Multi-Aspect Heterogeneous Graph Augmentation

Anonymity Version

ABSTRACT

Data augmentation has been widely studied as it can be used to improve the generalizability of graph representation learning models. However, existing works focus only on the data augmentation on homogeneous graphs. Data augmentation for heterogeneous graphs remains under-explored. Considering that heterogeneous graphs contain different types of nodes and links, ignoring the type information and directly applying the data augmentation methods of homogeneous graphs to heterogeneous graphs will lead to suboptimal results. In this paper, we propose a novel Multi-Aspect Heterogeneous Graph Augmentation framework named MAHGA. Specifically, MAHGA consists of two core augmentation strategies: structure-level augmentation and metapath-level augmentation. Structure-level augmentation pays attention to network schema aspect and designs a relation-aware conditional variational autoencoder that can generate synthetic features of neighbors to augment the nodes and the node types with scarce links. Metapath-level augmentation concentrates on metapath aspect, which constructs metapath reachable graphs for different metapaths and estimates the graphons of them. By sampling and mixing up based on the graphons, MAHGA yields intra-metapath and inter-metapath augmentation. Finally, we conduct extensive experiments on multiple benchmarks to validate the effectiveness of MAHGA. Experimental results demonstrate that our method improves the performances across a set of heterogeneous graph learning models and datasets.

CCS CONCEPTS

• **Computing methodologies** → **Semi-supervised learning settings**; • **Regularization**; • **Neural networks**;

KEYWORDS

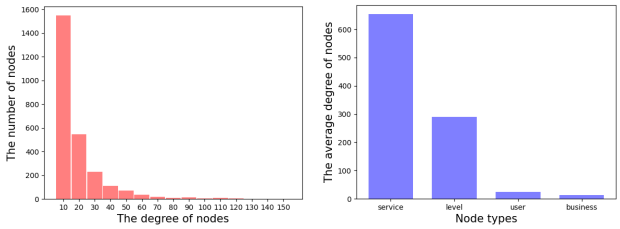
Data Augmentation, Heterogeneous Graph Neural Networks, Heterogeneous Information Network, Graph Mining

ACM Reference Format:

Anonymity Version. 2023. Multi-Aspect Heterogeneous Graph Augmentation. In *Proceedings of In Proceedings of the ACM Web Conference 2023 (WWW '23).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Data augmentation as an effective strategy to improve the generalization capability and performance of model has got widespread adoption in various fields such as computer vision (CV) [6, 11, 33] and natural language processing (NLP) [3, 14, 15]. The core idea of data augmentation is to design various augmentation strategies to generate new plausible data based on existing data without additional ground-truth labels so as to enhance the quantity and/or the quality of existing data. Since graph learning usually faces with many dilemmas such as feature data incompleteness, structural data sparsity brought by power-law distributions, costly data annotations and so on, data augmentation naturally provides a good solution to help graph learning models deal with above problems. However, due to the irregular and non-Euclidean nature of graph data, the structured data augmentation operations used frequently in CV and NLP cannot be applied to graph learning models. Moreover, different information modalities and graph properties yield a broader design space for graph data augmentation. Therefore, more and more researchers are starting to pay their attention to the data augmentation for graph data.



(a) (b)

Figure 1: The distribution of node degree on yelp dataset. (a) depicts the degree distribution of different business nodes. (b) depicts the average degree distribution of different types of nodes.

Recently, there has been a growing number of works on graph data augmentation [26, 29, 31, 34]. Despite their successes, all the current augmentation methods are developed for homogeneous graph and there has never been a work exploring the data augmentation on heterogeneous graphs. Heterogeneous graphs which come with multi-types of nodes and links are ubiquitous in real-world scenarios, ranging from bibliographic networks, social networks to recommendation systems. They usually contain more comprehensive information and richer semantics than homogeneous graphs. Directly adapting homogeneous graph augmentation methods to heterogeneous graph results in the loss of type information and semantics. Moreover, it even introduces unexpected noise which hurts the performance of graph learning models. Consequently, it is important to develop a special data augmentation framework for heterogeneous graphs.

In order to effectively augment heterogeneous graphs, we first analyse the common challenges faced by heterogeneous graph learning models. On the one hand, besides the data skew caused by the power-law distribution of the graph, there are extremely imbalanced regarding different node types because the degree distribution of different types of nodes varies dramatically. To verify above phenomena, we depict the degree distribution of different

WWW '23, April 30– May 04, 2023, Austin, Texas, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

nodes of the same type and the average degree distribution of different types of nodes in Figure 1. From the figure, it is clear that there are significant differences in the degrees of different types of nodes. Even for nodes of the same type, the degree distribution is not uniform. Therefore, we argue that the link imbalances between different nodes and between different types of nodes restrict the performance of heterogeneous graph learning models. Especially, the nodes and the node types with a limited number of links cannot provide sufficient information and become an information bottleneck.

On the other hand, metapath, which describes a composite relation between the node types involved, has been widely used to describe the diverse semantics of a heterogeneous graph. Taking a bibliographic graph as an example, a metapath Paper-Author-Paper (PAP) represents that two papers are written by the same author, while Paper-Subject-Paper (PSP) represents that two papers belong to the same subject. Even though metapath is successful at improving performance by giving a clear guide to heterogeneous graph learning models, existing models get metapath-based node relations from observed heterogeneous graphs. However, the raw observed graphs are usually extracted from complex real-world interaction systems by some predefined rules. They are often noisy or even incomplete due to the inevitably error-prone data measurement or collection. Therefore, obtaining metapath based node relations directly from the observed graph leads to inaccurate results and affects the performance of heterogeneous graph learning models. Furthermore, existing models explicitly offer the sequences of node types to define the metapaths which needs sufficient domain knowledge and is difficult to extend. In summary, defining a good set of metapaths and obtaining accurate metapath based node relations are challenging for heterogeneous graph learning models.

In order to tackle the challenges addressed above, we develop a novel Multi-Aspect Heterogeneous Graph Augmentation framework (MAHGA) that contains two aspects of augmentation strategy: structure-level augmentation and metapath-level augmentation. Structure-level augmentation focuses on the local structures of nodes and aims to augment the neighbor information. Specifically, it designs a relation-aware conditional variational auto-encoder to learn the conditional distribution of neighbor nodes' features given the center node's features and the type of neighbor nodes. By sampling from the learned distribution, we can generate synthetic neighborhood features to augment the nodes and the node types with scarce links. Metapath-level augmentation follows with interest metapath information and employs graphon as a generator to conduct intra-metapath and inter-metapath augmentation. Graphon, a function that determines the matrix of edge probabilities, reflects the underlying topology structure of graph so that it is well suited to deal with the challenges in metapath aspect. We first construct metapath reachable graphs for pre-defined metapaths and estimate the graphons of these graphs. In intra-metapath augmentation, we repeatedly sample from the graphon to generate several new metapath reachable graphs for each metapath. By training heterogeneous graph learning models based on generated and original metapath reachable graphs, MAHGA improves the generalization of heterogeneous graph learning model and alleviates the inaccuracy of metapath caused by the mistakes and incompleteness of observed graph. In inter-metapath augmentation, we draw on the idea of mixup augmentation method which has been widely used in CV field [27, 35, 36]. By mixing the graphons of metapath reachable graphs of different pre-defined metapaths, we can generate many new graphons which imply the new metapaths. What's more, the new metapaths determined by augmented graphons are implicit, meaning that we do not require extra domain knowledge to explicitly define the node type sequences of them.

