Signed Graph Neural Network with Latent Groups

Haoxin Liu*
Tsinghua University
China
liu-hx20@mails.tsinghua.edu.cn

Ziwei Zhang*
Tsinghua University
China
zw-zhang16@mails.tsinghua.edu.cn

Peng Cui
Tsinghua University
China
cuip@tsinghua.edu.cn

Yafeng Zhang
Qiang Cui
Meituan
China
zhangyafeng@meituan.com
cuiqiang04@meituan.com

Jiashuo Liu
Tsinghua University
China
liujiashuo77@gmail.com

Wenwu Zhu
Tsinghua University
China
wwzhu@tsinghua.edu.cn

ABSTRACT

Signed graph representation learning is an effective approach to analyze the complex patterns in real-world signed graphs with the co-existence of positive and negative links. Most previous signed graph representation learning methods resort to balance theory, a classic social theory that originated from psychology as the core assumption. However, since balance theory is shown equivalent to a simple assumption that nodes can be divided into two conflicting groups, it fails to model the structure of real signed graphs. To solve this problem, we propose Group Signed Graph Neural Network (GS-GNN) model for signed graph representation learning beyond the balance theory assumption. GS-GNN has a dual GNN architecture that consists of the global and the local module. In the global module, we adopt a more generalized assumption that nodes can be divided into multiple latent groups and that the groups can have arbitrary relations and propose a novel prototype-based GNN to learn node representations based on the assumption. In the local module, to give the model enough flexibility in modeling other factors, we do not make any prior assumptions, treat positive links and negative links as two independent relations, and adopt a relational GNN to learn node representations. Both modules can complement each other, and the concatenation of two modules is fed into downstream tasks. Extensive experimental results demonstrate the effectiveness of our GS-GNN model on both synthetic and real-world signed graphs by greatly and consistently outperforming all the baselines and achieving new state-of-the-art results. Our implementation is available in PyTorch[1].

CCS CONCEPTS

• **Information systems** → **Social networks**; *Data mining*.

*Both authors contributed equally to this research
[1]https://github.com/haoxin1998/GS-GNN

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467355

KEYWORDS

Signed Graphs, Graph Neural Networks, Balance Theory

ACM Reference Format:
Haoxin Liu, Ziwei Zhang, Peng Cui, Yafeng Zhang, Qiang Cui, Jiashuo Liu, and Wenwu Zhu. 2021. Signed Graph Neural Network with Latent Groups. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3447548.3467355

1 INTRODUCTION

With the prevalence of social media, social interactions are created every moment. People create positive relationships such as friendships, agreements, and supports, as well as negative relationships such as foes, disagreements, and boycotts. For example, most customer review sites allow users to express both positive and negative comments. Signed graphs are popular data structures to simultaneously represent these positive and negative relationships by assigning signs to the links, i.e., co-existence of positive and negative links. Compared with unsigned graphs, signed links can reflect more complex social relationships [23].

To mine the rich value underlying graph data, graph representation learning techniques have been boosted recently. By learning a low-dimensional representation for each node, various graph analytical tasks such as node classification and link prediction can be performed. Among different methods, graph neural networks (GNNs) demonstrate great potentials by designing deep neural networks in learning node representations. When generalizing representation learning methods to signed graphs, modeling the co-existence of positive and negative links becomes the principle design. Most existing GNNs for signed graphs [4, 5] resort to balance theory, a well-known social theory originating from psychology, in designing learning frameworks. In a nutshell, balance theory indicates that "the foe of my foe is my friend" [10].

However, does balance theory perfectly model real signed graphs? To answer this question, we find it has been shown [9, 12] that balance theory is equivalent to a simple assumption that the nodes can be partitioned into two groups and the signs of links are completely determined by the group structure, i.e., nodes form positive links if they are in the same group and negative links if they are in different groups. Clearly, such an assumption is too ideal for real signed graphs. Taking social networks as examples, people form

positive and negative links based on various reasons rather than a dichotomy. The existing signed graphs representation learning methods do not consider these complex relationships, thus failing to model the true underlying structure of real signed graphs. As a result, the performance of the learned representations in signed graph analytical tasks is seriously harmed.

A natural generalization to the balance theory is that there exist multiple groups underlying the signed graph structure [3, 29]. Taking one step further, the groups can also form complex relationships, i.e., not every two groups are in conflicts and only form negative links. Such a generalized assumption can better model the real underlying structure of signed graphs, which we term such as the *k-group theory*. It is easy to see that the k-group theory includes the balance theory as a special case when the number of groups equals two and the two groups are in conflict.

Nevertheless, when adopting the k-group theory as the major inductive bias in designing a representation learning method, we should also give the model enough flexibility to accommodate other factors such as the micro-structures within groups, influences of node features, or even tolerate the randomness in people's behaviors or noises in the data. Such flexibility is critical since the observed data may not perfectly fit the k-group theory, even when it is the prime underlying mechanism of forming a signed graph. Otherwise, the model can be easily affected by these other factors and fails to discover the latent group structures.

Motivated by these analyses, we propose Group Signed Graph Neural Network (GS-GNN), a novel signed GNN that goes beyond the balance theory. Specifically, GS-GNN adopts a dual architecture that learns two representations for each node. For the first representation, we utilize the k-group theory and propose a novel prototype-based GNN to jointly learn the representation for each group and the affiliation of nodes to different groups. The representation of a node is composed of a linear combination of the group representations weighed by the node affiliations. Since this representation depicts the node relationships with all the groups, we name it the *global representation* of nodes. For the second representation, we do not make any prior assumption on the generative mechanism of signs and adopt a multi-relational GNN to model positive and negative links separately for node presentation learning. It complements the global representation by capturing information exceeding the k-group theory, e.g., structures within groups or information related to node features. Accordingly, we name such a representation as to the *local representation*. Finally, we concatenate the global and local representation to handle downstream tasks. By adopting this decoupled dual architecture, GS-GNN can capture various latent factors and the complex generative mechanism underlying the formation of signed graphs.

We conduct extensive experiments on four publicly available real signed graphs for the link sign prediction task. The experimental results show that our proposed method significantly and consistently outperforms all the existing methods, including signed network embedding methods and signed GNNs. Experimental results on simulation datasets and ablation studies further verify that the design of our method can indeed capture more complex structures than the balance theory.

The main contributions of the paper are summarized as follows.

