To capture general graph information from the metapath-based corpora from various source HGs, we subsequently fine-tune a language model using these corpora.

First, we employ an LM as an encoder to obtain the representation of node v as:

$$\boldsymbol{z}_v = \text{Mean-Pooling}(\text{LM}(W_v)), \tag{1}$$

where $\boldsymbol{z}_v \in \mathbb{R}^b$ and b is the hidden dimension, W_v is node v's metapath-based sequences of language tokens, $\text{LM}(\cdot)$ denotes the language model adopted as an encoder and the encoded representations are averaged to derive the final representation of node v.

To fully leverage the rich labels in the source HGs, we employ a multi-layer perceptron (MLP) as a decoder to project the encoded-LM embedding $\boldsymbol{z}_v$ into the LM predicted classification $\hat{y}_{v\text{-LM}}$ as $\hat{y}_{v\text{-LM}} = \text{softmax}(\text{MLP}(\boldsymbol{z}_v))$. Then, the optimization objective $\mathcal{L}_{\text{LM}}$ is defined based on cross entropy:

$$\mathcal{L}_{\text{LM}} = -\sum_{v \in \mathcal{Y}} \sum_{c \in \mathcal{C}} y_v^c \log\left(\hat{y}_{v\text{-LM}}^c\right) + \lambda_{\text{LM}} \sum_{\theta \in \Theta_{\text{LM}}} \theta^2, \tag{2}$$

where $\mathcal{Y}$ is the set of nodes that have labels in all source HGs, $\mathcal{C}$ is the set of all classes in source HGs, $y_v^c = 1$ if the true class of node v is c, and 0 otherwise. λ_{LM} is the coefficient of L2 regularization for LM parameters, and Θ_{LM} denotes the parameters of LM that need to be trained.

4.3 GNN-Supervised LM Fine-Tuning

The cross-heterogeneity LM fine-tuning enables the LM to acquire general knowledge from source HGs. To transfer the knowledge for a target HG, we continue to fine-tune the LM on the target dataset.

To fully leverage the rich information of unlabeled nodes in the few-labeled target HG, we propose an iterative training pipeline with the help of an extra GNN predictor. In each iteration, the LM-encoded node embeddings are used as input features for the GNN, which helps GNN rapidly capture general knowledge. Subsequently, GNN generates soft labels for unlabeled nodes and supervises LM fine-tuning. Each iteration initiates with the training of GNN and is followed by the fine-tuning of LM.