In summary, the overall contributions of this paper can be summarized as follow:

- We analyse the common challenges faced by heterogeneous graph learning models and use them as the guide to design effective data augmentation strategies for heterogeneous graph.
- We propose MAHGA, which is the first work to explore data augmentation on heterogeneous graphs. MAHGA constructs a relation-aware conditional variational auto-encoder and utilizes graphons of metapath reachable graphs to achieve structure-level and metapath-level augmentation.
- We conduct extensive experiments on three different datasets to show the effectiveness of our augmentation framework for improving mainstream heterogeneous graph learning models. The experimental results demonstrate that the state-of-the-art homogeneous graph augmentation methods cannot adapt to heterogeneous graph well. In contrast, our augmentation framework yields significant gains for heterogeneous graph learning models.

2 RELATED WORK

In this paper, we review some representative works on graph data augmentation and heterogeneous graph representation learning. Considering that there is no work to research the data augmentation on heterogeneous graphs, we only introduce the relevant literature in homogeneous graph data augmentation.

2.1 Graph Data Augmentation

Graph data augmentation aims to create new graph data via slightly modified copies of the original graph, or generate synthetic data based on the original graph to improve the generalization ability of graph learning models. According to the different augmentation objects, the existing graph augmentation methods can be divided into three categories: node-centralized augmentation, edge-centralized augmentation and graph-centralized augmentation.

Node-centralized augmentation methods take nodes as basic objects to design augmentation operations. NodeAug[30] designs three augmentation strategies including attribute replacing, edge removing and edge adding for nodes and regularizes the model predictions of nodes to be invariant with respect to changes induced by augmentation. NASA[2] defines augmentation operation as randomly replacing the immediate neighbors of nodes with their remote neighbors, and designs consistency and diversity metrics to improve the correctness and generalization of augmentation. LA[16] learns the distribution of the node representations of the neighbors conditioned on the central node's representation and generate new neighbor features to augment the central nodes.

Edge-centralized augmentation methods mainly conduct augmentation operations on edges. DropEdge[21] randomly removes a certain number of edges from the original graph at each training epoch. GAUG[39] designs a neural edge predictor to strategically

choose ideal edges to add or remove. CFLP[38] employs causal model to create counterfactual edges to improve the model performance.

Graph-centralized augmentation methods are more coarse-grained than above two augmentation methods as they directly generate synthetic graphs. MH-Aug[20] defines an explicit target distribution and draws augmented graphs from the distribution so that it enables flexible control of the strength and diversity of augmentation. Suresh et al.[23] employ information bottleneck principle to control the augmentation process and improve the qualities of augmented graphs. To avoid the generating of unbeneficial augmented graphs, MEGA[9] designs a learnable graph augmenter and trains a graph learning model based on a meta-learning paradigm.

Although above augmentation methods promote the performance of graph learning model, they are only designed for homogeneous graph and ignore the type information and the rich semantics of heterogeneous graph, which are important to heterogeneous graph learning models.

2.2 Heterogeneous Graph Representation Learning

Heterogeneous graph representation learning aims to model rich semantics of heterogeneous graph to learn low-dimensional node embedding, which can be used in various downstream tasks. CompGCN[24] leverages a variety of entity-relation composition operations from knowledge graph embedding techniques to embed both nodes and relations in a heterogeneous graph. Inspired by the architecture design of Transformer[25], HGT[12] proposes a heterogeneous mutual attention mechanism and uses it to achieve heterogeneous message passing and aggregation. Simple-HGN[18] combines three well-known techniques: learnable edge-type embedding, residual connections and normalization to design a heterogeneous neighborhood aggregation mechanism to model the complex neighborhood structure of nodes.

Except for modeling the local structure, there are many works that adopt metapath which defines high-order composite relation between nodes to capture meaningful semantics. HAN[28] designs node-level attention and semantic-level attention to model the importance of different metapath based neighbors and different metapaths respectively. Node embedding can be obtained by hierarchically aggregating features from metapath based neighbors. MAGNN[8] considers the semantics of intermediate nodes of metapath and designs intra-metapath and inter-metapath aggregation mechanism to acquire node embedding. HPN[13] designs the semantic propagation mechanism and the semantic fusion mechanism to alleviate semantic confusion and builds a more powerful heterogeneous graph learning architecture to learn node embedding.

3 PRELIMINARY

In this section, we first give formal definitions of some important terminologies related to heterogeneous graph. Then, we present the heterogeneous graph augmentation problem.

Definition 1. ***Heterogeneous Graph:*** *A heterogeneous graph is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ associated with a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and an edge type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the predefined sets of node types and edge types respectively, with $|\mathcal{A}| + |\mathcal{R}| > 2$.*

Definition 2. ***Metapath:*** *A metapath Φ is defined as a path in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{l-1}} A_l$ (abbreviated as $A_1A_2\cdots A_l$), which describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_{l-1}$ between objects A_1 and A_l, where $\circ$ denotes the composition operation on relations.*

Definition 3. ***Metapath based Neighbors:*** *Given a node i and a metapath Φ in a heterogeneous graph $\mathcal{G}$, the metapath Φ based neighbors $\mathcal{N}_i^{\Phi}$ of node i are defined as the set of nodes which connect with node i via metapath Φ.*

Definition 4. ***Metapath Reachable Graph:*** *Given a metapath Φ in a heterogeneous graph $\mathcal{G}$, the metapath reachable graph $\mathcal{G}^{\Phi}$ is constructed by all the metapath Φ based neighbor pairs in graph $\mathcal{G}$. Note that $\mathcal{G}^{\Phi}$ is homogeneous if Φ is symmetric.*

Definition 5. ***Heterogeneous Graph Augmentation:*** *Given a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, heterogeneous graph augmentation aims to find a mapping function $f_\theta : \mathcal{G} \rightarrow \overline{\mathcal{G}}$ such that the augmented graph $\overline{\mathcal{G}} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ can be used to improve the generalization ability of a heterogeneous graph learning model.*

4 MULTI-ASPECT HETEROGENEOUS GRAPH AUGMENTATION

In this section, we introduce the MAHGA whose overall structure is shown in Figure 2. MAHGA contains structure-level augmentation and metapath-level augmentation, which are designed for the augmentation of complex structure and meaningful metapath respectively. In structure-level augmentation, we construct a relation-aware conditional variational auto-encoder (RCVAE) which considers the type information in graph data augmentation. Given the central node and the neighbor type planned to augment, RCVAE can generate synthetic features of neighbors to augment the local structure of the central node. In metapath-level augmentation, we first construct a metapath reachable graph for each pre-defined metapath. Then, we estimate the graphons of these metapath reachable graphs and conduct intra-metapath and inter-metapath augmentation. For intra-metapath augmentation, we use graphon to define the Bernoulli distribution and resample several synthetic metapath reachable graphs to augment the semantic information of current metapath. For inter-metapath augmentation, we mixup the graphons of metapath reachable graphs of different metapaths to generate new graphons which can be regarded as new metapaths. Then, we generate synthetic metapath reachable graphs by sampling from above new graphons to augment the semantic information between metapaths.

