Ensemble Multi-Relational Graph Neural Networks

Yuling Wang[1,2] , **Hao Xu**[2] , **Yanhua Yu**[1*] , **Mengdi Zhang**[2] , **Zhenhao Li**[1] , **Yuji Yang**[2] and **Wei Wu** [2]

[1]Beijing University of Posts and Telecommunications
[2]Meituan
wangyl0612@bupt.edu.com, kingsleyhsu1@gmail.com, yuyanhua@bupt.edu.com, zhangmengdi02@meituan.com, lzhbupt@bupt.edu.com, {yangyujiyyj, wuwei19850318}@gmail.com

Abstract

It is well established that graph neural networks (GNNs) can be interpreted and designed from the perspective of optimization objective. With this clear optimization objective, the deduced GNNs architecture has sound theoretical foundation, which is able to flexibly remedy the weakness of GNNs. However, this optimization objective is only proved for GNNs with single-relational graph. *Can we infer a new type of GNNs for multi-relational graphs by extending this optimization objective, so as to simultaneously solve the issues in previous multi-relational GNNs, e.g., over-parameterization?* In this paper, we propose a novel ensemble multi-relational GNNs by designing an ensemble multi-relational (EMR) optimization objective. This EMR optimization objective is able to derive an iterative updating rule, which can be formalized as an ensemble message passing (EnMP) layer with multi-relations. We further analyze the nice properties of EnMP layer, e.g., the relationship with multi-relational personalized PageRank. Finally, a new multi-relational GNNs which well alleviate the over-smoothing and over-parameterization issues are proposed. Extensive experiments conducted on four benchmark datasets well demonstrate the effectiveness of the proposed model.[1]

1 Introduction

Graph neural networks (GNNs), which have been applied to a large range of downstream tasks, have displayed superior performance on dealing with graph data within recent years, e.g., biological networks [Huang *et al.*, 2020] and knowledge graphs [Yu *et al.*, 2021]. Generally, the current GNN architecture follows the message passing frameworks, where the propagation process is the key component. For example, GCN [Kipf and Welling, 2016] directly aggregates and propagates transformed features along the topology at each layer. PPNP [Klicpera *et al.*, 2018] aggregates both of the transformed features and the original features at each layer. JKNet [Xu *et al.*, 2018] selectively combines the aggregated messages from different layers via concatenation/max-pooling/attention operations.

Recent studies [Zhu *et al.*, 2021; Ma *et al.*, 2021] have proven that despite different propagation processes of various GNNs, they usually can be fundamentally unified as an optimization objective containing a feature fitting term O_{fit} and a graph regularization term O_{reg} as follows:

$$\boldsymbol{O} = \min_{\mathbf{Z}} \{ \underbrace{\zeta \|\mathbf{F}_1\mathbf{Z} - \mathbf{F}_2\mathbf{H}\|_F^2}_{O_{\text{fit}}} + \underbrace{\xi \operatorname{tr}\left(\mathbf{Z}^T\tilde{\mathbf{L}}\mathbf{Z}\right)}_{O_{\text{reg}}} \}, \tag{1}$$

where $\mathbf{H}$ is the original input feature and $\mathbf{L}$ is the graph Laplacian matrix encoding the graph structure. $\mathbf{Z}$ is the propagated representation, and $\mathbf{F}_1$, $\mathbf{F}_2$ are defined as arbitrary graph convolutional kernels and usually set as $\mathbf{I}$. This optimization objective reveals a mathematical guideline that essentially governs the propagation mechanism, and opens a new path to design novel GNNs. That is, such clear optimization objective is able to derive the corresponding propagation process, further making the designed GNN architecture more interpretable and reliable [Zhu *et al.*, 2021; Liu *et al.*, 2021; Yang *et al.*, 2021]. For example, [Zhu *et al.*, 2021] replaces $\mathbf{F}_1$ and $\mathbf{F}_2$ with high-pass kernel and infers new high-pass GNNs; [Liu *et al.*, 2021] applies l_1 norm to O_{reg} term and infers Elastic GNNs.

Despite the great potential of this optimization objective on designing GNNs, it is well recognized that it is only proposed for traditional homogeneous graphs, rather than the multi-relational graphs with multiple types of relations. However, in real-world applications, multi-relational graphs tend to be more general and pervasive in many areas. For instance, the various types of chemical bonds in molecular graphs, and the diverse relationships between people in social networks. Therefore, it is greatly desired to design GNN models that are able to adapt to multi-relational graphs. Some literatures have been devoted to the multi-relational GNNs, which can be roughly categorized into feature mapping based approaches [Schlichtkrull *et al.*, 2018] and learning relation embeddings based approaches [Vashishth *et al.*, 2019]. However, these methods usually design the propagation process heuristically without a clear and an explicit mathematical objective. Despite they improve the performance, they still suffer from the problems of over-parameterization [Vashishth *et al.*, 2019] and over-smoothing [Oono and Suzuki, 2019].

*Corresponding author.
[1]Code and appendix are at https://github.com/tuzibupt/EMR.

"Can we remedy the original optimization objective to design a new type of multi-relational GNNs that is more reliable with solid objective, and at the same time, alleviates the weakness of current multi-relational GNNs, e.g., over-smoothing and over-parameterization?"