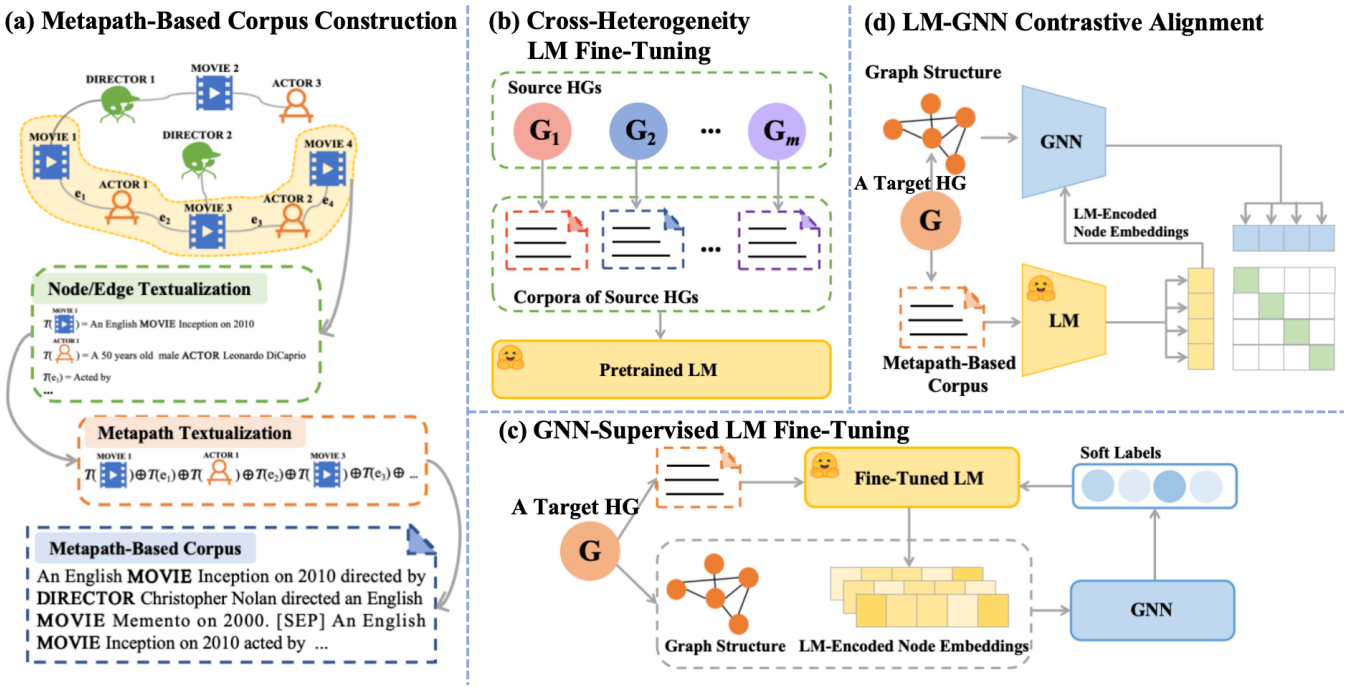

Figure 2: The overall framework of the proposed LMCH.

Specifically, in the GNN training phase, we use a GNN followed by an MLP as the backbone. To enable the GNN to swiftly capture general knowledge, we utilize the LM-encoded embeddings from the preceding iteration as GNN's inputs, with the initial iteration's inputs of GNN directly generated by the cross-heterogeneity fine-tuned LM. The GNN can be implemented using various HGNN methods, such as RGCN (Schlichtkrull et al. 2018), HAN (Wang et al. 2019), or HGT (Zhang et al. 2019). Previous research (Cai et al. 2024) has shown that RGCN has consistently demonstrated strong performance when combined with LM. Consequently, we introduce RGCN in this subsection and utilize it as the GNN in our experiments. Considering the diverse relations in a graph, RGCN employs relation-specific transformations, which can be formalized as:

$$\begin{aligned} \tilde{\boldsymbol{h}}_v^{(l+1)} &= \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{N}_v^r}\frac{1}{|\mathcal{N}_v^r|}\boldsymbol{W}_r^{(l)}\boldsymbol{h}_u^{(l)} + \boldsymbol{W}_0^{(l)}\boldsymbol{h}_v^{(l)}, \\ \boldsymbol{h}_v^{(l+1)} &= \sigma\left(\tilde{\boldsymbol{h}}_v^{(l+1)}\right), \end{aligned} \tag{3}$$

where $\mathcal{N}_v^r$ denotes the set of neighbor nodes of node v under relation $r \in \mathcal{R}$, $\boldsymbol{h}_v^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the representation of node v at the l-th layer with the hidden dimension $d^{(l)}$ and $\sigma(\cdot)$ is an activation function. $\boldsymbol{W}_r^{(l)} \in \mathbb{R}^{f^{(l)}\times d^{(l)}}$ and $\boldsymbol{W}_0^{(l)} \in \mathbb{R}^{f^{(l)}\times d^{(l)}}$ are trainable parameters at the l-th layer, where $f^{(l)}$ is the hidden dimension of node v at the $(l+1)$-th layer. We can stack L layers to get the embeddings $\boldsymbol{h}_v^{(L)} \in \mathbb{R}^{f^{(L)}}$ of node v, with $f^{(L)} = b$. Following an MLP decoder, we can obtain the predicted classification result $\hat{y}_{v\text{-GNN}}$ as $\hat{y}_{v\text{-GNN}} = \text{MLP}(\boldsymbol{h}_v^{(L)})$. The loss of GNN training is similar to $\mathcal{L}_{\text{LM}}$:

$$\begin{aligned} \mathcal{L}_{\text{LM}\to\text{GNN}} = &-\sum_{v\in\mathcal{Y}}\sum_{c\in\mathcal{C}_{tar}} y_{v\text{-GNN}}^c \log\left(\hat{y}_{v\text{-GNN}}^c\right) \\ &+ \lambda_{\text{GNN}}\sum_{\theta\in\Theta_{\text{GNN}}}\theta^2, \end{aligned} \tag{4}$$

where $\mathcal{C}_{tar}$ is the set of labels of the target graph. Subsequently, we select the highest-performing GNN on the validation set in this iteration to produce soft labels $\overline{y}_v$ for unlabeled nodes, which are then used to fine-tune the LM.

The LM fine-tuning process utilizes supervised learning methods and follows the same protocol as in cross-heterogeneity LM fine-tuning, except for incorporating the soft labels for unlabeled nodes into the LM fine-tuning. The optimization objective can be formalized as:

$$\begin{aligned} \mathcal{L}_{\text{GNN}\to\text{LM}} = &-\beta\sum_{v\in\mathcal{Y}}\sum_{c\in\mathcal{C}_{tar}} y_{v\text{-LM}}^c \log\left(\hat{y}_{v\text{-LM}}^c\right) \\ &+ (1-\beta)\sum_{v\in\mathcal{S}}\sum_{c\in\mathcal{C}_{tar}} \overline{y}_{v\text{-LM}}^c \log\frac{\overline{y}_{v\text{-LM}}^c}{\hat{y}_{v\text{-LM}}^c} \\ &+ \lambda_{\text{LM}}\sum_{\theta\in\Theta_{\text{LM}}}\theta^2, \end{aligned} \tag{5}$$

where β serves as a weight coefficient to balance the loss between soft labels of unlabeled nodes and true labels for few-shot nodes in the target dataset. $\mathcal{S}$ denotes the unlabeled node set with soft labels generated by GNN. Upon completing the LM fine-tuning, we select LM with the best performance on the validation set to produce embeddings as GNN's inputs in the next iteration.

4.4 LM-GNN Contrastive Alignment

The previous subsection unifies GNN and LM at the output level. But we empirically find that after the above iterative training process, there is still a performance discrepancy between the LM and the GNN (Zhao et al. 2022). To narrow this gap, we employ a LM-GNN contrastive alignment paradigm at the end of each iteration, based on the typical contrastive loss (Radford et al. 2021; Wen and Fang 2023) adapted for HG modeling. This contrastive optimization can help align GNN and LM at the representation level.

Specifically, we first obtain the sets of LM-encoded node embeddings $\mathbf{Z} \in \mathbb{R}^{|\mathcal{V}| \times b}$ and GNN-encoded node embeddings $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times b}$, where $\boldsymbol{z}_v \in \mathbf{Z}$ and $\boldsymbol{h}_v^{(L)} \in \mathbf{H}$. Subsequently, we apply the LM-GNN contrastive alignment paradigm, formally detailed as follows:

$$
\begin{aligned}
&\hat{\mathbf{H}} = \mathrm{norm}(\mathbf{H}), \hat{\mathbf{Z}} = \mathrm{norm}(\mathbf{Z}), \\
&\mathbf{Q} = \left(\hat{\mathbf{H}}\hat{\mathbf{Z}}^{\top}\right) \cdot \exp(\tau), \\
&\mathcal{L}_{\text{Align}} = -\frac{1}{2} \sum_{t \in \mathbf{T}, \mathbf{q} \in \mathbf{Q}} t \log\left(\mathrm{softmax}(\mathbf{q})\right) \\
&- \frac{1}{2} \sum_{t \in \mathbf{T}, \mathbf{q} \in \mathbf{Q}^{\top}} t \log\left(\mathrm{softmax}(\mathbf{q})\right),
\end{aligned} \tag{6}
$$

where $\mathrm{norm}(\cdot)$ is the L2 normalization, $\mathbf{T}$ is the contrastive labels $(0, 1, \cdots, n-1)^{\top}$, $\mathbf{Q}$ denotes scaled pairwise cosine similarities between $\hat{\mathbf{H}}$ and $\hat{\mathbf{Z}}$, and τ is a trainable temperature parameter to scale the similarity values.

