R-GSN: The Relation-based Graph Similar Network for Heterogeneous Graph

Anonymous authors
Paper under double-blind review

Abstract

Heterogeneous graph is a kind of data structure widely existing in real life. Nowadays, the research of graph neural network on heterogeneous graph has become more and more popular. The existing heterogeneous graph neural network algorithms mainly have two ideas, one is based on meta-path and the other is not. The idea based on meta-path often requires a lot of manual preprocessing, at the same time it is difficult to extend to large scale graphs. In this paper, we proposed the general heterogeneous message passing paradigm and designed R-GSN that does not need meta-path, which is much improved compared to the baseline R-GCN. Experiments have shown that our R-GSN algorithm achieves the state-of-the-art performance on the `ogbn-mag` large scale heterogeneous graph dataset.

1 Introduction

In practical applications, many data can be regarded as graphs, such as social networks, academic networks and knowledge graphs. There are three types of common tasks on the graph, namely the node classification, the link prediction and the graph classification. In recent years, due to the numerous application scenarios and challenges of graphs, the research of graph neural networks has become more and more popular, and many classic algorithms on the homogeneous graphs have been proposed, such as ChebNet (Defferrard et al., 2016), GCN (Kipf & Welling, 2016), Graph-SAGE (Hamilton et al., 2017), GAT (Veličković et al., 2017), GIN (Xu et al., 2018), DeepGCNs (Li et al., 2019) and GeomGCN (Pei et al., 2020).

But obviously in real scenes many graphs are heterogeneous. Therefore, researchers began to pay more attention to neural networks on heterogeneous graphs. R-GCN (Schlichtkrull et al., 2018) is a heterogeneous graph neural network designed for knowledge graphs, which solved the tasks of entity classification and link prediction on the knowledge graphs; note that R-GCN aggregates the first-order neighborhoods. In heterogeneous graphs, by predefining meta-path (Dong et al., 2017), the semantic high-order neighbors of the target node can be found and utilized. HAN (Wang et al., 2019) used meta-path to aggregate high-order neighbors and introduced attention mechanism. MAGNN is similar to HAN, the difference is that the features of all the nodes on the meta-path have been used in MAGNN (Fu et al., 2020). GTN (Yun et al., 2019) is a novel model that can automatically learn the meta-path without manually predefining it in advance. HGT (Hu et al., 2020b) got rid of the meta-path, adopted meta-relation, and designed an aggregation method similar to Transformer (Vaswani et al., 2017).

Since most of the existing algorithms are based on the meta-path, they require a lot of manual processing and have poor generalization. For this we propose an end-to-end R-GSN algorithm based on R-GCN, which does not need meta-path and can be easily applied to large scale heterogeneous graph datasets, such as `ogbn-mag` (Hu et al., 2020a). Our contributions are as follows:

- We propose a general heterogeneous message passing paradigm. Under this framework, heterogeneous message passing is summarized into the following four steps: Message Transform, Intra-relation Aggregation, Inter-relation Aggregation and Status Update. R-GCN (Schlichtkrull et al., 2018) is a special instance of this paradigm.

- We propose the R-GSN algorithm, which conforms to the general heterogeneous message passing paradigm, and we design two novel similarity aggregation methods, called the

SIM-ATTN aggregation and the SIM aggregation; note that the SIM-ATTN aggregation includes an attention mechanism while the SIM aggregation does not.

- In R-GSN, we use normalization operations in several places. Experiments prove the necessity of normalization operations. there are two other operations that are also useful, one is the adversarial training method called FLAG (Kong et al., 2020), and the other is specifically for the `ogbn-mag` dataset in generating parameterized initial features for nodes without initial features through pre-propagation.

- R-GSN achieves the state-of-the-art performance on the `ogbn-mag` dataset. The average accuracy on the validation dataset and test dataset are respectivly 51.82% and 50.32%, both are higher than those of the existing algorithms. At the same time, compared with the baseline R-GCN, there is a 4.21% gain on the validation dataset and a 3.54% gain on the test dataset.

- We extend R-GSN and propose a homogeneous version algorithm called GSN and a meta-path version algorithm called M-GSN. Respectively, we conducte experiments on `ogbn-arxiv`, `IMDB` and `DBLP` datasets, and prove the effectiveness of the aggregation method proposed in this paper.
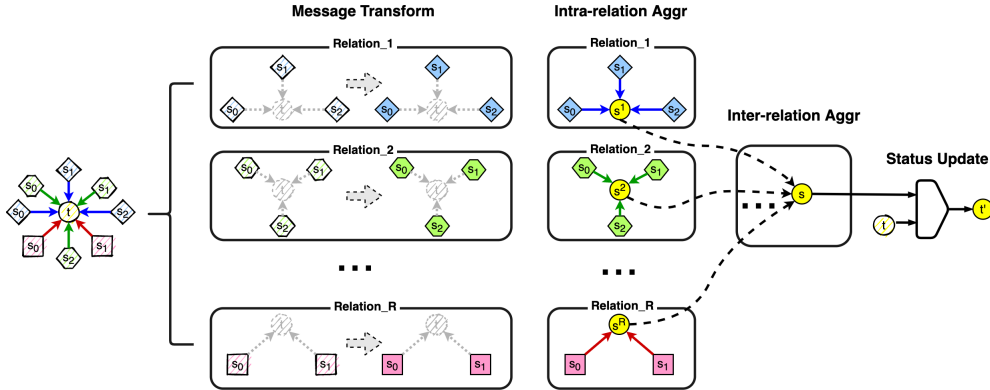

Figure 1: General heterogeneous message passing paradigm. It concludes the following four steps. Message Transform, Intra-relation Aggregation, Inter-relation Aggregation and Status Update.