Nevertheless, it is technically challenging to achieve this goal. Firstly, how to incorporate multiple relations into an optimization objective. Different relations play different roles, and we need to distinguish them in this optimization objective as well. Secondly, to satisfy the above requirements, it is inevitable that the optimization objective will become more complex, maybe with more constrains. How to derive the underlying message passing mechanism by optimizing the objective is another challenge. Thirdly, even with the message passing mechanism, it is highly desired that how to integrate it into deep neural networks via simple operations without introducing excessive parameters.

In this paper, we propose a novel multi-relational GNNs by designing an ensemble optimization objective. In particular, our proposed ensemble optimization objective consists of a feature fitting term and an ensemble multi-relational graph regularization (EMR) term. Then we derive an iterative optimization algorithm with this ensemble optimization objective to learn the node representation and the relational coefficients as well. We further show that this iterative optimization algorithm can be formalized as an ensemble message passing layer, which has a nice relationship with multi-relational personalized PageRank and covers some existing propagation processes. Finally, we integrate the derived ensemble message passing layer into deep neural networks by decoupling the feature transformation and message passing process, and a novel family of multi-relational GNN architectures is developed. Our key contributions can be summarized as follows:

- We make the first effort on how to derive multi-relational GNNs from the perspective of optimization framework, so as to enable the derived multi-relational GNNs more reliable. This research holds great potential for opening new path to design multi-relational GNNs.
- We propose a new optimization objective for multi-relational graphs, and we derive a novel ensemble message passing (EnMP) layer. A new family of multi-relational GNNs is then proposed in a decoupled way.
- We build the relationships between our proposed EnMP layer with multi-relational personalized PageRank, and some current message passing layers. Moreover, our proposed multi-relational GNNs can well alleviate the over-smoothing and over-parameterazion issues.
- Extensive experiments are conducted, which comprehensively demonstrate the effectiveness of our proposed multi-relational GNNs.

2 Related Work

Graph Neural Networks. The dominant paradigms of GNNs can be generally summarized into two branches: spectral-based GNNs [Defferrard *et al.*, 2016; Klicpera *et al.*, 2018] and spatial-based GNNs [Gilmer *et al.*, 2017; Klicpera *et al.*, 2018]. Various of representative GNNs have been proposed by designing different information aggregation and update strategies along topologies, e.g., [Gilmer *et al.*, 2017; Klicpera *et al.*, 2018]. Recent works [Zhu *et al.*, 2021; Ma *et al.*, 2021] have explore the intrinsically unified optimization framework behind existing GNNs.

Multi-relational Graph Neural Networks. The core idea of multi-relational GNNs [Schlichtkrull *et al.*, 2018; Vashishth *et al.*, 2019; Thanapalasingam *et al.*, 2021] is to encode relational graph structure information into low-dimensional node or relation embeddings. As a representative relational GNNs, RGCN [Schlichtkrull *et al.*, 2018] designs a specific convolution for each relation, and then the convolution results under all relations are aggregated, these excess parameters generated are completely learned in an end-to-end manner. Another line of literature [Ji *et al.*, 2021; Wang *et al.*, 2019; Fu *et al.*, 2020; Yun *et al.*, 2019] considers the heterogeneity of edges and nodes to construct meta-paths, then aggregate messages from different meta-path based neighbors.

3 Proposed Method

Notations. Consider a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ with nodes $v_i \in \mathcal{V}$ and labeled edges (relations) $\left(v_i, r, v_j\right) \in \mathcal{E}$, where $r \in \mathcal{R}$ is a relation type. Graph structure $\mathcal{G}^r$ under relation r can be described by the adjacency matrix $\mathbf{A}^r \in \mathbb{R}^{n\times n}$, where $\mathbf{A}^r_{i,j} = 1$ if there is an edge between nodes i and j under relation r, otherwise 0. The diagonal degree matrix is denoted as $\mathbf{D}^r = \text{diag}\left(d^r_1, \cdots, d^r_n\right)$, where $d^r_j = \sum_j \mathbf{A}^r_{i,j}$. We use $\tilde{\mathbf{A}}^r = \mathbf{A}^r + \mathbf{I}$ to represent the adjacency matrix with added self-loop and $\tilde{\mathbf{D}}^r = \mathbf{D}^r + \mathbf{I}$. Then the normalized adjacency matrix is $\hat{\tilde{\mathbf{A}}}^r = \left(\tilde{\mathbf{D}}^r\right)^{-1/2} \tilde{\mathbf{A}}^r \left(\tilde{\mathbf{D}}^r\right)^{-1/2}$. Correspondingly, $\tilde{\mathbf{L}}^r = \mathbf{I} - \hat{\tilde{\mathbf{A}}}^r$ is the normalized symmetric positive semi-definite graph Laplacian matrix of relation r.