- We study representation learning methods for signed graphs using the k-group theory, a general and more realistic assumption beyond the usual balance theory.
- We propose GS-GNN, a novel signed GNN with a dual architecture to learn global and local representations simultaneously. Our proposed method can fully leverage the k-group theory and is flexible to capture extra information beyond the theory.
- Extensive experimental results on real signed graphs demonstrate the superiority of our proposed method. GS-GNN outperforms all the existing methods and achieves new state-of-the-art, to the best of our knowledge.

2 RELATED WORK

In this section, we review related works. First, we review graph representation learning methods for unsigned graphs. Then, we review signed graph representation learning methods based on the balance theory and their relationships with the k-group theory. Finally, we review other signed graph representation learning methods.

Graph representation learning techniques, which learn low-dimensional vectors for nodes, have shown successes in various graph analysis tasks, such as node classification [27], link prediction [30], and network visualization [11]. Initial random-walk based methods such as DeepWalk [26], LINE [28], and Node2vec [8] regard nodes as words and generate random walks to preserve the graph structure. Other works such as SDNE [30] and SDAE [1] adopt deep neural networks and also achieve good performance. Recently, graph neural networks (GNNs) based methods have become increasingly popular [7, 18, 32, 37]. The message-passing framework [7] of GNNs propagates and aggregates information between neighbors. However, these methods are designed for unsigned graphs and thus not directly applicable to signed graphs.

Most of the existing signed graph representation learning methods [4, 5, 16, 17, 22, 25, 31] are designed based on the balance theory [10]. For example, SiNE [31] first learns signed network embedding with an objective function guided by the balance theory. SIDE [17] and SIGNet [15] use specifically designed random walks to model the balance theory. SGCN [5] first utilizes GNNs to model signed networks and uses the balance theory in designing the aggregating strategy. Similarly, SNEA [25] uses the attention mechanism to reveal the importance of neighbors and SGDN [16] uses feature diffusion to take high-order neighbors into consideration.

As mentioned in [9], the balance theory is equivalent to a simple assumption that the nodes can be partitioned into two conflicting groups and the signs of links are completely determined by the group structure. The weak balance theory is proposed in [2] and considers more than two conflicting groups. Based on this assumption, several signed graph clustering methods [3, 19] are proposed to discover the signed graph partition. However, these methods all assume the groups are all in conflicts. Besides, they do not utilize the advantages of representation learning. To the contrary, our proposed GS-GNN can utilize GNNs and also model arbitrary relations between groups.

There are also signed representation learning methods [13, 14] based on another theory, the social status theory [23]. However, the social status theory is only applicable to directed signed graphs, while we only consider undirected signed graphs in this paper.

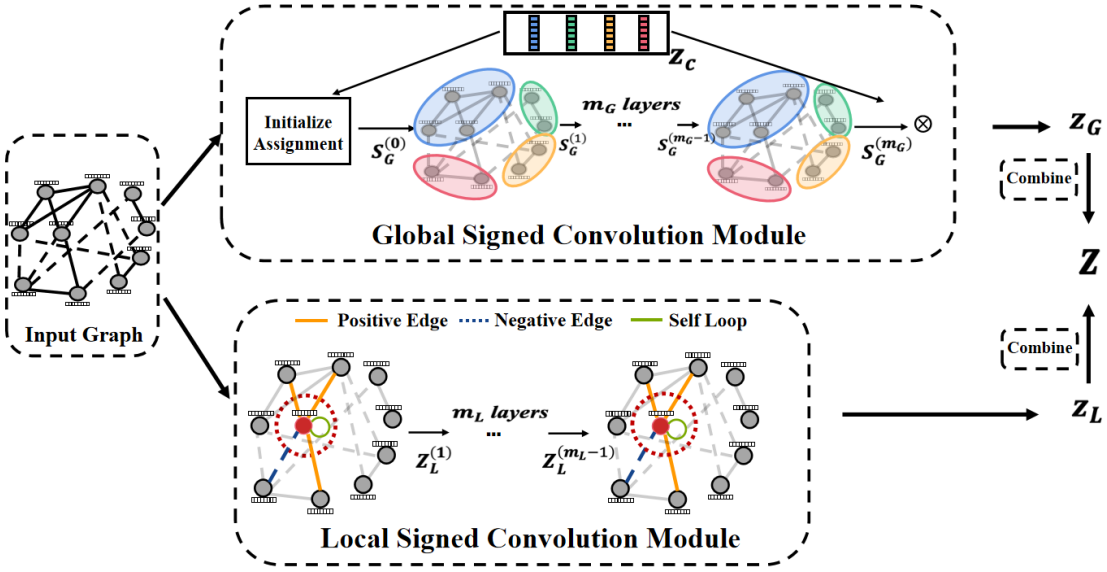


Figure 1: The overall framework of our GS-GNN model.

3 PROPOSED GS-GNN METHOD

3.1 Notations

First, we introduce some notations used throughout the paper. We consider a signed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$ where $\mathcal{V} = \{v_1, v_2, ..., v_{|V|}\}$ denotes the set of nodes, $\mathcal{E}^+ \subseteq \mathcal{V} \times \mathcal{V}$ denotes the positive links, and $\mathcal{E}^- \subseteq \mathcal{V} \times \mathcal{V}$ denotes the negative links. $\mathcal{E}^+ \bigcap \mathcal{E}^- = \emptyset$, i.e., a link is either positive or negative. The nodes also have input features denoted as $X \in \mathbb{R}^{|V| \times d_{in}}$, where d_{in} is the number of features. In Table 1, we summarize the main notations.