2 PROPOSED METHOD: R-GSN

2.1 GENERAL HETEROGENEOUS MESSAGE PASSING PARADIGM

Before introducing our R-GSN algorithm, in this part we first define a general paradigm for heterogeneous graph neural network message passing(see Figure 1). Both R-GCN and our R-GSN are special instances under this paradigm. Heterogeneous message passing is summarized into the following four steps. **Message Transform , Intra-relation Aggregation, Inter-relation Aggregation and Status Update.** Now we will take R-GCN as an example to analyze the detailed operation of each component under this paradigm.

(1) Message Transform

Message transform is the first step in each layer of heterogeneous message passing. For R-GCN, it is to assign the relation-specific parameter matrix W_{rel}^r to each relation r in the relation set $\mathcal{R}$. for the target node t, all neighbor nodes under any relation r will be multiplied by the relation-specific parameter matrix W_{rel}^r(1).

$$\widetilde{h_{s_i^r}} = W_{rel}^r h_{s_i^r} \tag{1}$$

the relation-specific parameter matrix can transform the dimension of the feature(status). when $h_{s_i^r} \in \mathbb{R}^d$ and $W_{rel}^r \in \mathbb{R}^{d' \times d}$, we can get $\widetilde{h_{s_i^r}} \in \mathbb{R}^{d'}$. In this way, the generated message feature to be aggregated are obtained.

(2) Intra-relation Aggregation

In heterogeneous graph, for the target node t, there may be multiple types of edges connected to t, and we will separately aggregate the neighbor nodes under each relation r. The core of intra-relation

aggregation is how to perform feature aggregation for a certain type of edge. For R-GCN, we can treat its intra-relation aggregation as **MEAN aggregation**(2).

$$h_s^r = \frac{1}{|\mathcal{N}_t^r|} \sum_{s_i^r \in \mathcal{N}_t^r} \widetilde{h_{s_i^r}} \tag{2}$$

where $\widetilde{h_{s_i^r}}$ is the transformed feature generated by (1), $\mathcal{N}_t^r$ is the set of the neighbor nodes of target node t under the relation r. the result of Mean aggregation is h_s^r, which represents the neighborhood aggregation feature of the target node t under the relation r.

(3) Inter-relation Aggregation

When the intra-relation aggregation is finished, we will get the aggregation feature h_s^r of the target node t under each relation. After that, we need to aggregate the features obtained under different relation again, which is the core operation of inter-relation. For R-GCN, its inter-relation aggregation is **SUM aggregation**(3).

$$h_s = \sum_{r \in \mathcal{R}} h_s^r \tag{3}$$

Intuitively, the aggregated features under different relations are aggregated again through the addition operation, and the final aggregated feature h_s of all neighboring nodes of the target node t is obtained.
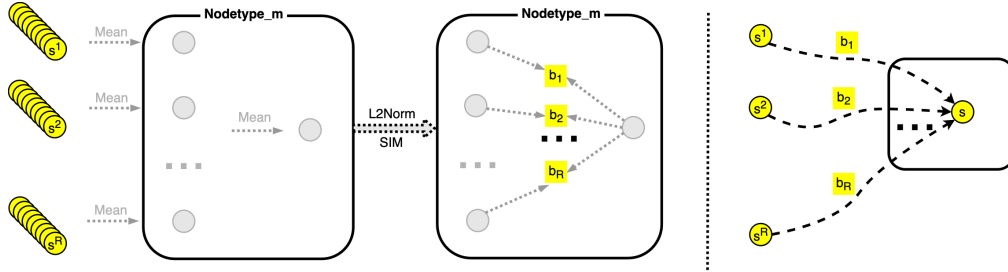

Figure 2: R-GSN Inter-realtion Aggregation(SIM aggregation).

(4) Status Update

Status update is the last step. We need to merge the feature h_t of the target node t with the feature h_s aggregated from all neighboring nodes together to obtain the final updated feature of the target node t, which is the output status h_t'(4).

$$h_t' = \sigma(h_s + W_{node}h_t) \tag{4}$$

where W_{node} is the node transform parameter matrix. σ is an element-wise activation function, such as ReLU($\cdot$).

We find that equation (19) is equivalent to concatenating equation (1)(2)(3) and (4). We can easily decompose and analyze R-GCN according to the general paradigm defined above.

2.2 RELATION-BASED GRAPH SIMILAR NETWORK(R-GSN)

The R-GSN we proposed is a structure that conforms to the general heterogeneous message passing paradigm. it is based on R-GCN, further introduces attention mechanism and similarity aggregation, as well as various normalizations. Next, we will continue to decompose the R-GSN algorithm according to the general paradigm.

(1) Message Transform

The message transformation of R-GSN is the same as that of R-GCN. it is to assign the relation-specific parameter matrix W_{rel}^r to each relation r in the relation set $\mathcal{R}$. For the target node t, all neighbors under any relation r will be multiplied by the relation-specific parameter matrix W_{rel}^r(5).

$$\widetilde{h_{s_i^r}} = W_{rel}^r h_{s_i^r} \tag{5}$$

where $h_{s_i^r} \in \mathbb{R}^d$ and $W_{rel}^r \in \mathbb{R}^{d' \times d}$, so $\widetilde{h_{s_i^r}} \in \mathbb{R}^{d'}$. In this way, the generated message feature to be aggregated are obtained.