3.1 Ensemble Optimization Framework

Given a multi-relational graph, one basic requirement is that the learned representation $\mathbf{Z}$ should capture the homophily property in the graph with relation r, i.e., the representations $\mathbf{Z}_i$ and $\mathbf{Z}_j$ should be similar if nodes i and j are connected by relation r. We can achieve the above goal by minimizing to the following term with respect to $\mathbf{Z}$:

$$\text{tr}\left(\mathbf{Z}^T \tilde{\mathbf{L}}^r \mathbf{Z}\right) = \sum_{i,j}^{n} \hat{\tilde{\mathbf{A}}}^r_{i,j} \left\|\mathbf{Z}_i - \mathbf{Z}_j\right\|^2, \tag{2}$$

where $\hat{\tilde{\mathbf{A}}}^r_{i,j}$ represents the node i and node j are connected under relation r.

With all the R relations, we need to simultaneously capture the graph signal smoothness. Moreover, consider that different relations may play different roles, we need to distinguish their importance as well, which can be modelled as an ensemble multi-relational graph regularization as follows:

$$O_{\text{e-reg}} = \begin{cases} \lambda_1 \sum_{r=1}^{R} \mu_r \sum_{i,j}^{n} \hat{\tilde{\mathbf{A}}}^r_{i,j} \left\|\mathbf{Z}_i - \mathbf{Z}_j\right\|^2 + \lambda_2 \left\|\boldsymbol{\mu}\right\|_2^2, \\ \text{s.t.} \ \sum_{r=1}^{R} \mu_r = 1, \mu_r \geq 0, \forall r = 1, 2, \ldots, R, \end{cases} \tag{3}$$

where λ_1 and λ_2 are non-negative trade-off parameters. $\mu_r \geq 0$ is the weight corresponding to relation r, and the sum of weights is 1 for constraining the search space of possible graph Laplacians. The regularization term $\|\boldsymbol{\mu}\|_2^2$ is to avoid the parameter overfitting to only one relation [Geng *et al.*, 2012].

In addition to the topological constraint by $O_{\text{e-reg}}$ term, we should also build the relationship between the learned representation $\mathbf{Z}$ with the node features $\mathbf{H}$. Therefore, there is a feature fitting term: $O_{\text{fit}} = \|\mathbf{Z} - \mathbf{H}\|_F^2$, which makes $\mathbf{Z}$ encode information from the original feature $\mathbf{H}$, so as to alleviate the over-smoothing problem. Finally, our proposed optimization framework for multi-relational graphs, which includes constraints on features and topology, is as follows:

$$\begin{aligned} \arg \quad & \min_{\mathbf{Z},\boldsymbol{\mu}} \underbrace{\|\mathbf{Z} - \mathbf{H}\|_F^2}_{O_{\text{fit}}} + \underbrace{\lambda_1 \sum_{r=1}^{R} \mu_r \mathbf{tr}\left(\mathbf{Z}^\top \tilde{\mathbf{L}}^r \mathbf{Z}\right) + \lambda_2 \|\boldsymbol{\mu}\|_2^2}_{O_{\text{e-reg}}}, \\ \text{s.t.} \quad & \sum_{r=1}^{R} \mu_r = 1, \mu_r \geq 0, \forall r = 1, 2, \ldots, R. \end{aligned} \tag{4}$$

By minimizing the above objective function, the optimal representation $\mathbf{Z}$ not only captures the smoothness between nodes, but also preserves the personalized information. Moreover, the optimal relational coefficients $\boldsymbol{\mu}$ can be derived, reflecting the importance of different relations.

3.2 Ensemble Message Passing Mechanism

It is nontrivial to directly optimize $\mathbf{Z}$ and $\boldsymbol{\mu}$ together because Eq.(4) is not convex w.r.t. $(\mathbf{Z}, \boldsymbol{\mu})$ jointly. Fortunately, an iterative optimization strategy can be adopted, i.e., i.) first optimizing Eq.(4) w.r.t. $\boldsymbol{\mu}$ with a fixed $\mathbf{Z}$, resulting in the solution of relational coefficients $\boldsymbol{\mu}$; ii.) then solving Eq.(4) w.r.t. $\mathbf{Z}$ with $\boldsymbol{\mu}$ taking the value solved in the last iteration. We will show that performing the above two steps corresponds to one ensemble message passing layer in our relational GNNs.

Update Relational Coefficients

We update relational parameters $\boldsymbol{\mu}$ by fixing $\mathbf{Z}$, then the objective function (4) w.r.t. $\boldsymbol{\mu}$ is reduced to:

$$\begin{aligned} \arg \quad & \min_{\mu} \sum_{r=1}^{R} \mu_r s_r + \frac{\lambda_2}{\lambda_1} \|\boldsymbol{\mu}\|_2^2, \\ \text{s.t.} \quad & \sum_{r=1}^{R} \mu_r = 1, \boldsymbol{\mu} \geq 0, \forall r = 1, 2, \ldots, R, \end{aligned} \tag{5}$$

where $s_r = \mathbf{tr}\left(\mathbf{Z}^\top \tilde{\mathbf{L}}^r \mathbf{Z}\right)$.

(1) When $\frac{\lambda_2}{\lambda_1} = 0$, the coefficient might concentrate on one certain relation, i.e., $\mu_j = 1$ if $s_j = \min_{r=1,\ldots,R} s_r$, and $\mu_j = 0$ otherwise. When $\frac{\lambda_2}{\lambda_1} = +\infty$, each relation will be assigned equal coefficient, i.e., $\mu_r = \frac{1}{R}$ [Geng *et al.*, 2012].