4.1 Structure-level Augmentation

Structure-level augmentation aims to provide more data to augment the nodes and the node types with scarce links so as to alleviate the sparsity and imbalance issues of heterogeneous graph data. The intuitive solution is to strategically select some nodes as the new neighbors of the central node to augment the local structure of the central node. However, this method has two main drawbacks. 1)

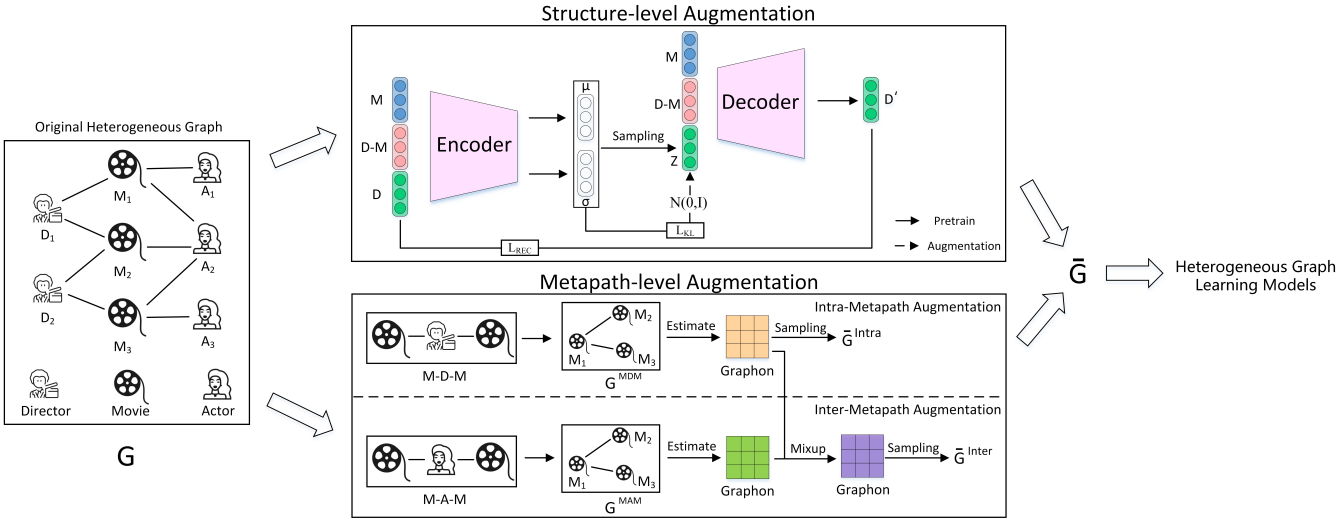


Figure 2: The overall framework of the proposed MAHGA

The good node selection strategy is difficult to define. Once the inappropriate nodes are selected as the new neighbors, the additional unexpected noise will be introduced to the heterogeneous graph learning model and affect the model performance. 2) This method can only select nodes that already exist in the heterogeneous graph as new neighbors to conduct augmentation. However, for the central node, it is possible that there is no appropriate node that can be selected in the heterogeneous graph. Therefore, in order to overcome above limitations, we design a generative model to learn the features' distribution of neighbors and directly sample new neighbors from the distribution rather than select existing nodes as new neighbors. Next, we will elaborate the generative model and the data augmentation process.

4.1.1 Relation-aware Conditional Variational Auto-encoder. In a heterogeneous graph, the features of different types of nodes are in different feature spaces. Therefore, to accurately model the features' distribution of neighbors, it is necessary to consider both the features of the central node and the types of neighbors. Based on the above motivation, we draw on the idea of the conditional variational autoencoder (CVAE) [16, 22] to construct the relation-aware conditional variational auto-encoder (RCVAE) as the generative model. Formally, given the features $\mathbf{X}_v$ of the central node v and the relation type embedding $\mathbf{r}_{uv}$, we have

$$\begin{aligned} log p_\theta(\mathbf{X}_u|\mathbf{X}_v,\mathbf{r}_{uv}) &= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} dz \\ &+ KL(q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})||p_\theta(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})) \\ &\geq \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} dz \end{aligned} \tag{1}$$

where $\mathbf{X}_u$ is the features of neighbor of the central node v. ϕ and θ are the variational parameters and generative parameters respectively. $\mathbf{z}$ is the latent variable which is generated from the prior distribution $p_\theta(\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})$. As relation type implies the types of two ends, we employ learnable relation type embedding as the condition in this paper. Then, the evidence lower bound (ELBO) can be written as:

$$\begin{aligned} L(\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv};\theta,\phi) &= -KL(q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})||p_\theta(\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})) \\ &+ \frac{1}{N^v}\sum_{n=1}^{N^v} log p_\theta(\mathbf{X}_u|\mathbf{z}^n,\mathbf{X}_v,\mathbf{r}_{uv}) \end{aligned} \tag{2}$$

where N^v is the number of neighbors of the central node v. $\mathbf{z}^n \sim \mathcal{N}(\boldsymbol{\mu}_{uv}, \boldsymbol{\sigma}^2_{uv})$ is generated by reparameterization trick. The mean $\boldsymbol{\mu}_{uv}$ and variance $\boldsymbol{\sigma}^2_{uv}$ of the distribution are generated by the encoder $g^{enc}_\phi(\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})$ of RCVAE. Finally, we sample a latent variable $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ as input for the decoder $g^{dec}_\theta(\mathbf{z},\mathbf{X}_v,\mathbf{r}_{uv})$ of RCVAE to obtain the synthetic neighbor features $\overline{\mathbf{X}}_v$.

4.1.2 Pre-training and Augmentation. In general, for a specific downstream task, We can use Maximum Likelihood Estimation (MLE) to estimate the parameter Υ of a heterogeneous graph learning model. It optimizes the following likelihood function:

$$max \prod_i P_\Upsilon(\mathbf{Y}_i|\mathcal{G}) \tag{3}$$

where $\mathbf{Y}$ is the class labels of downstream task. i represents the i-th data point in the training dataset. As our augmentation framework uses the original heterogeneous graph data to train RCVAE and employs RCVAE to generate synthetic neighbor features $\overline{\mathbf{X}}$ to conduct augmentation, The likelihood function in Eq.(3) can be rewritten as follows:

$$max \prod_i \int_{\overline{\mathbf{X}}} P_\Upsilon(\mathbf{Y}_i,\overline{\mathbf{X}}|\mathcal{G}) \tag{4}$$

According to the Bayes Rule, we can further decompose P_Υ in Eq.(4) as a product of two posterior probabilities:

$$P_{\Upsilon,\phi}(\mathbf{Y}_i,\overline{\mathbf{X}}|\mathcal{G}) = P_\Upsilon(\mathbf{Y}_i|\overline{\mathbf{X}},\mathcal{G}) Q_\Psi(\overline{\mathbf{X}}|\mathcal{G}) \tag{5}$$

where $P_\Upsilon(\mathbf{Y}_i|\overline{\mathbf{X}},\mathcal{G})$ and $Q_\Psi(\overline{\mathbf{X}}|\mathcal{G})$ are the probabilistic distributions approximated by heterogeneous graph learning model and RCVAE

respectively. By the decomposition, we decouple the training processes of RCVAE and heterogeneous graph learning model. Therefore, we can pre-train RCVAE first and then apply it to conduct augmentation. Specifically, in pre-training stage, we extract neighboring triple $(\mathbf{X}_u, \mathbf{X}_v, \mathbf{r}_{uv})$ from the original heterogeneous graph to train RCVAE by maximizing the ELBO (Eq. (2)). In augmentation stage, we employ RCVAE to generate the synthetic neighbor features and combine them with the original features of the central node as the augmented features of the central node. Finally, we use the augmented features to train various heterogeneous graph learning models so as to improve the performances of these models.