(2) Intra-relation Aggregation

The intra-relation aggregation of R-GCN is MEAN aggregation described in equation (2). In R-GSN, the intra-relation aggregation is **SIM-ATTN aggregation**(similarity-attention aggregation), which can also be regarded as a coefficient-weighted aggregation. As described in Figure 6 in subsubsection A.2 and equation (6).

$$h_s^r = \text{L2Norm}(\sum_{s_i^r \in \mathcal{N}_t^r} a_i^r \widetilde{h_{s_i^r}}) \tag{6}$$

where L2Norm is L2 normalization $\left(\text{L2Norm}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}\right)$. $\widetilde{h_{s_i^r}}$ is the transformed feature generated by equation (5). the result of SIM-ATTN aggregation is h_s^r, which represents the neighborhood aggregation feature of the target node t under the relation r, and $h_s^r \in \mathbb{R}^{d'}$. The difference from R-GCN's MEAN aggregation is that there is an additional weighted coefficient a_i^r, which can be calculated by the following equation (7)(8)(9)(10).

$$\ddot{h}_{s_i^r} = \text{L2Norm}(h_{s_i^r}) \tag{7}$$

$$\ddot{h}_{s^r} = \text{L2Norm}(\frac{1}{|\mathcal{N}_t^r|} \sum_{s_i^r \in \mathcal{N}_t^r} h_{s_i^r}) \tag{8}$$

$$e_i^r = attn^r * [\ddot{h}_{s_i^r} || \ddot{h}_{s^r}] \tag{9}$$

$$a_i^r = \text{softmax}(e_i^r) = \frac{e_i^r}{\sum_{j \in \mathcal{N}_t^r} e_j^r} \tag{10}$$

where $h_{s_i^r} \in \mathbb{R}^d$, so $\ddot{h}_{s_i^r} \in \mathbb{R}^d$ and $\ddot{h}_{s^r} \in \mathbb{R}^d$, symbol $||$ means concatenation, so $[\ddot{h}_{s_i^r} || \ddot{h}_{s^r}] \in \mathbb{R}^{2d}$, $attn^r \in \mathbb{R}^{2d}$ is the trainable attention mechanism parameters which is also relation-specific, the $*$ here represents the inner product. When e_i^r is calculated, the final weighted coefficient a_i^r can be obtained through the neighborhood softmax on e_i^r.

Note that our SIM-ATTN aggregation method is a little similar to GAT's attention aggregation(11)(12) (Veličković et al., 2017), but there is still a big difference. For example, in the process of calculating the weighted coefficient, only the feature of the source node is used, and the feature of the target node is not used. Before similarity calculation, we first performed the L2 normalization operation in order to make the features concatnate at the same scale. Meanwhile in SIM-ATTN aggregation, the Leakey ReLU operation is abandoned.

$$e_i = \text{LeakyReLU}(attn * [Wh_{s_i} || Wh_t]) \tag{11}$$

$$a_i = \text{softmax}(e_i) = \frac{e_i^r}{\sum_{j \in \mathcal{N}_t^r} e_j^r} \tag{12}$$

(3) Inter-relation Aggregation

As mentioned above, inter-ralation aggregation will aggregate the feature h_s^r of the target node t under all relations. In R-GCN, SUM aggregation is used. In R-GSN, we propose to use **SIM aggregation**, which is somewhat similar to SIM-ATTN aggregation. The only difference is that no attention vector is introduced here, so there are no parameters. We can use equation (13) to describe SIM aggregation (Figure 2).

$$h_s = \sum_{r \in \mathcal{R}} b_r^m h_s^r \tag{13}$$

Here $h_s \in \mathbb{R}^{d'}$. The difference between R-GSN(13) and R-GCN(3)'s inter-relation aggregation is b_r^m. b_r^m is also a similarity weighted coefficient, which represents the importance weight that the nodetype m should be given under the relation r. Of course, the premise is that assuming the nodetype of the target node t is m. $m \in \mathcal{M}$ and $\mathcal{M}$ is a collection of node types. b_r^m can be

calculated by the following equation (14)(15)(16)(17).

$$h_{s^r}^{\dddot{m}} = \text{L2Norm}(\frac{1}{|\mathcal{V}^m|} \sum_{t_j \in \mathcal{V}^m} h_{s^r}(t_j)) \tag{14}$$

$$h_s^{\dddot{m}} = \text{L2Norm}(\frac{1}{|\mathcal{R}| \times |\mathcal{V}^m|} \sum_{r \in \mathcal{R}} \sum_{t_j \in \mathcal{V}^m} h_{s^r}(t_j)) \tag{15}$$

$$e_r^m = h_{s^r}^{\dddot{m}} * h_s^{\dddot{m}} \tag{16}$$

$$b_r^m = \text{softmax}(e_r^m) = \frac{e_r^m}{\sum_{j \in \mathcal{R}} e_j^m} \tag{17}$$

We mentioned that h_s^r is the neighborhood aggregated feature of the target node t under the relation r, We can also write h_s^r as a verbose form $h_s^r(t)$. In equation (14), $\mathcal{V}^m$ represents the collection of all nodes of nodetype m. Where $h_{s^r}(t_j) \in \mathbb{R}^d$, so $h_{s^r}^{\dddot{m}} \in \mathbb{R}^d$ and $h_s^{\dddot{m}} \in \mathbb{R}^d$, the $*$ here represents the inner product. When e_r^m is calculated, the final weighted coefficient b_r^m can be obtained through the softmax on e_r^m across all relations.
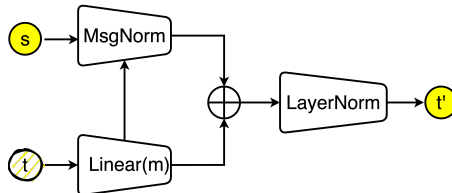
(4) Status Update

Figure 3: R-GSN Status Update.

