Status-Aware Signed Heterogeneous Network Embedding with Graph Neural Networks

Wanyu Lin, *Member, IEEE,* Baochun Li, *Fellow, IEEE*

Abstract—Many real-world applications are inherently modeled as signed heterogeneous networks or graphs with positive and negative links. Signed graph embedding embeds rich structural and semantic information of a signed graph into low-dimensional node representations. Existing methods usually exploit social structural balance theory to capture the semantics of the complex structure in a signed graph. These methods either omit the node features or may discard the direction information of the links. To address these issues, we propose a new framework, called Status-Aware Graph Neural Network (*S-GNN*), to boost the representation learning performance. *S-GNN* is equipped with a loss function designed based on *status theory*, a social-psychological theory specifically developed for directed signed graphs. Extensive experimental results on benchmarking datasets verified that *S-GNN* can distill comprehensive information ingrained in a signed graph in the embedding space. Specifically, *S-GNN* achieves state-of-the-art accuracy, robustness, and scalability: it speeds up the processing time of link sign prediction by up to $6.5\times$ and increases accuracy by up to 18.8% as compared with the alternatives. We also show that *S-GNN* can obtain effective status scores of nodes for link sign prediction and node ranking tasks, both of which yield state-of-the-art performance.

Index Terms—Signed Heterogeneous Information Networks, Signed Network Embedding/Representation Learning, Status Theory

1 Introduction

In many real-world applications, relationships between two entities or users have opposite properties, such as trust-distrust relationships in trust networks [1], friend-foe relationships in social networks [2], and support-dissent opinions in opinion networks [3]. These networks can be represented as graphs with positive and negative links or edges, which we refer to as *signed networks* or *signed graphs*. The link heterogeneity (with positive and negative links) conveys a rich set of information that can be leveraged to enhance network mining tasks, such as link sign prediction, node ranking, and community detection [4]. In particular, predicting the sign of links is a fundamental task in many areas, such as personal advertising and public opinion analysis.

With the prevalence of signed networks, extensive work on representation learning on signed networks has been reported in the literature. Signed graph embedding, or representation learning, aims to learn a low-dimensional representation for each node. Such a representation can then be applied to downstream mining tasks on signed graphs. With relatively few exceptions (e.g., BESIDE [5]), research on signed graph embedding has focused on characterizing social structural balance theory (e.g., SIGNet [6], SIDE [7], SiNE [8], SNE [9]), which defines an organizing principle for the complex structure on signed graphs and implies that cycles with an even number of negative links are more plausible, thus should be more prevalent in real-world networks (e.g., a balanced triangle with zero negative shown in Fig. 1a and balanced triangle with two negatives shown in Fig. 1b). However, structural balance theory omits the direction information of links on signed graphs [2] and may lose some aspects of information, leading to suboptimal performance.

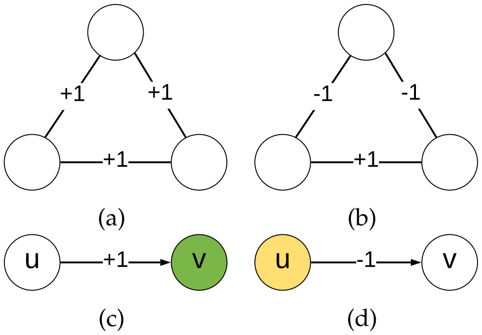


Fig. 1: (a) (b) Balanced triangles; (c) (d) Status theory illustration.

In recent years, we have witnessed encouraging developments in deep graph convolutional neural networks (GNNs) for representation learning on unsigned networks (networks contain one type of link only) [10]–[13]. GNNs have been proved to be capable of effectively and efficiently encoding the structural and node feature information in a graph for those unsigned networks. More specifically, feature information from local graph neighborhoods is iteratively aggregated. By stacking multiple convolutions, local information can be propagated throughout the entire network. Thus, given their advantages, excellent opportunities may exist in using GNNs to capture the complex structural and semantic information in a signed graph.

- W. Lin is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: wanylin@comp.polyu.edu.hk.
- B. Li is with the Department of Electrical and Computer Engineering, University of Toronto, Ontario, M5S 1A1, Canada. Email: bli@ece.toronto.edu.