(2) Otherwise, theoretically, Eq.(5) can be regarded as a convex function of $\boldsymbol{\mu}$ with the constraint in a standard simplex [Chen and Ye, 2011], i.e., $\Delta = \left\{\boldsymbol{\mu} \in \mathbb{R}^R : \sum_{r=1}^{R} \mu_r = 1, \boldsymbol{\mu} \geqslant 0\right\}$. Therefore, the mirror entropic descent algorithm (EMDA) [Beck and Teboulle, 2003] can be used to optimize $\boldsymbol{\mu}$, where the update process is described by Algorithm 1. The objective $f(\cdot)$ should be a convex Lipschitz continuous function with Lipschitz constant ϕ for a fixed given norm. Here, we derive this Lipschitz constant from $\|\nabla f(\boldsymbol{\mu})\|_1 \leq \frac{2\lambda_2}{\lambda_1} + \|\boldsymbol{s}\|_1 = \phi$, where $\boldsymbol{s} = \{s_1, \ldots, s_R\}$.

Update Node Representation

Then we update node representation $\mathbf{Z}$ with fixing $\boldsymbol{\mu}$, where the objective function Eq. (4) w.r.t. $\mathbf{Z}$ is reduced to:

$$\arg\min_{\mathbf{Z}} \|\mathbf{Z} - \mathbf{H}\|_F^2 + \lambda_1 \sum_{r=1}^{R} \mu_r \mathbf{tr}\left(\mathbf{Z}^\top \tilde{\mathbf{L}}^r \mathbf{Z}\right). \tag{6}$$

We can set the derivative of Eq. (6) with respect to $\mathbf{Z}$ to zero and get the optimal $\mathbf{Z}$ as:

$$\frac{\partial \left\{\|\mathbf{Z} - \mathbf{H}\|_F^2 + \lambda_1 \sum_{r=1}^{R} \mu_r \mathbf{tr}\left(\mathbf{Z}^\top \tilde{\mathbf{L}}^r \mathbf{Z}\right)\right\}}{\partial \mathbf{Z}} = 0 \tag{7}$$

$$\Rightarrow \quad \mathbf{Z} - \mathbf{H} + \lambda_1 \sum_{r=1}^{R} \mu_r \tilde{\mathbf{L}}^r \mathbf{Z} = 0. \tag{8}$$

Since the eigenvalue of $\mathbf{I} + \lambda_1 \sum_{r=1}^{R} \mu_r \tilde{\mathbf{L}}^r$ is positive, it has an inverse matrix, and we can obtain the closed solution as follows:

$$\begin{aligned} \mathbf{Z} &= \left(\mathbf{I} + \lambda_1 \sum_{r=1}^{R} \mu_r \tilde{\mathbf{L}}^r\right)^{-1} \mathbf{H} \\ &= \frac{1}{1+\lambda_1} \left(\mathbf{I} - \frac{\lambda_1}{1+\lambda_1} \sum_{r=1}^{R} \mu_r \hat{\tilde{\mathbf{A}}}^r\right)^{-1} \mathbf{H}. \end{aligned} \tag{9}$$

However, obtaining the inverse of matrix will cause a computational complexity and memory requirement of $O\left(n^2\right)$, which is inoperable in large-scale graphs. Therefore, we can approximate Eq.(9) using the following iterative update rule:

$$\mathbf{Z}^{(k+1)} = \frac{1}{(1+\lambda_1)} \mathbf{H} + \frac{\lambda_1}{(1+\lambda_1)} \left(\sum_{r=1}^{R} \mu_r^{(k)} \hat{\tilde{\mathbf{A}}}^r\right) \mathbf{Z}^{(k)}. \tag{10}$$

where k is the iteration number.

Ensemble Message Passing Layer (EnMP layer)

Now with the node representation $\mathbf{Z}$ and the relation coefficient $\boldsymbol{\mu}$, we can propose our ensemble message passing layer, consisting of the following two steps: (1) relational coefficient learning step (RCL step), i.e., update the relational coefficients $\boldsymbol{\mu}$ according to Algorithm 1; (2) propagation step (Pro step), i.e., update the node representation $\mathbf{Z}$ according to Eq.(10). The pseudocode of EnMP layer is shown in appendix A. We will show some properties of our proposed EnMP layer.

Remark 1 (Relationship with Multi-Relational/Path Personalized PageRank)**.** *Given a realtion r, we have the relation based probability transition matrix* $\mathbf{A}_{rw}^r = \mathbf{A}^r(\mathbf{D}^r)^{-1}$. *Then, the single relation based PageRank matrix is calculated via:*

$$\mathbf{\Pi}_{\text{pr}}^r = \mathbf{A}_{\text{rw}}^r \mathbf{\Pi}_{\text{pr}}^r. \tag{11}$$