4.5 Discussion

Detailed algorithm is presented in Appendix A.1[1]. Here, we discuss the time complexity of LMCH, which consists of three parts. Given a sequence of length D, the time complexity of LM fine-tuning is $O(B \times (D^2 b + D b^2))$, where B is the number of sequences within a batch. Depending on the number of edges and the embedding dimension, GNN's time complexity is $O(L \times |\mathcal{E}| \times d)$ where d is the hidden dimension of each GNN layer. For LM-GNN contrastive alignment, the time complexity is $O(|\mathcal{V}'|^2 \times b)$, where $|\mathcal{V}'|$ is the number of target type nodes that are significantly smaller than $|\mathcal{V}|$ in the target HG. Since these training processes are performed sequentially, the time complexity of LMCH is the sum of their individual complexities, primarily influenced by the LM fine-tuning and LM-GNN contrastive alignment. A more detailed comparison about the running time of LMCH and baselines can be found in Appendix A.7.

5 Experiments and Analysis

To evaluate the effectiveness of our LMCH model, we perform extensive experiments on four real-world datasets to answer the following research questions:

- **RQ1:** How does LMCH perform compared with baseline methods in cross-heterogeneity scenarios?
- **RQ2:** How does each pivotal component contribute to the overall performance?
- **RQ3:** How does the number of source HGs affect the performance of LMCH?
- **RQ4:** How do different corpus construction methods affect our model performance?

A hyper-parameter study that explores how does different hyper-parameter settings impact our model performance can be found in Appendix A.2. To facilitate a more intuitive comparison of various corpus construction methods, we also performed a case study in Appendix A.3.

5.1 Experimental Settings

Datasets. We conduct extensive experiments on four benchmark datasets: IMDB, DBLP (Wang et al. 2019), YELP (Lu et al. 2019) and PubMed (Zhang et al. 2024a). These datasets exhibit diverse heterogeneity and have been widely used on node classification tasks in HGs. Metapaths used for corpus construction and other detailed statistics of these datasets are summarized in Appendix A.4 (Table 4).

Baselines. To comprehensively assess the performance of our approach, we compare our LMCH with the following 11 representative and state-of-the-art methods from six different categories. More details can be found in Appendix A.6.

For LM with GNN methods, we focus on methods with masked LMs such as BERT (Devlin et al. 2018). The combination with recent large language models (LLMs) using training methods such as LoRA (Hu et al. 2021) is orthogonal to the innovations presented in this work.

- **Homogeneous GNNs:** GCN (Kipf and Welling 2016), GAT (Veličković et al. 2017)
- **Heterogeneous GNNs:** RGCN (Schlichtkrull et al. 2018), HAN (Wang et al. 2019)
- **Few-shot learning methods:** MAML (Finn, Abbeel, and Levine 2017), ProtoNet (Snell, Swersky, and Zemel 2017)
- **Graph few-shot learning methods:** GPN (Ding et al. 2020), G-Meta (Huang and Zitnik 2020)
- **Language model with GNN methods:** GLEM (Zhao et al. 2022), LMBot (Cai et al. 2024)
- **Cross-heterogeneity few-shot learning methods:** CGFL (Ding, Wang, and Liu 2023)

Evaluation Protocols. Following prior research (Ding, Wang, and Liu 2023), we also employ a leave-one-out strategy, wherein one dataset is designated as the target HG while the remaining datasets function as source HGs. For the fair comparison, we leverage a pretrained LM to encode the nodes' attribute information as initial features. We train all models using supervised node classification tasks both on the source and target HGs.

Parameter Settings. In our experiments, few-shot learning follows an N-way K-shot setting, with N in $\{2, 3\}$ and K in $\{1, 3, 5\}$. To extract maximum information from a HG, we randomly sample all available metapaths around each node, with each metapath sampled 10 times. This provides sufficient sequence length for the language model, which

[1]Appendixes can be found in the code repository.