For R-GSN, the process of State Update is described in equation (18) and Figure 3.

$$h_t' = \sigma(\text{LayerNorm}(\text{MsgNorm}(h_s, W_{node}^m h_t) + W_{node}^m h_t)) \tag{18}$$

where W_{node}^m is nodetype-specific parameter matrix for target node t of nodetype m, $W_{node}^m \in \mathbb{R}^{d' \times d}$. MsgNorm is proposed by Li et al. (2020), its core idea is to normalize the message feature first, and then multiply it by the norm of the target node feature. In R-GSN, we design a normalization layer that cascades MsgNorm and LayerNorm (Ba et al., 2016), which can integrate message features and node features well. σ is an element-wise activation function, such as ReLU($\cdot$). h_t' is the updated feature of the target node t and $h_t' \in \mathbb{R}^{d'}$.

3 EXPERMENTS

3.1 HETEROGENEOUS GRAPH DATASET: OGBN-MAG

The `ogbn-mag` dataset is a heterogeneous network (Wang et al., 2020). It contains four types of entitiesłpapers (736,389 nodes), authors (1,134,649 nodes), institutions (8,740 nodes), and fields of study (59,965 nodes) as well as four types of directed relations connecting two types of entitiesłan author is affiliated with an institution, an author writes a paper, a paper cites a paper, and a paper has a topic of a field of study. each paper is associated with a 128-dimensional word2vec feature vector, and all the other types of entities are not associated with input node features. The task on the heterogeneous `ogbn-mag` dataset is to predict the venue (conference or journal) of each paper, given its content, references, authors, and authors affiliations. This is of practical interest as some manuscripts venue information is unknown or missing in MAG, due to the noisy nature of Web data. In total, there are 349 different venues in `ogbn-mag`, making the task a 349-class classification problem.

3.2 NODE CLASSIFICATION

3.2.1 R-GSN IMPLEMENTATION DETAILS

In R-GSN, the implementation details of each heterogeneous message passing layer have been described in section 2.2. Our R-GSN model has two layers, the feature dimension of the hidden layer is

Table 1: Results for `ogbn-mag`.

Method	Accuracy (%)		
	Params	Validation	Test
MLP	188,509	26.26±0.16	26.92±0.26
GCN	1,495,901	29.53±0.22	30.43±0.25
GRAPHSAGE	1,495,901	30.70±0.19	31.53±0.15
METAPATH2VEC	94,479,069	35.06±0.17	35.44±0.36
CLUSTERGCN	154,366,772	38.40±0.31	37.32±0.37
SIGN	3,724,645	40.68±0.10	40.46±0.12
R-GCN	154,366,772	47.61±0.68	46.78±0.67
GRAPHSAINT	154,366,772	48.37±0.26	47.51±0.22
HGT	21,173,389	49.89±0.47	49.27±0.61
R-GSN	154,373,028	**51.82**±0.41	**50.32**±0.37

Table 2: Ablation study of R-GSN on `ogbn-mag`.

Method	SIM-ATTN	SIM	Norm	FT	FLAG	Accuracy (%)		
						Params	Validation	Test
R-GCN	–	–	–	–	–	154,366,772	47.61±0.68	46.78±0.67
R-GCN(1+)	–	–	✓	–	–	154,370,340	49.86±0.29	48.83±0.50
R-GCN(2+)	–	–	✓	✓	–	154,370,340	50.72±0.27	49.66±0.37
R-GCN(3+)	–	–	✓	✓	✓	154,370,340	51.04±0.48	49.74±0.47
R-GSN(1+)	✓	✓	✓	–	–	154,373,028	50.43±0.46	49.26±0.40
R-GSN(2+)	✓	✓	✓	✓	–	154,373,028	51.33±0.35	50.10±0.42
R-GSN(3+)	✓	✓	✓	✓	✓	154,373,028	**51.82**±0.41	**50.32**±0.37

64, and the feature dimension of the output layer is 349 (the number of categories of the `ogbn-mag` dataset). We use cross entropy loss and Adamw (Loshchilov & Hutter, 2018) optimizer for training, the learning rate is set to 0.004, and use early-stop technology and Dropout (Srivastava et al., 2014) technology to prevent overfitting. We use the minibatch neighborhood sampling method for training (Hamilton et al., 2017), the batchsize is 1024. For the original feature, we set the feature to be trainable and use FLAG (Kong et al., 2020) adversarial training. The division of train/val/test is in accordance with the official division. Table 1 reports the accuracy of the validation and test dataset of our R-GSN on the `ogbn-mag`(Note that the report is the mean and standard deviation of the results of 10 random runs), the results of other algorithms are all from the public results on the official leaderboard (`https://ogb.stanford.edu/docs/leader_nodeprop/`).

Because `ogbn-mag` is a heterogeneous graph, some changes need to be made to migrate the GNN algorithm on a homogeneous graph. Specifically, for GCN (Kipf & Welling, 2016) and Graph-SAGE (Hamilton et al., 2017), since they were originally designed for homogeneous graphs, apply the model to homogeneous subgraphs, keeping only the paper nodes and their citation relations. For MLP, the graph structure is ignored and only paper nodes are considered. Except for these three algorithms, the others are directly designed for heterogeneous graphs.