Table 1: Main notations used throughout the paper

Notations	Descriptions
$\mathcal{G}$	A signed graph
$\mathcal{V}$	A set of nodes
$\mathcal{E}^+(\mathcal{E}^-)$	A set of positive (negative) links
$\mathcal{N}_i^+(\mathcal{N}_i^-)$	The positive (negative) neighbors of node v_i
$\mathbf{X}$	An input features matrix
$\mathbf{Z}_G \in \mathbb{R}^{\|V\| \times d_L}$	The global embedding matrix of nodes
$\mathbf{Z}_L \in \mathbb{R}^{\|V\| \times d_G}$	The local embedding matrix of nodes
$d_G(d_L)$	The dimensionality of the global (local) embedding
$\mathbf{Z} = [\mathbf{Z}_G, \mathbf{Z}_L]$	The final embedding matrix of nodes
K	The number of groups
$\mathbf{Z}_C$	The embedding matrix of groups
$\mathbf{S}^{(m)}$	The assignment matrix of the m^{th} global layer
$M_G(M_L)$	The number of layers in the global (local) module
$\mathbf{W}_G^{(m)}, \mathbf{W}_G^{(m)\prime}$	The weight matrices in the m^{th} layer of the global module
$\mathbf{W}_L^{(m)}$ $\left(b_L^{(m)}\right)$	The weight matrix (bias vector) in the m^{th} layer of the local module

3.2 Overall Framework

The overall framework of GS-GNN is shown in Figure 1. The major design of GS-GNN is that we use a dual architecture to learn a global embedding matrix $\mathbf{Z}_G \in \mathbb{R}^{|V| \times d_G}$ and a local embedding matrix $\mathbf{Z}_L \in \mathbb{R}^{|V| \times d_L}$ as the node representations. The global embedding utilizes the k-group theory that signed graphs have a latent k-group structure and the relations between groups are unknown. On the other hand, the local embedding aims to incorporate the individual information not captured by the k-group theory. Both embeddings are learned with GNNs using specifically designed convolution layers. To keep it simple, we set the final embedding $\mathbf{Z}$ as the concatenation of the local and global embeddings, though more advanced methods such as using a gating or attention mechanism can be explored in the future.

3.3 Global Signed Convolutional Module

Next, we explain the details for the global signed convolution module and local signed convolution module, respectively. Finally, we introduce our objective function.

The goal of the global signed convolution module is to discover the latent community structure of signed graphs based on the k-group theory. There are several challenges for designing such an architecture. First, we need to model the complex relationships between groups beyond simple heuristic assumptions, e.g., all groups are in conflict. Second, we need to represent nodes from the perspective of groups, i.e., discovering the affiliation between nodes and groups. Lastly, we do not want to increase the number of parameters and time complexity so that our model can apply to large-scale graphs.

With the above goals in mind, we propose a novel prototype-based GNN as our global signed convolutional module. The framework is shown in Figure 2. We denote an embedding matrix for the groups as $\mathbf{Z_C} = \left[\mathbf{Z}_{C_1}, \mathbf{Z}_{C_2}, ..., \mathbf{Z}_{C_K}\right] \in \mathbb{R}^{K \times d_G}$, where $\mathbf{Z}_{C_k} \in \mathbb{R}^{d_G}$

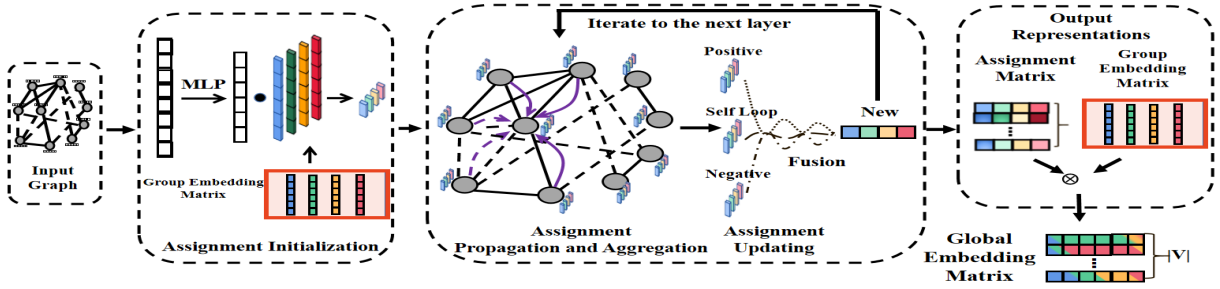


Figure 2: The detailed designs of the global module in GS-GNN. There are four major steps. First, we initialize the assignment matrix of nodes to different groups by transforming the node features into a latent space using an MLP and calculating the attention score with the group embedding matrix. Second, we conduct message-passing on the assignment matrix to propagate the aggregate information from both positive and negative neighbors. Third, we fuse the received messages and update the assignment matrix. After iterating steps two and three, i.e. multiple message-passing layers, we calculate the final global embedding matrix as a linear combination of the group embedding using the assignment as weights.

is the embedding of the k^{th} group. $\mathbf{Z}_C$ is randomly initialized and then optimized through our learning framework. In this way, we can freely model the relationships between different groups in the hidden space using their embedding vectors. Then, by regarding these group embeddings as prototypes, we can represent nodes from the group view as a linear combination of the group embeddings. The weights for the combination are learned by a learnable group assignment matrix. Similarly, graph pooling methods [21, 35] designed for the unsigned graph classification task also adopt learnable assignment matrices. Specifically, we decompose the procedure into the following four steps.

3.3.1 Assignment Initialization. The goal of this step is to generate the initial assignment probability of nodes to the K groups. We donate the initial assignment matrix as $\mathbf{S}^{(0)} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}|\times K}$, where $\mathbf{S}^{(0)}_{v,C_i}$ is the probability of node v belong to the i^{th} group. To calculate $\mathbf{S}^{(0)}$, we first transform the input node features into hidden representations through a multilayer perceptron (MLP), i.e.,

$$\mathbf{X}'_v = \text{MLP}(\mathbf{X}_v), \forall v \in \mathcal{V} \tag{1}$$

where $\mathbf{X}'_v \in \mathbb{R}^{d_G}$. The parameters for different nodes are shared in the MLP and the output dimensionality matches the group embedding matrix. Next, we calculate the inner product, i.e., a simple attention value $\mathbf{Q}^{(0)}_{v,C_i}$ between the representation of node v and the i^{th} group embedding:

$$\mathbf{Q}^{(0)}_{v,C_i} = \mathbf{Z}^T_{C_i}\mathbf{X}'_v. \tag{2}$$

We then normalize the initial attention values with the softmax function to get the initial assignment probability

$$\mathbf{S}^{(0)}_{v,C_i} = \frac{\exp\left(\mathbf{Q}^{(0)}_{v,C_i}\right)}{\sum_{j=1}^{K}\exp\left(\mathbf{Q}^{(0)}_{v,C_j}\right)}. \tag{3}$$

3.3.2 Assignment Propagation and Aggregation. Simply adopting $\mathbf{S}^{(0)}$ cannot model the complex relationships between nodes. Following the message-passing framework of GNNs, we next conduct assignment propagation and aggregation to update the assignment matrix while incorporating the neighborhood information of nodes. In the message-passing mechanism, the aggregation function plays an important role in obtaining high-quality representations. Previous works of signed GNNs usually adopt the mean aggregation function [5, 13]. However, we find that the mean aggregation function cannot effectively model the complex neighborhood structure of nodes in signed graphs. We show a toy example in Figure 3. We can easily see that the neighborhoods of node a and b are different. However, the mean aggregator cannot differentiate these two cases and thus produce identical neighborhood messages. Similar phenomenons are observed in unsigned GNNs and it is shown that the sum aggregator can achieve the maximum expressiveness [32]. Therefore, we adopt the sum aggregator instead of the mean aggregator in all our message-passing layers.