	DBLP		IMDB		YELP		PubMed		DBLP		IMDB		YELP		PubMed	
	2-*way*	3-*way*	2-*way*	3-*way*	2-*way*	3-*way*	2-*way*	3-*way*	2-*way*	3-*way*	2-*way*	3-*way*	2-*way*	3-*way*	2-*way*	3-*way*
	Acc								Ma-F1							
1-*shot*																
GCN	71.28	59.37	47.62	34.05	58.54	41.22	43.22	33.29	67.43	57.24	46.73	32.12	56.33	36.58	41.21	31.78
GAT	73.21	62.30	49.90	35.21	60.37	41.70	45.50	34.21	72.10	60.15	48.32	32.23	58.14	37.25	42.31	32.44
RGCN	67.76	54.67	51.23	36.75	59.41	42.80	46.60	36.42	66.43	53.82	50.49	34.23	55.37	36.37	44.32	35.18
HAN	65.32	53.40	50.26	36.11	58.92	41.34	45.59	34.35	64.22	51.47	50.24	32.81	51.21	34.68	43.52	31.85
MAML	59.28	43.92	50.11	35.27	56.73	39.76	45.11	33.36	58.45	43.77	49.56	34.33	55.83	38.01	42.00	32.24
ProtoNet	61.34	47.02	50.90	35.21	57.23	39.87	45.95	33.54	60.97	48.91	47.66	34.17	56.99	37.10	39.04	28.82
GPN	73.44	64.19	50.37	37.24	66.01	43.82	45.60	34.47	71.23	61.88	49.96	34.34	64.39	41.35	45.57	35.11
G-Meta	75.87	65.74	51.70	36.12	66.32	43.28	46.69	35.24	73.45	63.79	49.67	35.50	58.18	41.09	45.63	34.13
GLEM	60.78	39.25	54.99	38.48	50.90	33.62	33.33	27.69	58.82	38.79	47.09	33.26	42.32	31.88	33.30	25.39
LMBot	61.65	39.32	58.33	37.75	59.38	42.83	40.00	36.92	59.01	39.09	55.90	33.25	53.63	35.44	38.04	28.64
CGFL	87.12	78.49	53.31	38.46	70.28	43.77	53.35	36.76	84.58	75.36	52.44	34.69	68.76	41.26	51.55	35.20
LMCH	**88.87**	**80.08**	**61.69**	**43.51**	**71.52**	**44.26**	**60.00**	**38.24**	**88.85**	**75.98**	**60.46**	**39.35**	**69.83**	**41.54**	**59.82**	**37.29**
3-*shot*																
GCN	78.56	68.44	52.10	34.81	61.27	42.73	44.91	32.21	75.21	66.56	51.24	33.65	59.47	37.55	42.38	31.19
GAT	81.36	71.48	55.43	36.15	63.13	42.12	47.28	31.24	78.83	69.54	50.00	31.13	61.02	35.41	45.83	30.05
RGCN	72.37	61.43	58.20	38.10	65.61	43.40	53.32	37.78	69.23	60.25	53.24	32.54	64.21	36.66	52.10	34.12
HAN	70.59	59.69	56.18	37.26	64.63	42.52	52.87	36.92	67.53	58.88	51.77	32.40	63.29	36.28	50.12	33.44
MAML	65.01	50.33	55.85	36.79	58.23	40.24	51.67	35.62	62.79	49.97	54.76	36.01	57.42	39.54	50.71	33.21
ProtoNet	68.23	56.90	54.74	37.78	59.05	42.26	51.19	34.71	62.89	50.33	47.31	28.15	54.56	35.11	47.68	33.91
GPN	82.79	79.01	57.84	40.65	67.48	48.20	53.79	41.07	80.12	78.93	56.77	38.76	66.32	46.75	52.17	39.37
G-Meta	84.53	81.40	56.47	40.33	68.59	48.43	54.92	40.03	82.45	77.47	55.07	37.19	64.27	45.32	54.73	37.05
GLEM	63.48	39.32	56.47	42.62	54.24	36.31	35.56	41.54	60.70	38.98	47.21	37.09	51.34	30.78	32.21	34.22
LMBot	63.78	41.68	59.00	42.97	61.18	45.81	48.89	43.08	60.75	38.01	57.13	37.71	59.09	42.01	48.41	38.29
CGFL	90.26	85.72	57.45	41.81	75.32	49.04	58.33	42.04	89.74	85.12	55.39	39.57	70.07	46.64	56.15	40.13
LMCH	**97.30**	**92.31**	**63.62**	**44.92**	**76.43**	**49.72**	**62.22**	**44.62**	**97.30**	**91.71**	**60.65**	**44.21**	**75.24**	**49.30**	**56.92**	**41.07**