Obviously, R-GSN has a powerful improvement over R-GCN (Schlichtkrull et al., 2018), and its performance is better than the previous best algorithm HGT (Hu et al., 2020b). R-GSN achieves the state-of-the-art performance on the `ogbn-mag` dataset. The average accuracy on the validation dataset and test dataset are respectivly 51.82% and 50.32%, both are higher than those of the existing algorithms. At the same time, compared with the baseline R-GCN, there is a 4.21% gain on the validation dataset and a 3.54% gain on the test dataset. Compared with HGT, there is a 1.93% gain on the validation dataset and a 1.05% gain on the test dataset.
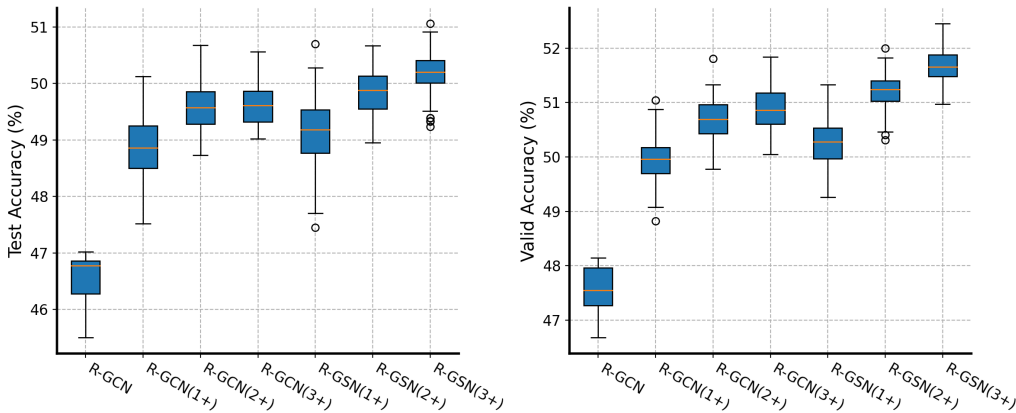
Figure 4: Ablation study. Left: Test accuracy on the `ogbn-mag` dataset. Right: Valid accuracy on the `ogbn-mag` dataset.
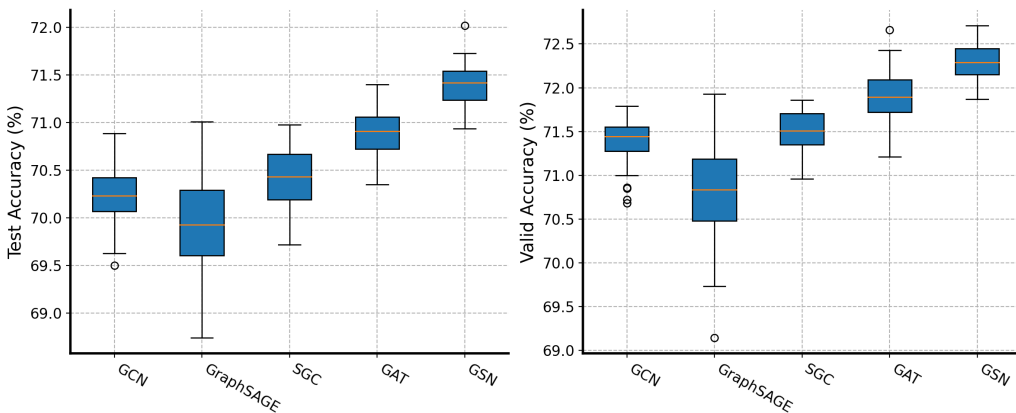

Figure 5: Results on the `ogbn-arxiv` dataset. Left: Test accuracy. Right: Valid accuracy.

3.3 ABLATION STUDY

In this part, we show ablation experiments to analyze which operations boost the performance of R-GCN and R-GSN, The results of ablation experiments are shown in Table 2. **SIM-ATTN**: the method used by R-GSN Intra-relation aggregation. **SIM**: the method used by R-GSN Inter-relation aggregation. **Norm**: Normalization is used in 4 places, namely LayerNorm at the input, L2Norm at the Intra-relation aggregation, MsgNorm and LayerNorm at Status Update. **FT**: the abbreviation of feature training, because the original heterogeneous graph data only has the features of the paper node, so we first perform feature pre-propagation according to the connection relation to obtain the initial features of other types of nodes, and set these features as trainable parameter. **FLAG**: It is an adversarial training method that improves the robustness of the trained model by perturbing the original data.

In Table 2, the result of R-GCN is the result reported on the official website, and we use it as the baseline model. R-GCN(1+), R-GCN(2+) and R-GCN(3+) are R-GCN with one, two and three operations added, namely Norm, FT, FLAG . We can find that when three operations are successively added to R-GCN and R-GSN, the accuracy of the validation and test dataset will increase accordingly. Therefore, these three operations all have the positive effect. Of course, by comparing R-GCN(1+) and R-GSN(1+), R-GCN(2+) and R-GSN(2+), R-GCN(3+) and R-GSN(3+), it can be found that R-GSN with SIM-ATTN aggregation and SIM aggregation has better performance than R-GCN with MEAN aggregation and SUM aggregation. Figure 4 shows more intuitively in Table 2. The boxplot of the accuracy distribution of the 7 models after multiple random trainings (more than 10 times).