Algorithm 1 Relational Coefficients Learning

Input: Candidate Laplacians $\left\{\tilde{\mathbf{L}}^1, \cdots, \tilde{\mathbf{L}}^R\right\}$, the embedding matrix $\mathbf{Z}$, the Lipschitz constant ϕ, the tradeoff parameters λ_1, λ_2.
Output: Relational coefficients $\boldsymbol{\mu}$.

```
Initialization: μ = [1/R, 1/R, ··· , 1/R]
for r = 1 to R do
    s_r = tr(Z^⊤ L̃^r Z)
    repeat
        T_t ← sqrt(2lnR / tφ^2),
        f'(μ_r^t) ← (2λ_2/λ_1) μ_r^t + s_r,
        μ_r^{t+1} ← μ_r^t exp[−T_t f'(μ_r^t)] / Σ_{r=1}^R μ_r^t exp[−T_t f'(μ_r^t)],
    until Convergence
end for
return μ
```

Considering we have R relations, i.e., $r = 1, 2, \ldots, R$, the weights of each relation are $\{\mu_1, \ldots, \mu_R\}$, according to [Lee et al.*, 2013; Ji* et al.*, 2021], we can define the multiple relations based PageRank matrix:*

$$\mathbf{\Pi}_{\text{pr}} = \left(\sum_{r=1}^{R} \mu_r \mathbf{A}_{\text{rw}}^r\right) \mathbf{\Pi}_{\text{pr}}^r. \tag{12}$$

According to [Klicpera et al.*, 2018], the multi-relational personalized PageRank matrix can be defined:*

$$\mathbf{\Pi}_{\text{ppr}} = \alpha \left(\boldsymbol{I}_n - (1-\alpha)\left(\sum_{r=1}^{R} \mu_r \hat{\tilde{\boldsymbol{A}}}^r\right)\right)^{-1}, \tag{13}$$

where $\hat{\tilde{\boldsymbol{A}}}^r$ is a normalized adjacency matrix with self-loops, $\boldsymbol{I}_n$ represents unit matrix, $\alpha \in (0, 1]$ is teleport (or restart) probability. If $\alpha = \frac{1}{(1+\lambda_1)}$, the closed-form solution in Eq.(9) *is to propagate features via multi-relational personalized PageRank scheme.*

Remark 2 (Relationship with APPNP/GCN)**.** *if $\lambda_2 = +\infty$, the solution in Eq.*(5) *is $\boldsymbol{\mu} = [\frac{1}{R}, \frac{1}{R}, \cdots, \frac{1}{R}]$, i.e., each relaiton is assigned equal coefficient, then the ensemble multi-relational graph $\sum_{r=1}^{R} \mu_r^{(k)} \hat{\tilde{\mathbf{A}}}^r$ reduces to a normalized adjacency matrix $\frac{1}{R}\sum_{r=1}^{R} \hat{\tilde{\mathbf{A}}}^r$ averaged over all relations. The proposed message passing scheme reduces to:*

$$\mathbf{Z}^{(k+1)} = \frac{1}{(1+\lambda_1)}\mathbf{H} + \frac{\lambda_1}{(1+\lambda_1)}\frac{1}{R}\sum_{r=1}^{R} \hat{\tilde{\mathbf{A}}}^r \mathbf{Z}^{(k)}, \tag{14}$$

if $\lambda_1 = \frac{1}{\alpha} - 1$, it recovers the message passing in APPNP on the averaged relational graph:

$$\mathbf{Z}^{(k+1)} = \alpha\mathbf{H} + (1-\alpha)\frac{1}{R}\sum_{r=1}^{R} \hat{\tilde{\mathbf{A}}}^r \mathbf{Z}^{(k)}, \tag{15}$$

if $\lambda_1 = +\infty$, it recovers the message passing in GCN on the averaged relational graph:

$$\mathbf{Z}^{(k+1)} = \frac{1}{R}\sum_{r=1}^{R} \hat{\tilde{\mathbf{A}}}^r \mathbf{Z}^{(k)}. \tag{16}$$

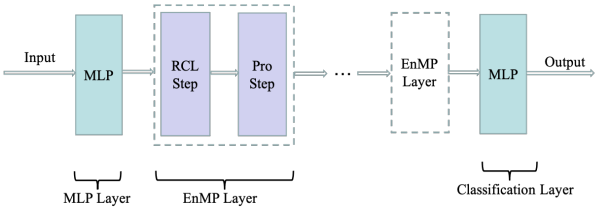


Figure 1: Model architecture.

Datasets	Nodes	Node Types	Edges	Edge Types	Target	Classes
MUTAG	23,644	1	74,227	23	molecule	2
BGS	333,845	1	916,199	103	rock	2
DBLP	26,128	4	239,566	6	author	4
ACM	10,942	4	547,872	8	paper	3

Table 1: Statistics of multi-relational datasets.