We aggregate positive and negative neighbors separately

$$\mathbf{p}^{(m)}_v = \sum_{u\in\mathcal{N}^+_v}\mathbf{S}^{(m-1)}_u, \mathbf{n}^{(m)}_v = \sum_{u\in\mathcal{N}^-_v}\mathbf{S}^{(m-1)}_u, \tag{4}$$

where $\mathbf{p}^{(m)}_v$ and $\mathbf{n}^{(m)}_v$ denotes the information for node v from positive and neighbors neighbors in the m^{th} layer, respectively.

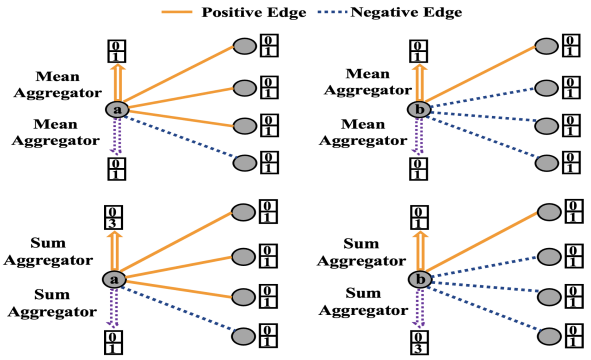


Figure 3: A toy example when the sum aggregator outperforms the mean aggregator. Though node a and b have different neighborhood structures, the mean aggregator cannot differentiate these two nodes, while the sum aggregator can.

3.3.3 Assignments Updating. After aggregating the messages from neighborhoods, we update the assignment matrix of node v by fusing three sources of information: the assignment matrix in the last layer, the information from positive neighbors, and the information from negative neighbors. Specifically, we set

$$\begin{aligned}\mathbf{S}_v^{(m)} &= \mathcal{F}^{(m)}\left(\mathbf{S}_v^{(m-1)}, \mathbf{p}_v^{(m)}, \mathbf{n}_v^{(m)}\right) \\ &= \text{softmax}\left(\sigma\left(\left[\mathbf{S}_v^{(m-1)}, \mathbf{p}_v^{(m)}, \mathbf{n}_v^{(m)}\right]\mathbf{W}_G^{(m)\prime}\right)\mathbf{W}_G^{(m)}\right)\end{aligned} \tag{5}$$

where $\mathbf{W}_G^{(m)\prime} \in \mathbb{R}^{3K\times d_{\text{hidden}}}$, $\mathbf{W}_G^{(m)} \in \mathbb{R}^{d_{\text{hidden}}\times K}$ are learnable weight matrices, d_{hidden} is the dimensionality of the hidden representation, and σ is ac activation function. We empirically find that additional bias terms do not lift the performance and thus remove the bias terms to reduce the number of parameters.

3.3.4 Obtaining Global Representations. After repeating the steps in Section 3.3.2 and Section 3.3.3 M_G times, i.e., adopting M_G message-passing layers, we have obtained the final assignment matrix $\mathbf{S} = \mathbf{S}^{(M_G)}$. We calculate the global representation of nodes

$$\mathbf{Z}_G = \mathbf{S}\mathbf{Z}_C, \tag{6}$$

i.e., the global representation is a linear combination of the group embeddings $\mathbf{Z}_C$ with weights defined in the assignment matrix $\mathbf{S}$.

Notice that in the global signed convolutional module, the number of parameters is linear with respect to the number of groups K and the dimensionality of hidden representations d_G and d_{hidden}, but is not related to $|\mathcal{V}|$.

3.4 Local Signed Convolutional Module

In the local module, we do not adopt any assumption on the relations between positive and negative links. Instead, we simply treat self connections, positive links, and negative links as three independent relations. In this way, we can capture information not modeled by the K-group theory, such as structures within each group or information more related to node features. Such a module is important to complement the global module, especially when the real-world signed graphs do not strictly follow the K-group theory.

We also adopt message-passing layers in learning the local representations. Specifically, we denote the hidden representation of node v in the m^{th} layer of the local module as $\mathbf{z}_v^{(m)}$. The node representations are updated by aggregating information from neighborhoods as follows.

$$\begin{aligned}\mathbf{h}_v^{(m)} &= \left[\sum\nolimits_{u\in\mathcal{N}_v^+}\mathbf{z}_u^{(m-1)}, \sum\nolimits_{u\in\mathcal{N}_v^-}\mathbf{z}_u^{(m-1)}\right] \\ \mathbf{z}_v^{(m)} &= \sigma\left(\left[\mathbf{z}_v^{(m-1)}, \mathbf{h}_v^{(m)}\right]\mathbf{W}_L^{(m)} + \mathbf{b}_L^{(m)}\right),\end{aligned} \tag{7}$$

where $\mathbf{h}_v^{(m)}$ is the message of node v by concatenating the information from positive neighbors and negative neighbors, $\mathbf{W}_L^{(m)}$ and $\mathbf{b}_L^{(m)}$ are learnable parameters for the local module, and $\sigma(\cdot)$ is the activation function. Note that, the aggregating functions used in the local module are also the sum function.