Table 1: Average node classification accuracy and Macro-F1 score (%) over five runs on four datasets in cross-heterogeneity few-shot settings. The best results are in bold and the second best are underlined. Our results are significantly better than the best baseline method under the 0.01-level student t-test.

is DistilBERT (Sanh et al. 2019), while the GNN used is RGCN (Schlichtkrull et al. 2018). All MLPs are 2-layer networks with a hidden dimension of 128. For fairness, we set the node embedding dimension to 128 for both LMCH and baselines. We apply early stopping to control iterations in the GNN-supervised LM fine-tuning, with a maximum of 10 iterations. LMCH hyper-parameters are optimized via grid-searching for best performance, while baseline parameters are initially set according to the original papers and then optimized. Full hyper-parameter settings are provided in Appendix A.5 (Table 5), with a detailed study in Appendix A.2.

Evaluation Metrics. We assess the performance of all models using Accuracy (Ding, Wang, and Liu 2023) and Macro-F1 score (Fu et al. 2020), which are standard metrics for node classification tasks. For each N-way K-shot setting, we report the average results from five independent runs. More experimental settings can be found in Appendix A.5.

5.2 Experimental Results

Performance Comparison with Baselines (RQ1). Table 1 reports the performances of LMCH and baselines. (1) LMCH outperforms various state-of-the-art baselines across 32 groups in four datasets, with an average improvement of 5.16% in accuracy and 6.22% in Macro-F1 score. (2) End-to-end homogeneous and heterogeneous GNNs are limited by single-heterogeneity and label scarcity, leading to poor results. (3) GFL and LM-GNN methods achieve second-best performance on IMDB and YELP datasets, demonstrating their strong capabilities in few-shot scenario. However, the lack of consideration for cross-heterogeneity scenarios ultimately leads to suboptimal outcomes. (4) Although CGFL achieves near-optimal result, its reliance on predefined general knowledge based on human expertise limits flexibility and hinders further performance improvements.

	DBLP		IMDB	
	Acc	**Ma-F1**	**Acc**	**Ma-F1**
LMCH-V1	84.88	83.31	42.86	42.23
LMCH-V2	90.93	89.99	44.64	43.83
LMCH-V3	88.43	87.48	34.94	33.87
LMCH-V4	91.45	90.75	40.09	39.00
LMCH	**92.31**	**91.71**	**44.92**	**44.21**

Table 2: Node classification accuracy and Macro-F1 score (%) of LMCH variants on DBLP and IMDB in a 3-way 3-shot setting.

These findings robustly demonstrate that our approach effectively enables cross-heterogeneity few-shot learning in HGs, whereas baseline methods struggle to extract general knowledge from HGs with distinct heterogeneity or face significant challenges in few-shot scenarios.

Ablation Study (RQ2). To validate the impact of each component on the model's performance, we conduct experiments on various LMCH variants. Here, **LMCH-V1** indicates that our model excludes the metapath-based corpora and instead uses the nodes' own attributes as input for LM fine-tuning. **LMCH-V2** denotes the LM is not fine-tuned in advance and is directly used in GNN-supervised LM fine-tuning. **LMCH-V3** denotes that the LM is fine-tuned without GNN supervision after LM cross-heterogeneity fine-tuning. **LMCH-V4** signifies that the model does not perform LM-GNN contrastive alignment at the end of each iteration. The results are presented in Table 2.