3.3 Ensemble Multi-Relational GNNs

Now we propose our ensemble multi-relational graph neural networks (EMR-GNN) with the EnMP layer. Similar as [Klicpera *et al.*, 2018], we employ the decoupled style architecture, i.e., the feature transformation and the message passing layer are separated. The overall framework is shown in Figure 1, and the forward propagation process is as follows:

$$\mathbf{Y}_{\text{pre}} = g_\theta\left(\mathbf{EnMP}^{(K)}\left(f\left(\mathbf{X}; \mathbf{W}\right), R, \lambda_1, \lambda_2\right)\right), \tag{17}$$

where $\mathbf{X}$ is the input feature of nodes, and $f(\mathbf{X}; \mathbf{W})$ denotes the MLPs or linear layers (parameterized by $\mathbf{W}$) which is used to feature extraction. $\mathbf{EnMP}^{(K)}$ represents our designed ensemble relational message passing layer with K layers, where R is the number of relations, and λ_1, λ_2 are hyperparameters in our message passing layer. $g_\theta(\cdot)$ is MLPs as classifier with the learnable parameters θ. The training loss is: $\ell(\mathbf{W}, \theta) \triangleq \mathcal{D}\left(\boldsymbol{y}_i^*, \hat{\boldsymbol{y}}_i\right)$, where $\mathcal{D}$ is a discriminator function of cross-entropy, $\boldsymbol{y}_i^*$ and $\hat{\boldsymbol{y}}_i$ are the predicted and ground-truth labels of node i, respectively. Backpropagation manner is used to optimize parameters in MLPs, i.e., $\mathbf{W}$ and θ, and the parameters in our EnMP layers are optimized during the forward propagation. We can see that EMR-GNN is built on a clear optimization objective. Besides, EMR-GNN also has the following two advantages:

- As analyzed by Remark 1, our proposed EnMP can keep the original information of the nodes with a teleport (or restart) probability, thereby alleviating over-smoothing.
- For each relation, there is a parameterized relation-specific weight matrix or parameterized relation encoder used in the traditional RGCN [Vashishth *et al.*, 2019; Schlichtkrull *et al.*, 2018]. While in our EnMP, only one learnable weight coefficient is associated with a relation, greatly alleviating the over-parameterization problem.

4 Experiment

4.1 Experimental Settings

Datasets. The following four real-world heterogeneous datasets in various fields are utilized and can be divided

Dataset	DBLP		ACM		MUTAG		BGS	
Metric	Acc (%)	Recall (%)	Acc (%)	Recall (%)	Acc (%)	Recall (%)	Acc (%)	Recall (%)
GCN	90.39±0.38	89.49±0.52	89.58±1.47	89.47±1.49	72.35±2.17	63.28±2.95	85.86±1.96	80.21±2.21
GAT	91.97±0.40	91.25±0.58	88.99±1.58	88.89±1.56	70.74±2.13	63.01±3.79	88.97±3.17	86.13±4.96
HAN	91.73±0.61	91.15±0.72	88.51±0.35	88.50±0.30	-	-	-	-
RGCN	90.08±0.60	88.56±0.76	89.79±0.62	89.71±0.59	71.32±2.11	61.97±3.52	85.17±5.87	81.58±7.94
e-RGCN	91.77±0.90	91.18±1.02	83.00±1.04	84.03±0.75	69.41±2.84	**67.57±8.04**	82.41±1.96	84.51±3.38
EMR-GNN	**93.54±0.50**	**92.39±0.78**	**90.87±0.11**	**90.84±0.13**	**74.26±0.78**	64.19±1.08	**89.31±4.12**	**86.39± 5.33**

Table 2: The mean and standard deviation of classification accuracy and recall over 10 different runs on four datasets.

GCN	GAT	HAN	RGCN	e-RGCN	EMR-GNN
$O\left(Kd^2\right)$	$O\left(2KNd^2\right)$	$O\left(2K\left(\lvert\mathcal{R}\rvert N+1\right)d^2+2Kd\right)$	$O\left(\mathcal{B}Kd^2+\mathcal{B}K\lvert\mathcal{R}\rvert\right)$	$O\left(\mathcal{B}(K-1)d^2+\lvert\mathcal{R}\rvert d+\mathcal{B}(K-1)\lvert\mathcal{R}\rvert\right)$	$O\left(2d^2+K\lvert\mathcal{R}\rvert\right)$

Table 3: Comparison of the number of parameters. Here, K denotes the number of layers in the model, d is the embedding dimension, $\mathcal{B}$ represents the number of bases, $|\mathcal{R}|$ indicates the total number of relations in the graph and N is the number of heads of attention-based models.

into two categories: i) the node type and edge type are both heterogeneous (DBLP [Fu *et al.*, 2020], ACM [Lv *et al.*, 2021]). ii) the node type is homogeneous but the edge type is heterogeneous (MUTAG [Schlichtkrull *et al.*, 2018], BGS [Schlichtkrull *et al.*, 2018]). The statistics of the datasets can be found in Table 1. The basic information about datasets is summarized in appendix B.1.

Baselines. To test the performance of the proposed EMR-GNN, we compare it with five state-of-the-art baselines. Among them, GCN [Kipf and Welling, 2016] and GAT [Veličković *et al.*, 2017] as two popular approaches are included. In addition, we compare with the heterogeneous graph model HAN [Wang *et al.*, 2019], since HAN can also employ multiple relations. Two models that are specially designed for multi-relational graphs are compared, i.e., RGCN [Schlichtkrull *et al.*, 2018] and e-RGCN [Thanapalasingam *et al.*, 2021].

Parameter settings. We implement EMR-GNN based on Pytorch.[2] For $f(\mathbf{X};\mathbf{W})$ and $g_\theta(\cdot)$, we choose one layer MLP for DBLP and ACM, and linear layers for MUTAG and BGS. We conduct 10 runs on all datasets with the fixed training/validation/test split for all experiments. More implementation details can be seen in appendix B.3.