4.2 Metapath-level Augmentation

Metapath-level augmentation aims to alleviate the incompleteness and mistake issues in the metapath based sampling (intra-metapath augmentation), while generating new synthetic metapath without extra domain knowledge (inter-metapath augmentation). However, metapath is more complex than the adjacency relation as different metapaths contain different node types and the length of them varies. It is difficult to design a strategy to directly augment metapath. Consequently, we use metapath based neighbors to transform metapaths into metapath reachable graphs and estimate the graphons of these graphs. Then, we design augmentation strategies operated on graphon to indirectly augment metapath as graphon can maintain the semantics of metapath by modeling the latent structure of metapath reachable graph. Next, we will elaborate more details about metapath-level augmentation.

4.2.1 Graphon Estimation. Graphon is a nonparametric graph model which characterizes the observed graph and reflects its latent graph structure. Mathematically, a graphon is a two-dimensional symmetric Lebesgue measurable function, denoted as $\mathbf{W} : \Omega^2 \mapsto [0, 1]$, where Ω is a measure space, e.g.,$\Omega = [0, 1]$. As graphon does not have a closed-form expression, how to robustly learn graphon from observed graph becomes a thorny problem. Existing methods mainly depend on the weak regularity lemma of graphon[7] to solve the aforementioned problem, they learn a two-dimensional step function to approximate graphon. The step function $\mathbf{W}^P : [0, 1]^2 \mapsto [0, 1]$ is defined as $\mathbf{W}^P(x, y) = \sum_{k,k'}^{K} w_{k,k'} \mathbb{1}_{P_k \times P_{k'}}(x, y)$, where $P = (P_1, ..., P_k)$ partitions the interval $[0, 1]$ into K adjacent intervals with a length of $1/K$. Each $w_{k,k'} \in [0, 1]$ and the indicator function $\mathbb{1}_{P_k \times P_{k'}}$ is 1 if $(x, y) \in P_k \times P_{k'}$, otherwise it is 0. There are many step function learning methods, e.g., the sorting-and-smoothing method (SAS)[4], the stochastic block approximation (SBA)[1], the universal singular value thresholding algorithm (USVT)[5] and so on. In this paper, we employ structured Gromov-Wasserstein barycenters method (SGWB)[32] because it is a computationally-efficient algorithm with solid theoretical guarantee.

More specifically, we first define the squared 2-order Gromov-Wasserstein distance as follows:

$$d^2_{gw,2}(\mathbf{W}_1, \mathbf{W}_2) = min_{\mathbf{T} \in \Pi(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)} \langle \mathbf{D} - 2\mathbf{W}_1 \mathbf{T} \mathbf{W}_2^\top, \mathbf{T} \rangle \tag{6}$$

where $\mathbf{W}_1 = [w_{1,ij}] \in [0, 1]^{I \times I}$ and $\mathbf{W}_2 = [w_{2,i'j'}] \in [0, 1]^{J \times J}$ are the step functions of two graphs. Vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ represent the marginal probability measures in partitions. $\mathbf{T} = [T_{ii'}] \in \mathbb{R}^{I \times J}$ is a doubly-stochastic matrix in the set $\Pi(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \{\mathbf{T} \geq \mathbf{0} | \mathbf{T}\boldsymbol{\mu}_2 = \boldsymbol{\mu}_1, \mathbf{T}^\top \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2\}$, whose element $T_{ii'} = \int_{P_i \times Q_{i'}} d\pi(x, x')$. The matrix $\mathbf{T}$ defines a transport or coupling between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. $\langle \cdot, \cdot \rangle$ is the inner product of two matrices. $\mathbf{D} = (\mathbf{W}_1 \odot \mathbf{W}_1)\boldsymbol{\mu}_1 \mathbf{1}_J^\top + \mathbf{1}_I \boldsymbol{\mu}_2^\top (\mathbf{W}_2 \odot \mathbf{W}_2)$. $\mathbf{1}_I$ and $\mathbf{1}_J$ are the I-dimensional and J-dimensional all-one vectors. $\odot$ represents the Hadamard product. Then, we learn the optimal step function by minimizing the Gromov-Wasserstein distance between the observed graph and step function of the target graphon:

$$min_{\mathbf{W} \in [0,1]^{K \times K}} \frac{1}{M} \sum_{m=1}^{M} d^2_{gw,2}(\mathbf{A}_m, \mathbf{W}) \tag{7}$$

where M is the number of observed graphs and $\mathbf{A}_m$ is the adjacency matrix of m-th observed graph. Given the transports $\{\mathbf{T}_m\}_{m=1}^M$, the above problem has a closed-form solution as follows:

$$\boldsymbol{\mu}_W = \frac{1}{M} \sum_{m=1}^{M} interp1d_K(sort(\boldsymbol{\mu}_m)) \tag{8}$$

$$\mathbf{W} = \frac{1}{\boldsymbol{\mu}_W \boldsymbol{\mu}_W^\top} \sum_{m=1}^{M} \mathbf{T}_m^\top \mathbf{A}_m \mathbf{T}_m \tag{9}$$

where $\boldsymbol{\mu}_m = \frac{1}{||\mathbf{A}_m \mathbf{1}_{N_m}||_1} \mathbf{A}_m \mathbf{1}_{N_m}$. N_m is the node number of m-th graph. $sort(\cdot)$ sorts the elements of the input vector in descending order and $interp1d_k(\cdot)$ samples K values from the input vector via linear interpolation. After obtaining graphon $\mathbf{W}$, we can conduct the intra-metapath and inter-metapath augmentation.

4.2.2 Intra-metapath Augmentation. Intra-metapath augmentation uses estimated graphon to augment a single metapath. Specifically, for each metapath, it directly samples new synthetic metapath reachable graphs from graphon. All the generated synthetic graphs reflect the semantic information of current metapath. By training heterogeneous graph learning models on the synthetic graphs, intra-metapath augmentation can effectively improve the generalization ability of models and alleviate the influences of data incompleteness and mistake. Formally, for metapath ϕ, the sampling process of synthetic metapath reachable graph $\mathcal{G}^\phi$ can be formulated as follow:

$$\begin{aligned} &v_1, v_2, ..., v_N \overset{iid}{\sim} Uniform(0, 1) \\ &\mathcal{G}^\phi_{i,j} \overset{iid}{\sim} Bernoulli(\mathbf{W}^\phi(v_i, v_j)) \;\; \forall i, j \in [1, N] \end{aligned} \tag{10}$$

where N is the node number and $\mathcal{G}^\phi$ is the augmented metapath reachable graph.