3.4 HOMOGENEOUS GRAPH VERSION OF R-GSN: GSN

We hope that the aggregation method proposed in R-GSN and some tricks should also be beneficial to homogeneous graphs. To this end, we conducted experiments on the paper citation network: `ogbn-arxiv` dataset, which is significantly larger than the Cora, and PubMed datasets, and is more in line with real-world applications. Small datasets often have only hundreds of thousands of edges, while the `ogbn-arxiv` has 169,343 nodes and 1,162,243 edges, with an average degree of 13.7. The `ogbn-arxiv` dataset is a directed graph, which is a citation network between ARXIV papers. Each node is a paper published on ARXIV, and each directed edge represents the paper citation relationship between the two papers. Each paper has a 128-dimensional initial feature vector. The Word2vec model is run on the MAG corpus to calculate the embedding of a single word. The initial feature vector is obtained by averaging the embeddings of the words in its title and abstract. The task is to predict 40 subject areas of computer science papers on ARXIV, such as artificial intelligence, operating systems, and etc., so this is a multi-classification task with 40 categories.

We designed a homogenous version of R-GSN, called GSN. It can be regarded as the R-GSN with only one type of node and one type of edge. We conducted a simple experiment on the `ogbn-arxiv` dataset and compared it with several classic graph neural network algorithms. The experimental results are shown in Table 3, which reported the mean and standard deviation of the accuracy after 10 random trainings.

Table 3: Results for `ogbn-arxiv`.

Method		Accuracy (%)	
	Params	Validation	Test
MLP	38,696	54.22±0.12	56.25±0.12
GCN	38,184	70.24±0.28	71.39±0.22
GRAPHSAGE	76,072	69.90±0.49	70.81±0.50
SGC	38,184	70.43±0.30	71.51±0.23
GAT	38,776	70.89±0.26	71.89±0.27
GSN	39,547	**71.40**±0.22	**72.31**±0.21

For a fair comparison, all the graph neural network algorithms in the table 3 are unified to 3 layers, the hidden layer dimension is 128, the Adam optimizer is used, the learning rate is 0.01, and the Dropout and Early-stop techniques are adopted to prevent over-fitting. It can be found that our GSN has better results than MLP and these classic graph neural network algorithms, which proves the effectiveness of our proposed aggregation method and several tricks. The box plot in Figure 5 can more intuitively show the performance distribution of these graph neural network models after multiple random training (more than 10 times). It can be seen that GSN has the best performance on both the validation and the test dataset.

3.5 EXTENDED GENERAL HETEROGENEOUS MESSAGE PASSING PARADIGM

The existing heterogeneous graph neural network algorithms mainly have two ideas, one is based on meta-path and the other is not. The idea based on meta-path often requires a lot of manual preprocessing, and it is difficult to extend to large scale graphs. The general heterogeneous message passing paradigm and the R-GSN algorithm that we introduced earlier are all based on relation, and meta-path can be seen as a generalized higher-order relation. In this part, we appropriately extend the previous general heterogeneous message passing paradigm, which can be adjusted to the following four steps: Message Transform, Intra-metapath Aggregation, Inter-metapath Aggregation and Status Update. At the same time, we propose the M-GSN algorithm under this framework. In the M-GSN algorithm, two aggregation methods are used in the Intra-metapath Aggregation, MEAN aggregation and SIM aggregation.

We used two datasets for experiments, `IMDB` and `DBLP`. `IMDB` is an online dataset about movies and TV shows. We use a subset of `IMDB` here, which contains 4278 movies, 208 directors, and 5257 actors after data preprocessing. Each movie is represented by a bag of words describing its keywords. For the semi-supervised learning model, the movie node is divided into train/val/test sets, which contain 400 (9.35%), 400 (9.35%) and 3478 (81.30%) training nodes, validation nodes and test nodes, respectively. `DBLP` is a computer science website. We use a subset of `DBLP`. After data

Table 4: Results for `IMDB` and `DBLP`.

Datasets	Metrics	Train %	GCN	GAT	HAN	MAGNN paper	MAGNN repro	**M-GSN Mean**	**M-GSN Mean(N)**	**M-GSN Sim**	**M-GSN Sim(N)**
IMDB	Macro F1	20%	52.73	53.64	56.19	59.35	58.41	**61.04**	60.67	61.01	60.74
		40%	53.67	55.50	56.15	60.27	59.44	61.28	**61.36**	61.34	61.36
		60%	54.24	56.46	57.29	60.66	59.91	61.47	**61.50**	61.59	61.48
		80%	54.77	57.43	58.51	61.44	60.20	**61.92**	61.82	62.04	61.81
	Micro F1	20%	52.80	53.64	56.32	59.60	58.42	**61.08**	60.81	61.03	60.87
		40%	53.76	55.56	57.32	60.50	59.51	61.40	**61.52**	61.44	61.53
		60%	54.23	56.47	58.42	60.88	59.96	61.55	**61.65**	61.66	61.61
		80%	54.63	57.40	59.24	61.53	60.28	**62.01**	61.98	62.14	61.97
DBLP	Macro F1	20%	88.00	91.05	91.69	93.13	92.61	**93.33**	93.78	92.95	93.84
		40%	89.00	91.24	91.96	93.23	93.04	**93.49**	93.84	93.11	93.92
		60%	89.43	91.42	92.14	93.57	93.23	**93.60**	93.93	93.32	93.96
		80%	89.98	91.73	92.50	**94.10**	93.38	93.59	**93.90**	93.40	93.99
	Micro F1	20%	88.51	91.61	92.33	93.61	93.16	**93.82**	94.22	93.48	94.28
		40%	89.22	91.77	92.57	93.68	93.53	**93.95**	94.25	93.60	94.32
		60%	89.57	91.97	92.72	93.99	93.73	**94.08**	94.35	93.82	94.38
		80%	90.33	92.24	93.23	**94.47**	93.85	94.04	**94.31**	93.86	94.38

preprocessing, it contains 4057 authors, 14328 papers, 7723 institutions and 20 publication locations. The author is divided into four research fields (database, data mining, artificial intelligence, information retrieval). Each author is described by a bag of words of keywords in their papers. For the semi-supervised learning model, the author nodes are divided into 400 (9.35%), 400 (9.35%) and 3257 (80.28%) training nodes, validation nodes and test nodes.