After conduct message-passing for M_L layers, the final local embeddings of nodes are

$$\mathbf{Z}_L = \left[\mathbf{z}_1^{(M_L)}, \mathbf{z}_2^{(M_L)}, ..., \mathbf{z}_{|\mathcal{V}|}^{(M_L)}\right]. \tag{8}$$

3.5 Objective Function and Discussions

Finally, we obtain the node embedding by concatenating global and local embeddings

$$\mathbf{Z} = [\mathbf{Z}_G, \mathbf{Z}_L]. \tag{9}$$

It is easy to see that our model can be trained end-to-end for a given task. In this paper, we adopt link sign prediction, which is a widely adopted task in signed graph representation learning. Specifically, link sign prediction is a binary edge classification problem. Therefore, we use the binary cross-entropy (BCE) loss as follows.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{E}^+\cup\mathcal{E}^-|}\left(\sum_{(u,v)\in\mathcal{E}^+}\log p(u,v) + \sum_{(u,v)\in\mathcal{E}^-}(1-\log p(u,v))\right), \tag{10}$$

where $p(u,v)$ denotes the probability of node u and v forms an link. We use another MLP as the predictor with node embeddings as inputs:

$$p(u,v) = \text{Sigmoid}\left(\text{MLP}\left([\mathbf{Z}_u, \mathbf{Z}_v]\right)\right). \tag{11}$$

The final objective function $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda\mathcal{L}_{\text{reg}}, \tag{12}$$

where $\mathcal{L}_{\text{reg}}$ is the L_2 regularization term and λ is a hyper-parameter to control the weights of the regularization.

4 EXPERIMENTS

In this section, we conduct experiments to demonstrate the merits of GS-GNN. We aim to answer the following questions:

- **Q1**: Can GS-GNN fully utilize the k-group theory and discover the underlying structure of signed graphs?
- **Q2**: How does GS-GNN perform compared with other state-of-the-art signed graphs representation learning methods?
- **Q3**: Do both the global and local representation contribute to our proposed GS-GNN method and how do essential parameters affect the model?

To answer **Q1**, we first conduct experiments on synthetic datasets. Then, we compare GS-GNN with various baselines on four real signed graphs to answer **Q2**. Finally, ablation studies and parameter studies are reported to answer **Q3**.

Table 2: The statistics of datasets. We count each undirected link once in calculating the number of links.

Datasets	# Nodes	# Links	# Positive Links	# Negative Links
Bitcoin-Alpha	3,775	14,120	12,721	1,399
Bitcoin-OTC	5,875	21,489	18,230	3,259
Slashdot	37,626	419,072	313,543	105,529
Epinions	45,003	616,031	513,851	102,180

4.1 Experimental Settings

4.1.1 Real-world Datasets. We adopt the following four public signed graphs in our experiments:

- **Bitcoin-Alpha**[2][20] and **Bitcoin-OTC**[3][20] are two signed graphs extracted from bitcoin trading platforms. Since bitcoin

[2]http://www.btc-alpha.com
[3]http://www.bitcoin-otc.com

Table 3: The results of different methods on the synthetic dataset. K_S is the ground-truth number of groups in generating the synthetic graph and K is the assumed number of groups, i.e., a hyper-parameter, in our GS-GNN and SPONGE.

Assumption		Method	$K_S = 2$	$K_S = 3$	$K_S = 4$	$K_S = 5$	$K_S = 6$
Balance Theory		SGCN	0.442	0.398	0.362	0.334	0.357
		SGDN	0.791	0.682	0.612	0.530	0.495
K-Group	K=K_S	SPONGE	0.983	0.989	0.990	0.990	0.989
		GS-GNN	0.984	0.991	0.991	0.989	0.982
	K=2	SPONGE	0.983	0.853	0.749	0.670	0.600
		GS-GNN	0.984	0.991	0.988	0.984	0.889
	K=6	SPONGE	0.463	0.662	0.848	0.940	0.989
		GS-GNN	0.986	0.988	0.990	0.989	0.980

tradings are anonymous, people give trust or not-trust tags to others in order to enhance security.
- **Slashdot**[4][23] is a technology-related news website. Using the Slashdot Zoo feature, users can create friend or foe relationships with others.
- **Epinions**[5][23] is a consumer review site with trust and distrust relationships between users.

We use undirected versions of the datasets which are preprocessed by SNEA [13], i.e., neglecting the directions of links. We summarize the statistics of datasets in Table 2. We can see from the table that the positive and negative links are highly imbalanced in real-world signed graphs.

4.1.2 Task and Evaluation Metrics. Following previous works [5, 13], we adopt link sign prediction as our task, i.e., predicting the polar of the given links. Specifically, we randomly split the edges into a training set and a testing set with the ratio 8:2. For evaluation metrics, since positive and negative links are imbalanced, we adopt the following four metrics: area under curve (AUC), macro-averaged F1 score (Macro-F1), micro-average F1 score (Micro-F1), and binary-average F1 score (Binary-F1) [5]. For all the metrics, higher values indicate better performance. We repeat all experiments five times with random dataset splits and report the average results.

4.1.3 Baselines. We choose the following methods as baselines, including one signed graph clustering method (SPONGE), two signed network embedding methods (SLF, SIDE), one GNN method for unsigned graphs (GCN), and three GNNs for signed graphs (SGCN, SNEA, SGDN).

- **SPONGE** [3]: A k-way clustering method for signed graphs based on a generalized eigen-problem. All clusters are assumed to be in conflicts.
- **SLF** [33]: A signed network embedding method based on signed latent factors.
- **SIDE** [17]: A signed network embedding method that adopts random walks based on the balance theory.
- **GCN** [18]: A representative GNN originally designed for unsigned graphs based on the message passing mechanism. We neglect the signs of the links.
- **SGCN** [5]: It generalizes GCN to signed graphs by learning two node representations based on the balance theory.
- **SNEA** [13]: It adopts attention-based aggregators in message-passing steps and is also based on the balance theory.
- **SGDN** [16]: It proposes a signed random walk diffusion method to consider multi-hop neighbors, which is also based on the balance theory.

For SPONGE, we make link sign prediction based on the cluster assignments of nodes. For methods which are not trained in an end-to-end fashion, we first obtain the node presentations and then adopt a logistic regression as the classifier, as suggested in these papers. For the rest of the baselines, we adopt the standard end-to-end training.

4.1.4 Hyper-parameters. For SPONGE, we grid search the cluster number from $\{2, 3, ..., 20\}$. For other baselines and our proposed GS-GNN, we uniformly set the dimensionality of node representations as 64 (for GS-GNN, it includes both modules, i.e., $d_G + d_L = 64$). For all the baselines, other hyper-parameters are set as suggested in their papers. For our proposed method, we search the number of layers from $\{2, 3, 4, 5\}$ for the local module and from $\{2, 3, 4, 5, 6, 7, 8\}$ for the global module. We adopt the Adam optimizer with both the learning rate and weight decay searched from $\{0.01, 0.005, 0.001\}$. As no node feature is available, we follow previous works [13, 16] and use the top 64-dimensional vectors of truncated Singular Value Decomposition[6] as node features.