4.2.3 Inter-metapath Augmentation. The core idea of inter-metapath augmentation is to create new synthetic metapaths based on pre-defined metapaths. As graphon models the metapath reachable graph, it can represent the corresponding metapath. Therefore, we directly generate the graphons of new synthetic metapaths to avoid explicitly defining the node types sequence. Concretely, we employ mixup technology to interpolate the graphons of metapath reachable graphs of pre-defined metapaths to generate new graphons. Each augmented graphon represents an implicit synthetic metapath. Then, we sample metapath reachable graphs from the augmented graphons to train heterogeneous graph learning

Table 1: Statistics of datasets.

Datasets	Nodes	Edges	Edge Types	Features	Target	Labels
IMDB	11616	17106	2	3066	Movie	3
ACM	11246	17426	2	1902	Paper	3
Yelp	3913	36066	3	82	Business	3

models. The augmentation process can be denoted as follow:

$$\begin{aligned} &\mathcal{G}^{\phi_i} \longrightarrow \mathbf{W}^{\phi_i}, \mathcal{G}^{\phi_j} \longrightarrow \mathbf{W}^{\phi_j} \\ &\mathbf{W}^* = \lambda \mathbf{W}^{\phi_i} + (1-\lambda)\mathbf{W}^{\phi_j} \\ &\mathcal{G}^* \overset{iid}{\sim} Bernoulli(\mathbf{W}^*) \end{aligned} \tag{11}$$

where $\lambda \sim Uniform(0,1)$ is the trade-off coefficient to control the contributions from different metapaths. $\mathbf{W}^*$ and $\mathcal{G}^*$ are the augmented graphon and metapath reachable graph respectively.

4.2.4 Augmentation. Both of intra-metapath and inter-metapath augmentation generate new synthetic metapath reachable graphs as augmented data. For the heterogeneous graph learning models which require the metapath reachable graph, we directly train models on the synthetic graph. Otherwise, we regard the edges in the synthetic graph as the new types of edges and add them into the original heterogeneous graph, then we train models on the augmented heterogeneous graph.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets. We conduct experiments over three widely used heterogeneous graph datasets. Statistics of them are summarized in Table 1.

- **IMDB**: IMDB is an online database about movies and television programs. We use a subset of IMDB extracted by [8]. It contains nodes in three domains, including 5257 Actors(A), 2081 Directors(D) and 4278 Movies(M). Movies are divided into 3 categories based on their genre information. Each movie is also described by a bag-of-words representation of its plot keywords. We employ {MAM, MDM} as the pre-defined metapath set.
- **ACM**: ACM is a bibliography website. We adopt an ACM graph provided by [37], containing 7176 Authors(A), 4019 Papers(P) and 60 Subjects. Papers are labeled according to their conferences and each paper is described by a bag-of-words representation of their keywords. We employ {PSP, PAP} as the pre-defined metapath set.
- **Yelp**: Yelp is a social network which contains a large amount of user comment data. Here, we adopt the subset of Yelp constructed by [17]. It comprises 2614 businesses(B), 1286 users(U), 4 services(S) and 9 rating levels(L). The businesses are labeled by their categories and the node features are constructed by the bag-of-words representation of the related keywords. We employ {BUB, BSB, BUBLB, BUBSB} as the pre-defined metapath set.

5.1.2 Baselines. As there is no graph augmentation method for heterogeneous graphs, we select six state-of-the-art homogeneous graph augmentation methods as baselines and adapt them to heterogeneous graphs. The details of baselines are as follows:

- **Node-centralized Augmentation**: We select **NodeAug**[30], **NASA**[2] and **LA**[16] as baselines. NodeAug proposes three node augmentation strategies and we only apply these strategies to target nodes which the downstream task pays attention to. NASA replaces the immediate neighbors of nodes with their remote neighbors and we independently conduct above operation on different types of neighbors to achieve augmentation. LA learns the features' distribution of neighbors to generate augmented data and we also use it to augment target nodes.
- **Edge-centralized Augmentation**: We select **DropEdge**[21] and **GAUG**[39] as baselines. DropEdge randomly removes edges to augment the graph and we apply it to all types of edges. For GAUG, we construct multiple edge predictors to independently add or remove different types of edges.
- **Graph-centralized Augmentation**: We select **MH-Aug**[20] as baseline. As MH-Aug directly samples synthetic graphs from the target distribution, we construct different distribution for different types of edges to generate the augmented data.

5.1.3 Backbone models. In order to verify the effectiveness of our augmentation framework in improving the performance of heterogeneous graph learning models, we select six mainstream heterogeneous graph learning models as backbone models: HAN[28], CompGCN[24], MAGNN[8], HGT[12], Simple-HGN[18], HPN[13]. HAN, MAGNN, HGT and HPN explicitly use metapath information while other models do not.

5.1.4 Implementation. For all baselines, we tune the hyperparameters based on the validation set performance. For MAHGA, we employ two-layer-MLPs as the encoder g_ϕ^{enc} and decoder g_θ^{dec} of RCVAE, other model parameters such as the dimensions of relation type embedding $\mathbf{r}$ and latent variable $\mathbf{z}$, the pre-training setting of RCVAE are tuned by grid search. What's more, we utilize the open-source toolkit OpenHGNN[10] to implement the backbone models and the models parameters keep their reported optimal values.

5.2 Performance Comparison

We employ node classification as a downstream task to evaluate the impact of different graph augmentation methods on the performance of backbone models. The results are shown in Table 2. From the table, we make the following observations.

Firstly, although some homogeneous graph augmentation baselines can improve the performance of special backbone models on special heterogeneous datasets, there is no one baseline that can be applied to all backbone models to have a positive impact on all datasets. Especially on ACM dataset, all augmentation methods fail to promote MAGNN but even degrade its performance. The results indicate that homogeneous graph augmentation methods cannot generate the satisfied augmented data to effectively improve the generalization and modeling ability of heterogeneous graph learning models. So, directly adapting homogeneous graph augmentation method to heterogeneous graph is not a good solution and it is necessary to design a specialized augmentation framework for heterogeneous graphs.

Secondly, in most instances, node-centralized augmentation methods perform worse than edge-centralized augmentation methods and graph-centralized augmentation method. Some typical node-centralized augmentation methods such as LA even severely harm

Table 2: Results of node classification with different backbone models and graph augmentation methods on Yelp, ACM and IMDB datasets. The best results are Bold and the second results are Underline. ↑ represents the performance improvement and ↓ represents the performance deterioration comparing with the original backbone model.