4.2 Node Classification Results

Table 2 summarizes the performances of EMR-GNN and several baselines on semi-supervised node classification task. Since HAN's code uses the heterogeneity of nodes to design meta-paths, we do not reproduce the results of HAN on homogeneous dataset (MUTAG, BGS) with only one type of nodes. We use accuracy (Acc) and recall metrics for evaluation, and report the mean and standard deviation of classification accuracy and recall. We have the following observations: (1) Compared with all baselines, the proposed EMR-GNN generally achieves the best performance across all datasets on seven of the eight metrics, which demonstrates the effectiveness of our proposed model. e-RGCN has a higher recall but a lower accuracy on MUTAG, which may be caused by overfitting. (2) Meanwhile, the number of parameters of our model and other baselines are shown in Table 3. We can see that EMR-GNN is more parameter efficient than all baselines, i.e., $O\left(2d^2+K|\mathcal{R}|\right)$, but achieves maximum relative improvements of 4.14% than RGCN on BGS. It means that EMR-GNN largely overcomes the over-parameterization in previous multi-relational GNNs.

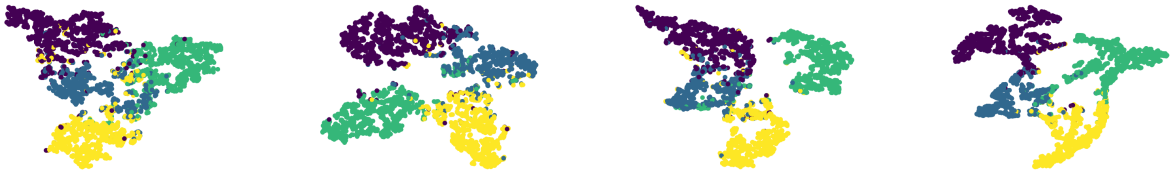
[2]https://pytorch.org/

4.3 Model Analysis

Alleviating over-smoothing problem. As mentioned before, EMR-GNN is able to alleviate over-smoothing issue. Here, we take one typical single-relation GCN (GAT) and one representative multi-relational GCN (RGCN) as baselines to test their performance with different propagation depths, where the results are shown in Figure.3. We have the following observations: (1) Our model significantly alleviates the over-smoothing problem, since there is generally no performance degradation when the depth increases. This benefits from the adjustable factor λ_1 in EMR-GNN, which flexible controls the influence of node feature information. In contrast, the performance of RGCN and GAT drops seriously with increasing depth, implying that these models suffer from the over-smoothing problem. (2) RGCN needs huge storage cost, making it difficult to stack multiple layers. Cuda out of memory occurs when the propagation depth increases, i.e., DBLP for more than 16 layers, and ACM can merely stack 8 layers. This is not available for capturing long-range dependencies.

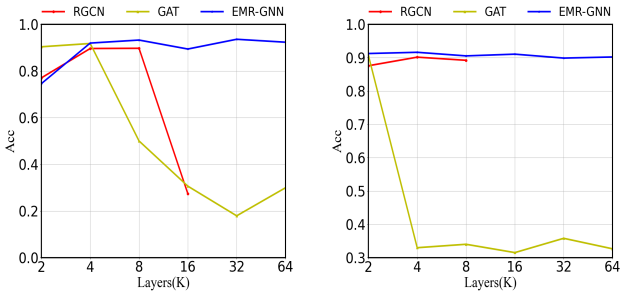
Datasets	Method	Size of training set			
		10%	15%	20%	25%
DBLP	RGCN	0.5381	0.6388	0.7515	0.7721
	EMR-GNN	0.8768	0.9109	0.9128	0.9364
ACM	RGCN	0.7492	0.8136	0.8278	0.8344
	EMR-GNN	0.8489	0.8654	0.8739	0.8753

Table 4: Classification accuracy w.r.t. different training set.



(a) GCN (SC=0.6196) (b) GAT (SC=0.6351) (c) RGCN (SC=0.6093) (d) EMR-GNN (SC=0.7799)

Figure 2: Visualization of the learned node embeddings on DBLP dataset.



(a) DBLP (b) ACM

Figure 3: Analysis of propagation layers. Missing value in red line means CUDA is out of memory.

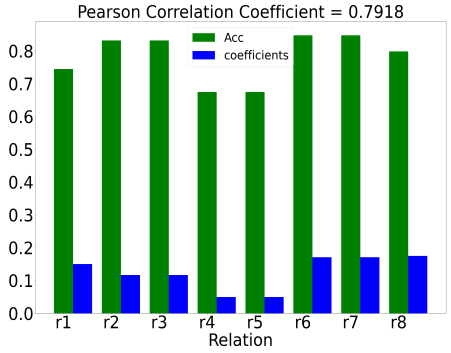


Figure 4: Accuracy under each single relation and corresponding relational coefficient.

Alleviating over-parameterization problem. To further illustrate the advantages of alleviating over-parameterization, we verify EMR-GNN with small-scale training samples. We conduct experiments on EMR-GNN and RGCN using two datasets. We only select a small part of nodes from original training samples as the new training samples. As can be seen in Table 4, EMR-GNN consistently outperforms RGCN with different training sample ratios, which again validates the superiority of the proposed method. One reason is that a limited number of parameters in EMR-GNN can be fully trained with few samples. In contrast, RGCN with excess parameters requires large-scale training samples as the number of relations increases. The time complexity is analyzed in appendix C.