4.2 Results on Synthetic Dataset

Before reporting the results of real-world datasets, we first conduct experiments on the synthetic dataset aiming to answer **Q1**, i.e., verifying our proposed GS-GNN can indeed model the underlying k-group structures of signed graphs. We generate synthetic signed graphs using the signed stochastic block model (SSBM)[34] as follows. A set of nodes $\mathcal{V}_S$ are divided into a set of K_S groups with equal sizes, i.e., each group has $\frac{|\mathcal{V}_S|}{K_S}$ nodes. Then, nodes within the same group have a probability p_{pos} to form a positive edge. All groups are assumed to be in conflicts with each other and thus nodes in different groups have a probability p_{neg} to form a negative edge. To further taking noises into consideration, the signs of all the links have a probability p_{noise} to be flipped. For the detailed parameters for our adopted SSBM, we set the number of nodes as $|\mathcal{V}_S| = 3,000$, the number of groups K_S is varied from $\{2, 3, 4, 5, 6\}$, $p_{\text{pos}} = 0.05$, $p_{\text{neg}} = 0.0125$, and $p_{\text{noise}} = 0.01$.

For brevity, we only adopt SPONGE [3], SGCN [5], and SNEA [13] that are most competitive as the baselines in this setting. Since both

[4]http://www.slashdot.com
[5]http://www.epinions.com

Table 4: The results of link sign prediction on real-world signed datasets. The best results and the second-best results for each dataset using each metric are in bold and underlined, respectively. The last column is the relative improvement of GS-GNN compared to the best-performing baseline.

Dataset	Metric	SPONGE	SLF	SIDE	GCN	SGCN	SNEA	SGDN	GS-GNN	
Bitcoin-Alpha	AUC	0.513	0.847	0.797	0.806	0.858	0.866	0.840	**0.893**	+3.12%
	Macro-F1	0.504	0.668	0.665	0.546	0.706	0.727	0.663	**0.793**	+9.08%
	Micro-F1	0.901	0.819	0.824	0.902	0.864	0.873	0.894	**0.930**	+3.10%
	Binary-F1	0.948	0.892	0.896	0.948	0.921	0.926	0.942	**0.961**	+1.37%
Bitcoin-OTC	AUC	0.700	0.873	0.828	0.845	0.871	0.863	0.863	**0.915**	+4.81%
	Macro-F1	0.644	0.735	0.713	0.675	0.754	0.760	0.734	**0.837**	+10.13%
	Micro-F1	0.763	0.828	0.820	0.875	0.850	0.858	0.871	**0.920**	+5.14%
	Binary-F1	0.850	0.892	0.889	0.928	0.908	0.914	0.926	**0.952**	+2.59%
Slashdot	AUC	0.500	0.888	0.820	0.819	0.873	0.888	0.887	**0.916**	+3.15%
	Macro-F1	0.432	0.772	0.725	0.670	0.760	0.769	0.769	**0.812**	+5.18%
	Micro-F1	0.752	0.812	0.773	0.797	0.802	0.812	0.838	**0.865**	+3.22%
	Binary-F1	0.861	0.867	0.840	0.875	0.859	0.868	0.896	**0.915**	+2.12%
Epinions	AUC	0.508	0.928	0.878	0.869	0.925	0.931	0.930	**0.959**	+3.01%
	Macro-F1	0.474	0.795	0.746	0.685	0.800	0.819	0.819	**0.865**	+5.62%
	Micro-F1	0.832	0.865	0.829	0.864	0.872	0.888	0.903	**0.931**	+3.10%
	Binary-F1	0.908	0.915	0.891	0.922	0.920	0.931	0.942	**0.961**	+2.02%

our proposed GS-GNN and SPONGE have the number of groups K as a hyper-parameter, we consider three settings: (1) The ground-truth K_S is known to the method, i.e., $K = K_S$; (2) We set the number of groups as the smallest possible value of the ground-truth number, i.e., $K = 2$; (3) We set the number of groups as the largest possible value of the ground-truth number, i.e., $K = 6$. We only report the results using Macro-F1 as the metric, while other metrics show similar patterns.

The results are shown in Table 3. We have the following findings. In all three settings, both GS-GNN and SPONGE greatly outperform SGCN and SGDN. Recall that SGCN and SGDN are based on the balance theory, and SPONGE and our GS-GNN are based on the k-group theory. The results clearly show that balance theory fails to model the generative mechanism of the synthetic dataset and the k-group theory is strictly more general. Notice that even for $K_S = 2$, SPONGE and GS-GNN outperform SGCN and SGDN (though the margins are smaller than other K_S), indicating that the previous signed GNNs even cannot fully utilize the balance theory.

In the first setting, i.e., when the ground-truth K_S is known, both GS-GNN and SPONGE report nearly perfect results, demonstrating that they can discover the underlying structure of the synthetic dataset. Notice that SPONGE has assumed that all groups are in conflicts and form negative links only, which is consistent with our generative mechanism. On the other hand, GS-GNN does not have such an assumption and only infer from the observed data.

In the second setting, i.e., when the number of groups is assumed minimum, GS-GNN generally outperforms SPONGE. The improvement of GS-GNN over SPONGE grows larger as there are more ground-truth groups. Even when the number of ground-truth groups is more than twice as many as the assumption in our model, i.e., $K_S = 5$ and $K = 2$, the performance of GS-GNN is nearly perfect. The results indicate that SPONGE cannot discover more groups than its assumption, while our proposed GS-GNN is much more capable. We attribute this advantage to fusing the k-group theory and representation learning, which greatly enhances the model capacity. For example, when $K < K_S$, our proposed GS-GNN can possibly learn to decompose different dimensions of one group representation to model more than one group.

In the third setting, i.e., when the number of groups is assumed maximum, GS-GNN again shows better performance compared to SPONGE in most cases. In this setting, the margin between GS-GNN and SPONGE has a negative relation with the ground-truth number of groups. The results indicate that SPONGE cannot function well if its assumed groups outnumber the ground-truth. In contrast, the results of GS-GNN are impressively stable, showing that our proposed method is not sensitive to the setting of K (more results of parameter sensitivity on real graphs are provided in Section 4.5).

The above results well demonstrate the superiority of GS-GNN in modeling the synthetic dataset and utilizing the k-group theory, which is much more general than the usual assumption of balance theory. Besides, GS-GNN even outperforms the SPONGE, which uses the ground-truth group number as input.