We conducted node classification experiments on the `IMDB` and `DBLP` datasets. After training, we use the trained model to extract the feature of the nodes in the test set, and then divide the data according to different train/test ratios (20%, 40%, 60%, 80%) are sent to the linear SVM model for training. In order to make a fair comparison, we just sent the nodes of the test dataset to the SVM model, because our semi-supervised graph embedding algorithm has seen the nodes of the training dataset during training. Therefore, the train/test ratio of the SVM here only involves the test dataset (that is, 3478 nodes for `IMDB` and 3257 nodes for `DBLP`). We report the average Macro-F1 and Micro-F1 of 10 runs for each embedded model in Table 4.

For GNNs, including GCN, GAT, HAN, MAGNN and our proposed M-GSN, we set the dropout rate to 0.5; we use the same splits of train/val/test sets; we employ the Adam optimizer with the learning rate set to 0.005 and the weight decay set to 0.001; we train the GNNs for 100 epochs and apply early stopping with a patience of 30. For a fair comparison, we set the embedding dimension of all the models mentioned above to 64. In the table 4, MAGNN paper is the result reported in (Fu et al., 2020), MAGNN repro is the result of our reproducing the MAGNN algorithm, M-GSN Mean represents the use of MEAN aggregation, M-GSN Mean(N) represents the use of MEAN aggregation while adding the layer L2 normalization, M-GSN Sim represents using SIM aggregation, and M-GSN Sim(N) represents the use of SIM aggregation while adding the layer L2 normalization. In each row of results, we highlight the Top 3 of the F1 metric. From the results, we can find that our M-GSN basically occupies the Top 3 of the F1 metric. In the node classification task, our algorithm can learn a better node representation of the node.

4 CONCLUSION

In this paper, we define a general heterogeneous message passing paradigm, under which we can design different heterogeneous graph neural network models. In theory, any homogenous graph neural network can be migrated to this framework to obtain the corresponding heterogeneous version (homogeneous graph aggregation is equivalent to Intra-relation aggregation in this paradigm). The R-GSN we designed under this paradigm can achieve the state-of-the-art performance on the `ogbn-mag` dataset. Of course, R-GSN can still be further improved. For example, our subgraph sampling algorithm is simple neighborhood sampling, we only experimented with 2 layers of R-GSN, which is limited by the 11GB memory limit of our NVIDIA-1080Ti GPU. The aggregation method of GSN is still the first-order neighborhood aggregation, therefore, our R-GSN can only receive the information of 2-hop neighbor nodes, and cannot receive the information of semantic high-order neighbors. Deepening the number of layers while avoiding overfitting is still an open question.

REFERENCES

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

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 135–144, 2017.

Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pp. 2331–2341. ACM / IW3C2, 2020. URL https://doi.org/10.1145/3366423.3380297.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020a.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, pp. 2704–2710, 2020b.

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

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Flag: Adversarial data augmentation for graph neural networks. *arXiv preprint arXiv:2010.09891*, 2020.

Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9267–9276, 2019.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 2020.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pp. 2022–2032, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In *Advances in Neural Information Processing Systems*, pp. 11983–11993, 2019.

A APPENDIX

A.1 RELATED WORK

A.1.1 HETEROGENEOUS GRAPH

A heterogeneous graph is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ associated with a nodetype mapping $\phi : \mathcal{V} \to \mathcal{M}$ and an edgetype mapping $\psi : \mathcal{E} \to \mathcal{R}$. $\mathcal{M}$ and $\mathcal{R}$ denote the predefined sets of nodetypes and edgetypes, respectively, with $|\mathcal{M}| + |\mathcal{R}| > 2$. When $|\mathcal{M}| = 1$ and $|\mathcal{R}| = 1$, it is a homogeneous graph.

A.1.2 RELATIONAL GRAPH CONVOLUTIONAL NETWORK (R-GCN).

R-GCN (Schlichtkrull et al., 2018) is the first method which shows that the GCN framework can be extended to modeling relational data. It is proposed to solve the task of entity classification (assigning types or categorical properties to entities) and link prediction (recovery of the missing triples) on the knowledge graphs. Obviously, this method can be easily transferred to solve the task of node classification and link prediction on the heterogeneous graph.

In the R-GCN model, the author proposed to use the following forward propagation model to update the status of the nodes in the multi-relation graph.

$$h'_t = \sigma(\sum_{r \in \mathcal{R}} \sum_{s_i^r \in \mathcal{N}_t^r} \frac{1}{|\mathcal{N}_t^r|} W_{rel}^r h_{s_i^r} + W_{node} h_t), \tag{19}$$

where t is the target node, $\mathcal{N}_t^r$ denotes the set of neighbors of node t under the relation $r \in \mathcal{R}$, while $\mathcal{R}$ is the set of all the relations. s_i^r represents the i-th neighbor node of the target node t under the relation r. W_{rel}^r is the relation-specific parameter matrix and W_{node} is the node transform parameter matrix. σ is an element-wise activation function, such as ReLU($\cdot$). Intuitively, equation (19) shows how the status h_t of the node t is abtained by aggregating the status of the neighbor nodes for all the relations and then updates to the new status h'_t.