Backbone Model	Augmentation	Yelp		ACM		IMDB	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
HAN	–	0.8833(-)	0.8843(-)	0.8900(-)	0.8905(-)	0.5736(-)	0.5633(-)
	NodeAug	0.8908(↑)	0.8975(↑)	0.8840(↓)	0.8843(↓)	0.5684(↓)	0.5546(↓)
	LA	0.8766(↓)	0.8771(↓)	0.8790(↓)	0.8776(↓)	0.5650(↓)	0.5626(↓)
	NASA	0.8913(↑)	0.8973(↑)	0.8960(↑)	0.8976(↑)	0.5791(↑)	0.5733(↑)
	DropEdge	0.8937(↑)	0.9004(↑)	0.8920(↑)	0.8924(↑)	0.5811(↑)	0.5745(↑)
	GAUG	0.8843(↑)	0.8869(↑)	0.9000(↑)	0.9007(↑)	0.5837(↑)	0.5794(↑)
	MH-AUG	0.8893(↑)	0.8962(↑)	0.8980(↑)	0.8952(↑)	0.5784(↑)	0.5717(↑)
	MAHGA	**0.8957**(↑)	**0.9061**(↑)	**0.9180**(↑)	**0.9196**(↑)	**0.5946**(↑)	**0.5934**(↑)
CompGCN	–	0.8997(-)	0.9083(-)	0.7630(-)	0.7227(-)	0.5725(-)	0.5687(-)
	NodeAug	0.8644(↓)	0.8745(↓)	0.7580(↓)	0.7071(↓)	0.5725(-)	0.5718(↑)
	LA	0.8885(↓)	0.8951(↓)	0.7470(↓)	0.7182(↓)	0.5564(↓)	0.5493(↓)
	NASA	0.8838(↓)	0.8853(↓)	0.7780(↑)	0.7493(↑)	0.5687(↓)	0.5655(↓)
	DropEdge	0.8823(↓)	0.8940(↓)	0.7570(↓)	0.7156(↓)	0.5653(↓)	0.5565(↓)
	GAUG	0.9062(↑)	0.9134(↑)	0.7680(↑)	0.7239(↑)	0.5692(↓)	0.5699(↑)
	MH-AUG	0.9084(↑)	0.9162(↑)	0.7680(↑)	0.7251(↑)	0.5704(↓)	0.5671(↓)
	MAHGA	**0.9171**(↑)	**0.9249**(↑)	**0.7910**(↑)	**0.7671**(↑)	**0.5825**(↑)	**0.5810**(↑)
MAGNN	–	0.8977(-)	0.8904(-)	0.8980(-)	0.8967(-)	0.5742(-)	0.5662(-)
	NodeAug	0.8924(↓)	0.8897(↓)	0.8740(↓)	0.8713(↓)	0.5703(↓)	0.5617(↓)
	LA	0.8817(↓)	0.8781(↓)	0.8870(↓)	0.8823(↓)	0.5653(↓)	0.5522(↓)
	NASA	0.8947(↓)	0.8897(↓)	0.8870(↓)	0.8851(↓)	0.5713(↓)	0.5602(↓)
	DropEdge	0.8906(↓)	0.8864(↓)	0.8890(↓)	0.8852(↓)	0.5748(↑)	0.5675(↑)
	GAUG	0.8931(↓)	0.8998(↑)	0.8920(↓)	0.8941(↓)	0.5785(↑)	0.5709(↑)
	MH-AUG	0.8911(↓)	0.8889(↓)	0.8890(↓)	0.8857(↓)	0.5766(↑)	0.5681(↑)
	MAHGA	**0.9091**(↑)	**0.9029**(↑)	**0.9090**(↑)	**0.9093**(↑)	**0.5854**(↑)	**0.5756**(↑)
HGT	–	0.9111(-)	0.9136(-)	0.8810(-)	0.8801(-)	0.5779(-)	0.5757(-)
	NodeAug	0.9017(↓)	0.9050(↓)	0.8720(↓)	0.8730(↓)	0.5476(↓)	0.5483(↓)
	LA	0.9032(↓)	0.9045(↓)	0.8690(↓)	0.8617(↓)	0.5548(↓)	0.5466(↓)
	NASA	0.8987(↓)	0.9015(↓)	0.8610(↓)	0.8581(↓)	0.5710(↓)	0.5669(↓)
	DropEdge	0.8967(↓)	0.9015(↓)	0.8450(↓)	0.8478(↓)	0.5420(↓)	0.5370(↓)
	GAUG	0.9146(↑)	0.9201(↑)	0.8920(↑)	0.8921(↑)	0.5851(↑)	0.5832(↑)
	MH-AUG	0.9056(↓)	0.9081(↓)	0.8770(↓)	0.8795(↓)	0.5715(↓)	0.5681(↓)
	MAHGA	**0.9201**(↑)	**0.9272**(↑)	**0.8950**(↑)	**0.8952**(↑)	**0.5903**(↑)	**0.5875**(↑)
Simple-HGN	–	0.8813(-)	0.8729(-)	0.8810(-)	0.8795(-)	0.5819(-)	0.5771(-)
	NodeAug	0.8208(↓)	0.7831(↓)	0.8770(↓)	0.8753(↓)	0.5822(↑)	0.5694(↓)
	LA	0.8726(↓)	0.8695(↓)	0.8610(↓)	0.8569(↓)	0.5736(↓)	0.5658(↓)
	NASA	0.8595(↓)	0.8339(↓)	0.8650(↓)	0.8679(↓)	0.5874(↑)	0.5852(↑)
	DropEdge	0.8923(↑)	0.8928(↑)	0.8920(↑)	0.8905(↑)	0.5779(↓)	0.5715(↓)
	GAUG	0.8858(↑)	0.8788(↑)	0.8770(↓)	0.8757(↓)	0.5842(↑)	0.5787(↑)
	MH-AUG	0.8781(↓)	0.8712(↓)	0.8770(↓)	0.8739(↓)	0.5796(↓)	0.5691(↓)
	MAHGA	**0.9201**(↑)	**0.9269**(↑)	**0.8920**(↑)	**0.8911**(↑)	**0.5923**(↑)	**0.5905**(↑)
HPN	–	0.9069(-)	0.8983(-)	0.8890(-)	0.8887(-)	0.5940(-)	0.5870(-)
	NodeAug	0.9052(↓)	0.8966(↓)	0.9030(↑)	0.9026(↑)	0.5854(↓)	0.5807(↓)
	LA	0.8952(↓)	0.8839(↓)	0.8770(↓)	0.8752(↓)	0.5765(↓)	0.5721(↓)
	NASA	0.9115(↑)	0.9068(↑)	0.9040(↑)	0.9055(↑)	0.5897(↓)	0.5845(↓)
	DropEdge	0.9136(↑)	0.9056(↑)	0.8980(↑)	0.8984(↑)	0.5909(↓)	0.5882(↑)
	GAUG	0.9111(↑)	0.9043(↑)	0.8970(↑)	0.8962(↑)	0.5937(↓)	0.5881(↑)
	MH-AUG	0.9098(↑)	0.9034(↑)	0.8970(↑)	0.8976(↑)	0.5914(↓)	0.5836(↓)
	MAHGA	**0.9178**(↑)	**0.9122**(↑)	**0.9120**(↑)	**0.9127**(↑)	**0.5998**(↑)	**0.5957**(↑)

the performances of backbone models. We attribute the results to that node-centralized augmentation methods are more fine-grained and they pay more attention to the local information of graphs. However, in heterogeneous graphs, the local information is more abundant than that in homogeneous graphs as different types of nodes have different feature spaces and the local structures of them are also various. Node-centralized homogeneous graph augmentation methods do not fully consider above characteristics of heterogeneous graphs in the augmentation process so that they cannot perform as well as they do on homogeneous graphs.

Finally, our augmentation framework MAHGA significantly and consistently outperforms all baselines on all backbone models and

Table 3: Ablation study on ACM dataset. The best results are Bold. (·) represents the improvement of performance comparing with the original backbone model.