4.3 Results on Real Graphs

Next, we present the experimental results on real signed graphs. The performance of all the methods for the link sign prediction task is shown in Table 4. We have the following observations. GS-GNN consistently outperforms all the baselines on all datasets with all evaluation metrics. On average, the improvement of GS-GNN compared to the most competent baseline is approximately 5% and in some cases, the improvement can be as large as 10%. The results clearly demonstrate the superiority of GS-GNN and the importance of extending the balance theory to the k-group theory as well as jointly learning the global and local node representations.

Notice that the positive links and negative links are imbalanced and thus different evaluation metrics focus on different aspects of the prediction results. For example, though SNEA achieves impressive results in terms of AUC and Macro-F1, SGDN and GCN report

better results when we adopt Micro-F1 and Binary-F1. Nevertheless, GS-GNN reports the best results in all aspects. The performance gains of GS-GNN are especially significant in terms of Macro-F1. Since Macro-F1 treats two classes equally and there are considerably fewer negative links, the results indicate that GS-GNN can better model the negative links, which are more vital information sources in signed graphs.

The performance of a representative unsigned graph representation learning methods GCN is worse than our proposed GS-GNN , indicating that modeling the sign of links can be beneficial. However, GCN outperforms SGCN in some cases, showing that the capability of SGCN is somewhat limited by the simple assumption of balance theory.

SPONGE does not work as well as in the synthetic graph, especially in terms of the Macro-F1 metric. The results show that the formation of real-world signed graphs is considerably more complicated than in our synthetic graph. For example, the relationships between different groups may be far more than simple conflicts. Compared to SPONGE, our proposed GS-GNN has more freedom to model such structures, thanks to the dual architecture of jointly learning global and local representations and the integration of k-group theory deep representation learning models.

4.4 Ablation Studies

4.4.1 The global and the local module. To verify our dual architecture design of jointly learning the global and local representations, we design two variants of GS-GNN . In GS-GNN$_L$, we only learn the local representation, $\mathbf{Z}_L$, as the final node representation, and in GS-GNN$_G$, we only utilize the global representation, $\mathbf{Z}_G$, for the downstream task. Other experimental settings are the same as Section 4. We only report the results on Bitcoin-Alpha and Bitcoin-OTC for brevity in Table 5. Overall, both modules contribute to our proposed GS-GNN, showing that the local and global representations of nodes are complementary.

Table 5: The ablation study results of only using the global and the local component of GS-GNN.

Dataset	Metric	GS-GNN$_L$	GS-GNN$_G$	GS-GNN
Bitcoin-Alpha	AUC	0.875	0.889	0.893
	Macro-F1	0.754	0.731	0.793
	Micro-F1	0.922	0.916	0.930
	Binary-F1	0.958	0.954	0.961
Bitcoin-OTC	AUC	0.891	0.906	0.915
	Macro-F1	0.801	0.786	0.837
	Micro-F1	0.901	0.899	0.920
	Binary-F1	0.946	0.942	0.952

4.4.2 The aggregation function. We also conduct ablation studies on the adopted aggregation function. As analyzed in Section 3.3.2, the sum aggregator is more expressive than the mean aggregator. To provide empirical analyses, we replace all the aggregation functions in GS-GNN with the mean aggregator and name such a method as GS-GNN$_{\text{mean}}$. The results are shown in Table 6. GS-GNN $_{\text{sum}}$ greatly outperforms GS-GNN$_{\text{mean}}$, verifying our analyses and demonstrating the importance of using the sum aggregator for positive and negative neighbors separately in signed graphs.

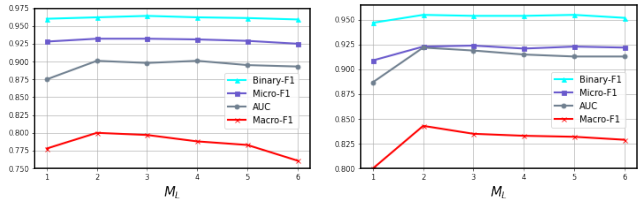
Table 6: The ablation study results of using different aggregators for GS-GNN.

Dataset	Metric	GS-GNN$_{\text{mean}}$	GS-GNN$_{\text{sum}}$
Bitcoin-Alpha	AUC	0.844	0.893
	Macro-F1	0.712	0.793
	Micro-F1	0.915	0.930
	Binary-F1	0.954	0.961
Bitcoin-OTC	AUC	0.900	0.915
	Macro-F1	0.812	0.837
	Micro-F1	0.914	0.920
	Binary-F1	0.950	0.952

4.5 Parameter Sensitivities

In this section, we investigate the parameter sensitivities of our GS-GNN. There are three important parameters in our proposed method: the number of layers in the local module M_L, the number of layers in the global module M_G, and the number of groups K. We conduct three controlled experiments to investigate the effects of three parameters. For brevity, we only report the results on Bitcoin-OTC and Bitcoin-Alpha.

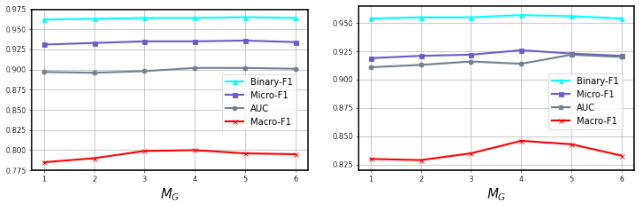
4.5.1 The number of layers in the local module M_L. We fix M_G to 5, K to 3, and vary M_L from $\{2, 3, 4, 5, 6\}$. The results are shown in Figure 4. Overall, $M_L = 2$ is a suitable choice for both datasets, indicating that we do not need a deep architecture for the local module. A plausible reason for the performance decay as the local module grows deeper is the over-smoothing problem widely known in GNNs [24, 36]. We leave improving the local module using studies to alleviate the over-smoothing problem as future works.



(a) Bitcoin-Alpha (b) Bitcoin-OTC

Figure 4: The parameter sensitivity results of varying M_L.