Yet, representation learning in signed networks using graph convolutional neural networks is quite challenging. GNNs were originally designed to capture the homophily nature of unsigned networks [10], [13], indicating that more similar users are more likely to connect. The notion of homophily may not be applicable when both positive and negative links are jointly considered for signed graph embedding. In the context of signed graphs, the first challenge is how to use GNNs to encode the heterogeneous relationships into a low-dimensional vector space. Though a few GNN-based methods have been proposed to address the problem of representation learning on signed graphs (e.g., SGCN [14], SNEA [15]), they were built on structural balance theory where the direction information of links are not explicitly aggregated and propagated, and thus may not work very well. Therefore, the second challenge is effectively characterizing the direction information of links in a signed graph.

To address these challenges, we propose a novel Status-Aware Graph Neural Network (*S-GNN*) for signed graph embedding. *S-GNN* addresses all the issues described above by applying link semantics transformation, exploiting *status theory*, and performing status aggregation and propagation in two ways to generate node embeddings. Status theory is a fundamental theory from social psychology, which provides an organizing principle of the rich structural and semantic information for signed directed networks [2]. Specifically, we leverage status theory to design a new objective function. In the theory of status [2], a positive link $e_{u\rightarrow v}$ implies that v has a higher status from the perspective of u (shown in Fig. 1c), while a negative link $e_{u\rightarrow v}$ indicates that v is regarded as having a lower status, as shown in Fig. 1d.

The key component of *S-GNN* is the status convolutional layer, which employs the notion of localized graph convolutions [10], [13]. To capture the direction information of links, each of our status convolutional layers consists of two components: *receptive-based status* aggregation and *generative-based status* aggregation. A plausible analogy for these two components can be represented in the context of social networks: receptive-based status aggregation is used to characterize the status that a user endorsed by the others (e.g., the popularity of the user), while generative-based status aggregation is for capturing the extent that a user is willing to endorse to the others (e.g., the deference of the user to the others). By doing so, *S-GNN* can distill comprehensive information from the heterogeneous relationships in a signed graph based on the theory of status. Further, by stacking fully-connected layers, *S-GNN* can obtain a status score for each node. Therefore, the relative relationship between any two entities can also be established with their status scores. Specifically, *S-GNN* is general as compared to most prior methods in the sense that it can deal with the link sign prediction and node ranking tasks simultaneously.

Highlights of our original contributions are as follows. *First*, we introduce a principled methodology to capture the rich structural and semantic information of signed graphs. *Second*, we design two types of status convolutions to capture the status of each entity such that the heterogeneous relationships can be captured in the embedding space. *Third*, we design an objective function for signed heterogeneous network embedding by leveraging *status theory*. *Fourth*, we demonstrate the effectiveness and efficiency of our proposed framework using four signed directed networks from different domains. Our extensive array of experiments on benchmarking datasets demonstrated that *S-GNN* can speedup representation learning for link sign prediction by up to $6.5\times$ as compared with the alternatives. More specifically, *S-GNN* achieves up to a $4\times$ speedup, as well as comparable accuracy as compared to BESIDE [5]. *S-GNN* also increases its accuracy by up to 18.8% compared with SIDE [7], and achieves state-of-the-art robustness and scalability compared to the literature. We also show that *S-GNN* can learn effective status scores of each node, which can be used for link sign prediction and node ranking and yield state-of-the-art performance.

2 PRELIMINARIES

Below, we introduce some necessary definitions to facilitate a better understanding of the problem and our proposed solution.

Definition 1. Social status: Status is broadly defined as the position of users, either communities or individuals, in a social hierarchy that results from accumulated acts of deference. The sociologists have widely recognized that status is fundamentally rooted in the accumulation of deference behaviors [16], [17]. In signed social networks, social status can be represented in many different ways, such as the rankings of nodes in social networks, and it represents the prestige/trustworthiness of nodes [4].

Definition 2. Status theory: Status theory defines an organizing principle for signed links on signed directed networks. In the theory of status [2], a positive link $e_{u\rightarrow v}$ implies that v has a higher status from the perspective of u (shown in Fig. 1c), while a negative link $e_{u\rightarrow v}$ indicates that v is regarded as having a lower status, as shown in Fig. 1d. In signed networks, these relative levels of status can be propagated and aggregated throughout the networks.