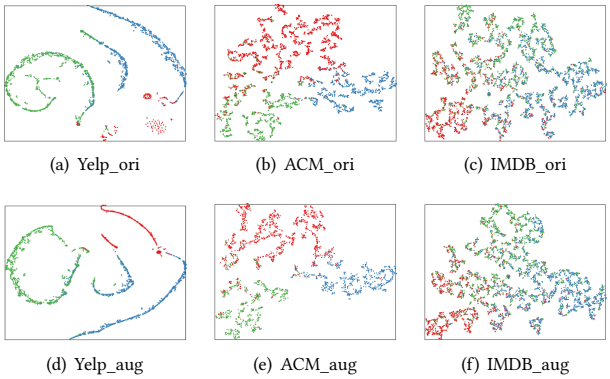
Backbone Model	Augmentation	ACM	
		Micro-F1	Macro-F1
HAN	–	0.8900	0.8905
	+Structure-level	0.9110(+0.021)	0.9126(+0.0221)
	+Intra-Metapath	0.9090(+0.019)	0.9114(+0.0209)
	+Inter-Metapath	0.9150(+0.025)	0.9166(+0.0261)
	MAHGA	**0.9180(+0.028)**	**0.9196(+0.0291)**
CompGCN	–	0.7630	0.7227
	+Structure-level	0.7850(+0.022)	0.7587(+0.0360)
	+Intra-Metapath	0.7690(+0.006)	0.7293(+0.0066)
	+Inter-Metapath	0.7710(+0.008)	0.7323(+0.0096)
	MAHGA	**0.7910(+0.028)**	**0.7671(+0.0444)**
MAGNN	–	0.8980	0.8967
	+Structure-level	0.9070(+0.009)	0.9061(+0.0094)
	+Intra-Metapath	0.9000(+0.002)	0.8980(+0.0013)
	+Inter-Metapath	0.9000(+0.002)	0.8975(+0.0008)
	MAHGA	**0.9090(+0.011)**	**0.9093(+0.0126)**
HGT	–	0.8810	0.8801
	+Structure-level	0.8930(+0.012)	0.8927(+0.0126)
	+Intra-Metapath	0.8840(+0.003)	0.8848(+0.0047)
	+Inter-Metapath	0.8830(+0.002)	0.8821(+0.0020)
	MAHGA	**0.8950(+0.014)**	**0.8952(+0.0151)**
Simple-HGN	–	0.8810	0.8795
	+Structure-level	0.8890(+0.008)	0.8893(+0.0098)
	+Intra-Metapath	0.8870(+0.006)	0.8881(+0.0086)
	+Inter-Metapath	0.8910(+0.010)	0.8901(+0.0106)
	MAHGA	**0.8930(+0.012)**	**0.9005(+0.021)**
HPN	–	0.8890	0.8887
	+Structure-level	0.9040(+0.015)	0.9054(+0.0167)
	+Intra-Metapath	0.9090(+0.020)	0.9103(+0.0216)
	+Inter-Metapath	0.9120(+0.023)	0.9127(+0.0240)
	MAHGA	**0.9170(+0.028)**	**0.9187(+0.0300)**

datasets. In general, MAHGA achieves relative performance gains over original backbone models by 0.58–3.88% in terms of Micro-F1 and 0.87–5.4% in terms of Macro-F1, which proves the effectiveness and superiority of MAHGA. By designing sophisticated augmentation strategies from network schema aspect and metapath aspect, MAHGA introduces the heterogeneity and semantics of heterogeneous graphs into augmentation process, which help to generate better augmented data to improve the performances of heterogeneous graph learning models.

5.3 Ablation Study

In order to verify the effectiveness of different augmentation strategies of MAHGA, we design three variants of MAHGA to conduct the ablation experiments. Only one augmentation strategy is available for each variant. For example, "+Structure-level" means that this variant of MAHGA only employs the structure-level augmentation strategy in Section 4.1. The experimental results are shown in Table 3. Due to the limitation of space, we only display the results on ACM dataset. Other results are similar so we omit them.

As we can observe, all augmentation strategies of MAHGA have a positive impact and they improve the performances of backbone models. However, for different backbone models and datasets, the effects of different augmentation strategies vary. For example, the performance improvements of MAGNN and HGT by metapath-level augmentation are not as significant as that by structure-level augmentation, while metapath-level augmentation perform better than structure-level augmentation on HAN and HPN. It is because MAGNN and HGT require complete path information including the intermediate nodes along the metapath, but HAN and HPN explicitly utilize the metapath based neighbors. Despite all this, MAHGA still outperforms all variants which indicates that different augmentation strategies augment heterogeneous graph from different aspects and combining them together can further improve the augmentation effect.



(a) Yelp_ori (b) ACM_ori (c) IMDB_ori

(d) Yelp_aug (e) ACM_aug (f) IMDB_aug

Figure 3: 2D visualization of node representation of HAN on three datasets using t-SNE. The different colors represent different classes.

5.4 Visualization

For a more intuitive comparison and to further show the effectiveness of our proposed augmentation framework, we select HAN as backbone model and conduct the task of visualization on three datasets. Here, we utilize t-SNE[19] to project the node representation learned by HAN into a 2-dimensional space. The experimental results are depicted in Figure 3. "Yelp_ori" means that we directly train HAN on Yelp dataset, while "Yelp_aug" means that we employ MAHGA in the training of HAN. From the visualization, we can see that the learned node representations have the higher intra-class similarity and the clearer distinct boundary among different classes when we adopt MAHGA to train HAN. It is a solid evidence to prove that MAHGA can enhance the backbone models well.

6 CONCLUSION

In this paper, we study the data augmentation on heterogeneous graphs, where node types and link types are diverse. Unfortunately, existing graph augmentation methods are only designed for homogeneous graphs and they cannot perform well on heterogeneous graphs because they ignore the type information and rich semantics in the latter. Therefore, we propose MAHGA, a novel heterogeneous graph augmentation framework to address the above issues. MAHGA designs augmentation strategies from network schema and metapath aspects to fully consider the heterogeneity and semantics. Extensive experiments on three common heterogeneous datasets and six popular heterogeneous graph learning models demonstrate the rationality and effectiveness of MAHGA.

REFERENCES

[1] Edoardo M. Airoldi, Thiago B. Costa, and Stanley H. Chan. 2013. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *27th Annual Conference on Neural Information Processing Systems 2013*. 692–700.

[2] Deyu Bo, Binbin Hu, Xiao Wang, Zhiqiang Zhang, Chuan Shi, and Jun Zhou. 2022. Regularizing Graph Neural Networks via Consistency-Diversity Graph Augmentations. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*. 3913–3921.

[3] Mihaela A. Bornea, Lin Pan, Sara Rosenthal, Radu Florian, and Avirup Sil. 2021. Multilingual Transfer Learning for QA using Translation as Data Augmentation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*. 12583–12591.

[4] Stanley H. Chan and Edoardo M. Airoldi. 2014. A Consistent Histogram Estimator for Exchangeable Graph Models. In *Proceedings of the 31th International Conference on Machine Learning*. 208–216.

[5] Sourav Chatterjee. 2015. Matrix Estimation by Universal Singular Value Thresholding. *The Annals of Statistics* 43, 1, 177–214.

[6] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M. Nasrabadi. 2021. SuperMix: Supervising the Mixing Data Augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 13794–13803.

[7] Alan M. Frieze and Ravi Kannan. 1999. Quick Approximation to Matrices and Applications. *Comb.* 19, 2 (1999), 175–220.

[8] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *The World Wide Web Conference*. 2331–2341.