These variants consistently perform worse than the original LMCH, underscoring the importance of each model component. In the DBLP dataset, LMCH-V1 performs the worst, while LMCH-V3 underperforms on IMDB. This may be due to DBLP's reliance on long-range semantic information from longer metapaths, whereas IMDB depends more on local structure. Without GNN-generated soft labels for LM fine-tuning, the LM lacks sufficient information, leading to poor performance on IMDB.

Impact of the Number of Source HGs (RQ3). We analyze the impact of varying the number of source HGs on LMCH performance. Source HGs used in the experiments are randomly selected to ensure a fair comparison. Figure 3 shows performance improves with more source HGs, as this enables the LM to acquire richer information and extract broader general knowledge.

Impact of Corpus Construction Methods (RQ4). There are various methods for corpus construction. We evaluate the impact of different approaches individually, including relying solely on node attribute method (**NA-based**), first-order neighbor-based method (**FN-based**), random walk-based method (**RW-based**), and metapath-based method (**MP-based**), as shown in Table 3. The results reveal that the metapath-based method outperforms the others. This demonstrates that the metapath-based method can extract as much information as possible from HGs, thereby enabling more knowledge transfer in the target HG. We also observe that in the DBLP dataset, the random walk-based method

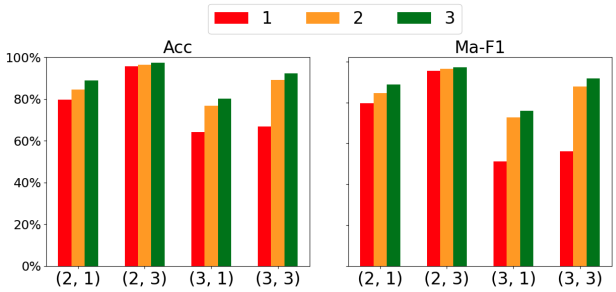

Figure 3: Node classification accuracy and Macro-F1 score of LMCH on DBLP with varying numbers of source HGs under (n-way, k-shot) settings.

	DBLP		IMDB	
	Acc	**Ma-F1**	**Acc**	**Ma-F1**
NA-based	82.31	80.74	36.11	21.47
FN-based	71.53	67.73	44.12	43.72
RW-based	88.63	88.03	42.76	42.61
MP-based	**92.31**	**91.71**	**44.92**	**44.21**

Table 3: Node classification accuracy and Macro-F1 score (%) of LMCH with different corpus construction methods on DBLP and IMDB in a 3-way 3-shot setting.

delivers the second-best performance, while in the IMDB dataset the suboptimal outcome is achieved by the first-order neighbor-based method. This is due to the fact that the IMDB dataset primarily relies on local structures, while the DBLP dataset depends more on long-range semantic information. To facilitate a more intuitive comparison, we perform a case study of the LM-encoded node embeddings with different corpus construction methods. Further details about the case study can be found in Appendix A.3.

6 Conclusion

In this paper, we propose an advanced LM-enhanced Cross-Heterogeneity learning model called LMCH. The core idea of LMCH is to first unify HG representations as languages for automatically extracting general knowledge from source HGs, and then transfer this knowledge to the target HG. As far as we know, we are the first to unify HG representations through metapath-based language and utilize LM for autonomous general knowledge extraction, establishing a new paradigm for cross-heterogeneity learning and paving a new path for future research. Extensive experiments have demonstrated the superior performance of LMCH. For future work, we will explore the possibility of using larger LMs such as ChatGPT and GPT-4 (Achiam et al. 2023).

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No.2023YFC3303800).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Cai, Z.; Tan, Z.; Lei, Z.; Zhu, Z.; Wang, H.; Zheng, Q.; and Luo, M. 2024. LMbot: distilling graph knowledge into language model for graph-less deployment in twitter bot detection. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 57–66.

Chien, E.; Chang, W.-C.; Hsieh, C.-J.; Yu, H.-F.; Zhang, J.; Milenkovic, O.; and Dhillon, I. S. 2021. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, K.; Wang, J.; Li, J.; Shu, K.; Liu, C.; and Liu, H. 2020. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 295–304.