Analysis of relational coefficients. Besides the performance, we further show that EMR-GNN can produce reasonable relational coefficients. To verify the ability of relational coefficients learning, taking ACM dataset as example, we evaluate the classification performance under each single relation. The classification accuracy and the corresponding relational coefficient value are reported in Figure 4. We can see that basically, the relation which achieves better accuracy is associated with a larger coefficient. Moreover, we compute the pearson correlation coefficient between the accuracy of a single relation and its relational coefficient, which is 0.7918, well demonstrating that they are positively correlated.

Visualization. For a more intuitive comparison, we conduct the task of visualization on DBLP dataset. We plot the output embedding on the last layer of EMR-GNN and three baselines (GCN, GAT and RGCN) using t-SNE [Van der Maaten and Hinton, 2008]. All nodes in Figure 2 are colored by the ground truth labels. It can be observed that EMR-GNN performs best, since the significant boundaries between nodes of different colors, and the relatively dense distribution of nodes with the same color. However, the nodes with different labels of GCN and RGCN are mixed together. In addition, we also calculate the silhouette coefficients (SC) of the classification results of different models, and EMR-GNN achieves the best score, furthering indicating that the learned representations of EMR-GNN have a clearer structure.

5 Conclusion

In this work, we study how to design multi-relational graph neural networks from the perspective of optimization objective. We propose an ensemble optimization framework, and derive a novel ensemble message passing layer. Then we present the ensemble multi-relational GNNs (EMR-GNN), which has nice relationship with multi-relational/path personalized PageRank and can recover some popular GNNs. EMR-GNN not only is designed with clear objective function, but also can well alleviate over-smoothing and over-parameterization issues. Extensive experiments demonstrate the superior performance of EMR-GNN over the several state-of-the-arts.

Acknowledgements

The research was supported in part by the National Natural Science Foundation of China (Nos. 61802025, 61872836, U1936104) and Meituan.

References

[Beck and Teboulle, 2003] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[Chen and Ye, 2011] Yunmei Chen and Xiaojing Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011.

[Defferrard *et al.*, 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.

[Fu *et al.*, 2020] Xinyu Fu, Jiani Zhang, Ziqiao Meng, et al. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, pages 2331–2341, 2020.

[Geng *et al.*, 2012] Bo Geng, Dacheng Tao, Chao Xu, et al. Ensemble manifold regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1227–1233, 2012.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, et al. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[Huang *et al.*, 2020] Kexin Huang, Cao Xiao, Lucas Glass, et al. Skipgnn: predicting molecular interactions with skip-graph networks. *Scientific reports*, 10(1):1–16, 2020.

[Ji *et al.*, 2021] Houye Ji, Xiao Wang, Chuan Shi, et al. Heterogeneous graph propagation network. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

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

[Klicpera *et al.*, 2018] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

[Lee *et al.*, 2013] Sangkeun Lee, Sungchan Park, Minsuk Kahng, et al. Pathrank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. *Expert Systems with Applications*, 40(2):684–697, 2013.

[Liu *et al.*, 2021] Xiaorui Liu, Wei Jin, Yao Ma, et al. Elastic graph neural networks. In *International Conference on Machine Learning*, pages 6837–6849. PMLR, 2021.

[Lv *et al.*, 2021] Qingsong Lv, Ming Ding, Qiang Liu, et al. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1150–1160, 2021.

[Ma *et al.*, 2021] Yao Ma, Xiaorui Liu, Tong Zhao, et al. A unified view on graph neural networks as graph signal denoising. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1202–1211, 2021.

[Oono and Suzuki, 2019] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, et al. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[Thanapalasingam *et al.*, 2021] Thiviyan Thanapalasingam, Lucas van Berkel, Peter Bloem, and Paul Groth. Relational graph convolutional networks: A closer look. *arXiv preprint arXiv:2107.10015*, 2021.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[Vashishth *et al.*, 2019] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, et al. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, et al. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.

[Xu *et al.*, 2018] Keyulu Xu, Chengtao Li, Yonglong Tian, et al. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pages 5453–5462. PMLR, 2018.

[Yang *et al.*, 2021] Yongyi Yang, Tang Liu, Yangkun Wang, et al. Graph neural networks inspired by classical iterative algorithms. In *International Conference on Machine Learning*, pages 11773–11783. PMLR, 2021.

[Yu *et al.*, 2021] Donghan Yu, Yiming Yang, Ruohong Zhang, et al. Knowledge embedding based graph convolutional network. In *Proceedings of the Web Conference 2021*, pages 1619–1628, 2021.

[Yun *et al.*, 2019] Seongjun Yun, Minbyul Jeong, Raehyun Kim, et al. Graph transformer networks. *Advances in Neural Information Processing Systems*, 32:11983–11993, 2019.

[Zhu *et al.*, 2021] Meiqi Zhu, Xiao Wang, Chuan Shi, et al. Interpreting and unifying graph neural networks with an optimization framework. In *Proceedings of the Web Conference 2021*, pages 1215–1226, 2021.