[9] Hang Gao, Jiangmeng Li, Wenwen Qiang, Lingyu Si, Fuchun Sun, and Changwen Zheng. 2022. Bootstrapping Informative Graph Augmentation via A Meta Learning Approach. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. 3001–3007.

[10] Hui Han, Tianyu Zhao, Cheng Yang, Hongyi Zhang, Yaoqi Liu, Xiao Wang, and Chuan Shi. 2022. OpenHGNN: An Open Source Toolkit for Heterogeneous Graph Neural Network. In *Proceedings of the 31th ACM International Conference on Information and Knowledge Management*.

[11] Nicklas Hansen, Hao Su, and Xiaolong Wang. 2021. Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*. 3680–3693.

[12] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *The World Wide Web Conference*. 2704–2710.

[13] Houye Ji, Xiao Wang, Chuan Shi, Bai Wang, and Philip S. Yu. 2021. Heterogeneous Graph Propagation Network. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[14] Hazel H. Kim, Daecheol Woo, Seong Joon Oh, Jeong-Won Cha, and Yo-Sub Han. 2022. ALP: Data Augmentation Using Lexicalized PCFGs for Few-Shot Text Classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*. 10894–10902.

[15] Jian Liu, Yufeng Chen, and Jinan Xu. 2022. Low-Resource NER by Data Augmentation With Prompting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. 4252–4258.

[16] Songtao Liu, Rex Ying, Hanze Dong, Lanqing Li, Tingyang Xu, Yu Rong, Peilin Zhao, Junzhou Huang, and Dinghao Wu. 2022. Local Augmentation for Graph Neural Networks. In *International Conference on Machine Learning*. 14054–14072.

[17] Yuanfu Lu, Chuan Shi, Linmei Hu, and Zhiyuan Liu. 2019. Relation Structure-Aware Heterogeneous Information Network Embedding. In *The Thirty-Third AAAI Conference on Artificial Intelligence*. 4456–4463.

[18] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress?: Revisiting, benchmarking and refining heterogeneous graph neural networks. In *The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1150–1160.

[19] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[20] Hyeon-Jin Park, Seunghun Lee, Sihyeon Kim, Jinyoung Park, Jisu Jeong, Kyung-Min Kim, Jung-Woo Ha, and Hyunwoo J. Kim. 2021. Metropolis-Hastings Data Augmentation for Graph Neural Networks. In *Annual Conference on Neural Information Processing Systems 2021*. 19010–19020.

[21] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *8th International Conference on Learning Representations*.

[22] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Annual Conference on Neural Information Processing Systems 2015*. 3483–3491.

[23] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. 2021. Adversarial Graph Augmentation to Improve Graph Contrastive Learning. In *Annual Conference on Neural Information Processing Systems 2021*. 15920–15933.

[24] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *8th International Conference on Learning Representations*.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Annual Conference on Neural Information Processing Systems 2017*. 5998–6008.

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

[27] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In *Proceedings of the 36th International Conference on Machine Learning*. 6438–6447.

[28] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference*. 2022–2032.

[29] Yiwei Wang, Yujun Cai, Yuxuan Liang, Henghui Ding, Changhu Wang, Siddharth Bhatia, and Bryan Hooi. 2021. Adaptive Data Augmentation on Temporal Graphs. In *Annual Conference on Neural Information Processing Systems 2021*. 1440–1452.

[30] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, Juncheng Liu, and Bryan Hooi. 2021. NodeAug: Semi-Supervised Node Classification with Data Augmentation. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 207–217.

[31] Zongqian Wu, Peng Zhou, Guoqiu Wen, Yingying Wan, Junbo Ma, Debo Cheng, and Xiaofeng Zhu. 2022. Information Augmentation for Few-shot Node Classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. 3601–3607.

[32] Hongteng Xu, Dixin Luo, Lawrence Carin, and Hongyuan Zha. 2021. Learning Graphons via Structured Gromov-Wasserstein Barycenters. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*. 10505–10513.

[33] Denis Yarats, Ilya Kostrikov, and Rob Fergus. 2021. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. In *9th International Conference on Learning Representations*.

[34] Jaemin Yoo, Sooyeon Shim, and U Kang. 2022. Model-Agnostic Augmentation for Accurate Graph Classification. In *The ACM Web Conference 2022*. 1281–1291.

[35] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations*.

[36] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. 2021. How Does Mixup Help With Robustness and Generalization?. In *9th International Conference on Learning Representations*.

[37] Jianan Zhao, Xiao Wang, Chuan Shi, Zekuan Liu, and Yanfang Ye. 2020. Network Schema Preserving Heterogeneous Information Network Embedding. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 1366–1372.

[38] Tong Zhao, Gang Liu, Daheng Wang, Wenhao Yu, and Meng Jiang. 2022. Learning from Counterfactual Links for Link Prediction. In *International Conference on Machine Learning*. 26911–26926.

[39] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver J. Woodford, Meng Jiang, and Neil Shah. 2021. Data Augmentation for Graph Neural Networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*. 11015–11023.

A APPENDIX

We give the detailed derivation of Equation (1) and Equation (2).

A.1 Proof of Equation (1)

$$
\begin{aligned}
log p_\theta(\mathbf{X}_u|\mathbf{X}_v,\mathbf{r}_{uv}) &= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log p_\theta(\mathbf{X}_u|\mathbf{X}_v,\mathbf{r}_{uv}) d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})}{p_\theta(\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) \cdot p_\theta(\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv},\mathbf{z})}{p_\theta(\mathbf{X}_v,\mathbf{r}_{uv}) \cdot p_\theta(\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv},\mathbf{z})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{p_\theta(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv}) \cdot q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})}{p_\theta(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) \cdot q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) \left(log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} + log \frac{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})}{p_\theta(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} \right) d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} + KL(q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})||p_\theta(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})) \\
&\geq \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z}
\end{aligned}
$$

A.2 Proof of Equation (2)

$$
\begin{aligned}
L_{ELBO}(\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv};\theta,\phi) &= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u,\mathbf{z},\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) \cdot p_\theta(\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u|\mathbf{z},\mathbf{X}_v,\mathbf{r}_{uv}) \cdot p_\theta(\mathbf{z},\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) \cdot p_\theta(\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{X}_u|\mathbf{z},\mathbf{X}_v,\mathbf{r}_{uv}) \cdot p_\theta(\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log \frac{p_\theta(\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})}{q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})} d\mathbf{z} + \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log p_\theta(\mathbf{X}_u|\mathbf{z},\mathbf{X}_v,\mathbf{r}_{uv}) d\mathbf{z} \\
&= -KL(q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})||p_\theta(\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})) + \int q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv}) log p_\theta(\mathbf{X}_u|\mathbf{z},\mathbf{X}_v,\mathbf{r}_{uv}) d\mathbf{z} \\
&= -KL(q_\phi(\mathbf{z}|\mathbf{X}_u,\mathbf{X}_v,\mathbf{r}_{uv})||p_\theta(\mathbf{z}|\mathbf{X}_v,\mathbf{r}_{uv})) + \frac{1}{N^v} \sum_{n=1}^{N^v} log p_\theta(\mathbf{X}_u|\mathbf{z}^n,\mathbf{X}_v,\mathbf{r}_{uv})
\end{aligned}
$$