It should be pointed out that we can regard R-GCN as a migration of spatial average aggregation GNN(like Kipf & Welling (2016), Hamilton et al. (2017)) on heterogeneous graph. When $|\mathcal{R}| = 1$, equation (19) becomes the following naive GNN.

$$h'_t = \sigma(\frac{1}{|\mathcal{N}_t|} \sum_{s_i \in \mathcal{N}_t} W h_{s_i} + W_{node} h_t), \tag{20}$$

A.2 R-GSN INTRA-REALTION AGGREGATION

Figure 6 shows the details of R-GSN intra-realtion aggregation.

A.3 EXPERIMENT

A.3.1 OGBN-MAG DATASET

The ogbn-mag dataset is a heterogeneous network composed of a subset of the Microsoft Academic Graph (MAG). It contains four types of entities: papers, authors, institutions, and fields as well as four types of directed relations. The schema of the ogbn-mag dataset is shown in Figure 7, and the statistical information is shown in Table 5 and 6.
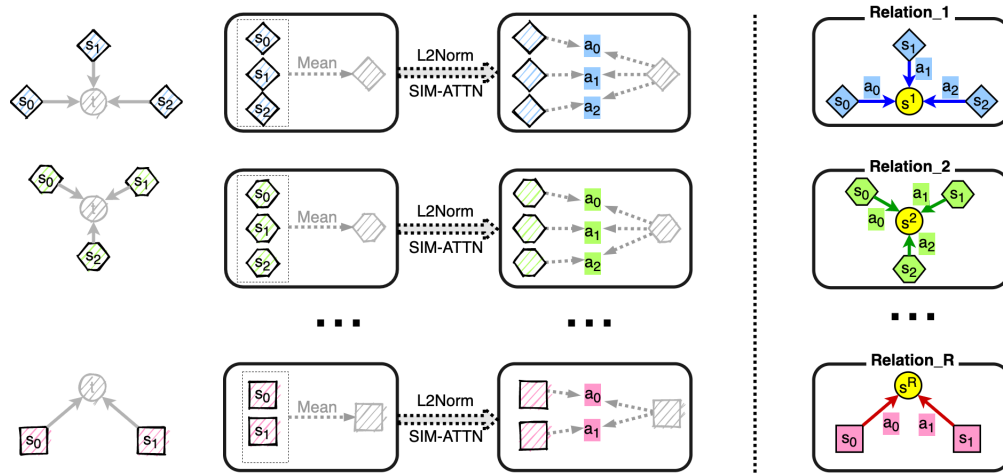
Figure 6: R-GSN Intra-realtion Aggregation (SIM-ATTN aggregation).
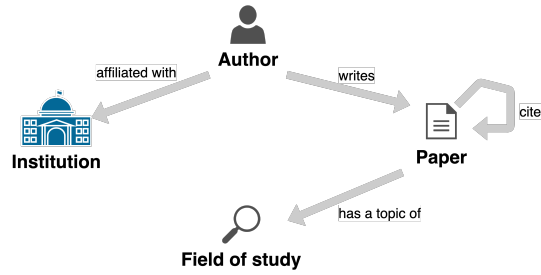

Figure 7: The schema of `ogbn-mag` dataset.

A.3.2 IMDB AND DBLP DATASETS

`IMDB` is an online dataset about movies and TV shows and `DBLP` is a dataset about computer science website. The schema and meta-path of `IMDB` and `DBLP` datasets are shown in Figure 8.
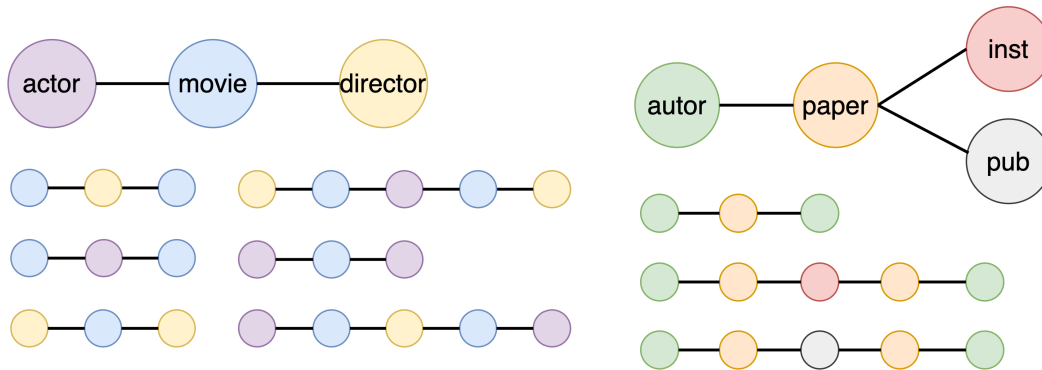

Figure 8: The schema and meta-path of `IMDB` and `DBLP` datasets

Table 5: `ogbn-mag` node statistics

Node type(4)	Number	description
Paper	736389	128-d feature, 349 categories
Author	1134649	None
Institution	8740	None
Field of study	59965	None

Table 6: `ogbn-mag` edge statistics

Edge type(4)	Number
'Author', 'affiliated with', 'Institution'	1043998
'Author', 'writes', 'Paper'	7145660
'Paper', 'cites', 'Paper'	5416271
'Paper', 'has a topic of', 'Field of study'	7505078