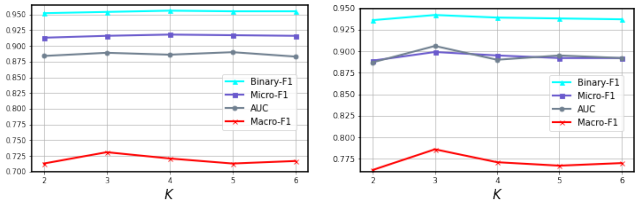
4.5.2 The number of layers in the global module M_G. We fix M_L to 2 and K to 3, and vary M_G from $\{2, 3, 4, 5, 6\}$. The results are shown in Figure 5. Contrary to the local module, the performance of GS-GNN increases as M_G grows larger. We achieve the best performance when M_G is approximately 5. The results show that a deeper architecture is needed for the global module. As for why the global module is not affected by the over-smoothing problem, we attribute this merit to the design of a prototype-based GNN as the global module, which has a mechanism different from the existing GNNs.



(a) Bitcoin-Alpha (b) Bitcoin-OTC

Figure 5: The parameter sensitivity results of M_G.

4.5.3 The number of groups K. To better understand the effectiveness of K, we only adopt the global module here and fix M_G as 5, and vary K from $\{2, 3, 4, 5, 6\}$. The results shown in Figure 6 suggest a consistent pattern as in the synthetic graph that our proposed GS-GNN is not sensitive to the setting of K. In general, setting K from 3 to 5 leads to the best results.



(a) Bitcoin-Alpha (b) Bitcoin-OTC

Figure 6: The parameter sensitivity results of K.

5 CONCLUSION

In this paper, we propose a novel GS-GNN model for signed graph representation learning beyond the usual balance theory assumption. Specifically, we adopt a dual GNN architecture, with the global module learning node representations based on the k-group theory, a more generalized assumption than the balance theory, and the local module complementing the global module using a relational GNN with no prior assumption. Extensive experimental results on both synthetic and real-world signed graphs demonstrate the effectiveness of GS-GNN by achieving new state-of-the-art.

ACKNOWLEDGEMENTS

This work was supported in part by National Key R&D Program of China (No. 2018AAA0102004, No. 2020AAA0106300), National Natural Science Foundation of China (No. U1936219, 61521002, 61772304), Beijing Academy of Artificial Intelligence (BAAI), Meituan and a grant from the Institute for Guo Qiang, Tsinghua University. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. Peng Cui and Yafeng Zhang are corresponding authors.

REFERENCES

[1] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep neural networks for learning graph representations. In *AAAI*.
[2] Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S Dhillon. 2012. Scalable clustering of signed networks using balance normalized cut. In *CIKM*.
[3] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. 2019. SPONGE: A generalized eigenproblem for clustering signed networks. In *AISTATS*.
[4] Tyler Derr, Charu Aggarwal, and Jiliang Tang. 2018. Signed network modeling based on structural balance theory. In *CIKM*.
[5] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed graph convolutional networks. In *ICDM*.
[6] Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* (1936).
[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*.
[8] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.
[9] Frank Harary et al. 1953. On the notion of balance of a signed graph. *Michigan Mathematical Journal* (1953).
[10] Fritz Heider. 1982. *The psychology of interpersonal relations*.
[11] Ivan Herman, Guy Melançon, and M Scott Marshall. 2000. Graph visualization and navigation in information visualization: A survey. *TVCG* (2000).
[12] Paul W Holland and Samuel Leinhardt. 2013. *Perspectives on social network research*. Elsevier.
[13] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. 2019. Signed graph attention networks. In *ICANN*.
[14] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. 2021. SDGNN: Learning Node Representation for Signed Directed Networks. In *AAAI*.
[15] Mohammad Raihanul Islam, B Aditya Prakash, and Naren Ramakrishnan. 2018. Signet: Scalable embeddings for signed networks. In *PAKDD*.
[16] Jinhong Jung, Jaemin Yoo, and U Kang. 2020. Signed Graph Diffusion Network. *arXiv:2012.14191* (2020).
[17] Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang. 2018. Side: representation learning in signed directed networks. In *WWW*.
[18] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
[19] Andrew V Knyazev. 2017. Signed Laplacian for spectral clustering revisited. *arXiv:1701.01394* (2017).
[20] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. 2016. Edge weight prediction in weighted signed networks. In *ICDM*.
[21] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-Attention Graph Pooling. In *ICML*.
[22] Yeon-Chang Lee, Nayoun Seo, Kyungsik Han, and Sang-Wook Kim. 2020. Asine: Adversarial signed network embedding. In *SIGIR*.
[23] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *SIGCHI*.
[24] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into gvolutional networks for semi-supervised learning. In *AAAI*.
[25] Yu Li, Yuan Tian, Jiawei Zhang, and Yi Chang. 2020. Learning signed network embedding via graph attention. In *AAAI*.
[26] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.
[27] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* (2008).
[28] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*.
[29] Ruo-Chun Tzeng, Bruno Ordozgoiti, and Aristides Gionis. 2020. Discovering conflicting groups in signed networks. In *NeurIPS*.
[30] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *KDD*.
[31] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. 2017. Signed network embedding in social media. In *ICDM*.
[32] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *ICLR*.
[33] Pinghua Xu, Wenbin Hu, Jia Wu, and Bo Du. 2019. Link prediction with signed latent factors in signed social networks. In *KDD*.
[34] Bo Yang, Xuehua Zhao, and Xueyan Liu. 2015. Bayesian approach to modeling and detecting communities in signed network. In *AAAI*.
[35] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *NeurIPS*.
[36] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *TKDE* (2020).
[37] Chenyi Zhuang and Qiang Ma. 2018. Dual graph convolutional networks for graph-based semi-supervised classification. In *WWW*.

A EXPERIMENTAL DETAILS FOR REPRODUCIBILITY

A.1 Datasets and Codes

We adopt the following publicly available source codes and datasets.

- Datasets: https://github.com/liyu1990/data
- SPONGE [3]: https://github.com/alan-turing-institute/SigNet
- SLF [33]: https://github.com/WHU-SNA/SLF
- SIDE [17]: https://datalab.snu.ac.kr/side/resources/side.zip
- GCN [18]: https://github.com/tkipf/pygcn
- SGCN [5]: https://github.com/benedekrozemberczki/SGCN
- SNEA [13]: https://github.com/liyu1990/snea
- SGDN [16]: https://openreview.net/attachment?id=YPm0fzy_z6R&name=supplementary_material

A.2 Software and Hardware Configurations

All experiments are conducted on a server with the following configurations.

- Operating System: Ubuntu 18.04.1 LTS
- CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz
- GPU: GeForce GTX TITAN X
- Software: Python 3.7.9, Cuda 10.2, PyTorch 1.7.1, TenfowFlow 1.13.2