Ding, P.; Wang, Y.; and Liu, G. 2023. Cross-heterogeneity graph few-shot learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 420–429.

Dong, Y.; Tang, J.; Wu, S.; Tian, J.; Chawla, N. V.; Rao, J.; and Cao, H. 2012. Link prediction and recommendation across heterogeneous social networks. In *2012 IEEE 12th International conference on data mining*, 181–190. IEEE.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Fu, X.; Zhang, J.; Meng, Z.; and King, I. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, 2331–2341.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, K.; and Zitnik, M. 2020. Graph meta learning via local subgraphs. *Advances in neural information processing systems*, 33: 5862–5874.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Liu, J.; Shi, C.; Yang, C.; Lu, Z.; and Philip, S. Y. 2022. A survey on heterogeneous information network based recommender systems: Concepts, methods, applications and resources. *AI Open*, 3: 40–57.

Lu, Y.; Shi, C.; Hu, L.; and Liu, Z. 2019. Relation structure-aware heterogeneous information network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 4456–4463.

Ma, A.; Wang, X.; Li, J.; Wang, C.; Xiao, T.; Liu, Y.; Cheng, H.; Wang, J.; Li, Y.; Chang, Y.; et al. 2023. Single-cell biological network inference using a heterogeneous graph transformer. *Nature Communications*, 14(1): 964.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, 593–607. Springer.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, X.; Bo, D.; Shi, C.; Fan, S.; Ye, Y.; and Philip, S. Y. 2022. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 9(2): 415–436.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019. Heterogeneous graph attention network. In *The world wide web conference*, 2022–2032.

Wen, Z.; and Fang, Y. 2023. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 506–516.

Xie, H.; Zheng, D.; Ma, J.; Zhang, H.; Ioannidis, V. N.; Song, X.; Ping, Q.; Wang, S.; Yang, C.; Xu, Y.; et al. 2023. Graph-aware language model pre-training on a large graph corpus can help multiple graph applications. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5270–5281.

Yan, B.; Cao, Y.; Wang, H.; Yang, W.; Du, J.; and Shi, C. 2024. Federated heterogeneous graph neural network for privacy-preserving recommendation. In *Proceedings of the ACM on Web Conference 2024*, 3919–3929.

Yasunaga, M.; Leskovec, J.; and Liang, P. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 793–803.

Zhang, Q.; Wu, X.; Yang, Q.; Zhang, C.; and Zhang, X. 2022a. Few-shot heterogeneous graph learning via cross-domain knowledge transfer. In *Proceedings of the 28th ACM*

SIGKDD Conference on Knowledge Discovery and Data Mining, 2450–2460.

Zhang, Q.; Wu, X.; Yang, Q.; Zhang, C.; and Zhang, X. 2022b. HG-Meta: Graph meta-learning over heterogeneous graphs. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, 397–405. SIAM.

Zhang, Z.; Wang, X.; Zhou, H.; Yu, Y.; Zhang, M.; Yang, C.; and Shi, C. 2024a. Can Large Language Models Improve the Adversarial Robustness of Graph Neural Networks? *arXiv preprint arXiv:2408.08685*.

Zhang, Z.; Zhang, M.; Yu, Y.; Yang, C.; Liu, J.; and Shi, C. 2024b. Endowing Pre-trained Graph Models with Provable Fairness. In *Proceedings of the ACM on Web Conference 2024*, 1045–1056.

Zhao, J.; Qu, M.; Li, C.; Yan, H.; Liu, Q.; Li, R.; Xie, X.; and Tang, J. 2022. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019. Meta-gnn: On few-shot node classification in graph meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2357–2360.

Zhu, J.; Cui, Y.; Liu, Y.; Sun, H.; Li, X.; Pelger, M.; Yang, T.; Zhang, L.; Zhang, R.; and Zhao, H. 2021. Textgnn: Improving text encoder via graph neural network in sponsored search. In *Proceedings of the Web Conference 2021*, 2848–2857.

Zhuang, Z.; Xiang, X.; Huang, S.; and Wang, D. 2021. Hinfshot: A challenge dataset for few-shot node classification in heterogeneous information network. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 429–436.