Definition 3. Structural balance theory: Balance theory [18] classifies cycles in a signed network as being balanced or unbalanced. It implies that cycles with even negative signs are more plausible and should be more prevalent in real networks. For simplicity, we illustrate balanced structures with triangles. More specifically, balanced triangles with three positives, shown in Fig. 1a, capture the notion that "the friend of my friend is my friend," while those with two negatives, shown in Fig. 1b, implies that "the enemy of my enemy is my friend". Balance theory was initially developed for undirected networks.

Comparisons of two theories. Status theory and balance theory both provide insights into ways in which users use linking mechanisms in social computing applications. Status theory is specialized in directional links, as it posits a status differential from the source node of a link to its target node. Balance theory was initially proposed for undirected networks, though it has been widely applied to directed networks (e.g., SIDE [7], SigNet [6]). Structural balance theory can be viewed as modeling like and dislike relationships [2], while in some important domains, such as Epinions and

Wikipedia, a positive link from u to v can be interpreted as "v has a higher status than I do" and a negative link can be viewed as a model of "v has lower status than I do" [19]. In these domains, status theory has been proved to be more expressive than balance theory [2].

3 Related Work

Our problem of signed network embedding connects to a large body of work on network representation learning, signed heterogeneous network embedding, and recent advancements in applying graph convolutional neural networks.

Network Embedding: The goal of network representation learning is to learn low-dimensional representations for all nodes, which can be used for many different network analysis tasks, such as link prediction [20], node classification, and community detection. An extensive amount of work has been developed in this area, including Node2Vec [21], Line [22], DeepWalk [23], and GCN [13], all of which are proposed for unsigned networks.

Signed Network Embedding: In today's social networks, users can express positive or negative attitudes towards others. This link heterogeneity builds a network topology called signed networks. With the prevalence of social media, signed network embedding has emerged as a promising direction that leverages both positive and negative links to enhance network mining performance on signed networks [6]–[9], [15], [24]–[27].

Two social-psychological theories, structural balance theory and status theory have been widely used in mining signed networks [2], [18]. According to the social theories they built on, we can roughly divide existing works into two categories: structural balance theory-based, including SiNE [9], SIDE [7], SIGNet [6], SGCN [14], and SNEA [15]; status theory-based. Most existing works mainly focused on representation learning with structural balance theory. They typically define some pre-defined balanced rules and associated optimization loss terms.

Rare efforts were on signed networks analysis with status theory. SLF [27] modeled four types of links in the latent space, and it is specialized for link prediction tasks. The closest to ours is BESIDE [5] in the sense that both BESIDE [5] and our proposed approach can deal with the link sign prediction and node ranking tasks simultaneously. Specifically, BESIDE first modeled "bridge" edges with status theory and triangles with balance theory and/or status theory. It then adopted a deep neural network to learn from these two pre-defined structures. Different from BESIDE, *S-GNN* captures the semantic information of signed graphs by using the expressive capability of graph neural networks equipped with a loss function designed based on status theory.

Graph Neural Networks (GNNs): GNNs have been proven to be powerful on representation learning on unsigned network graphs [10], [11], [13], leading to new state-of-the-art results on benchmarks such as node classification and link prediction. These GNN-based representation learning on unsigned networks consistently outperformed techniques based upon random walks (e.g, node2vec [21], Line [22], and DeepWalk [23]). Therefore, GNN-based approaches, e.g., SGCN [14], SNEA [15] and SiGAT [26] have been proposed to solve the representation learning on signed networks. SGCN introduced the definitions of balanced and unbalanced paths based on structural balance theory. SNEA [15] was built upon graph attention mechanism (GAT [11]) to capture balance theory, whereas SiGAT [26] incorporates graph motifs into GAT to capture balance theory and status theory simultaneously. These existing GNN-based approaches were proposed specifically for link signed prediction tasks. Unlike prior GNN-based approaches, our proposed framework is more general in that it can cope with link sign prediction tasks and node ranking tasks at the same time.

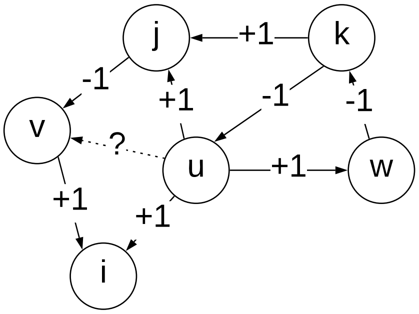


Fig. 2: A signed directed network: an example.

4 Problem Setup

In this paper, we consider a signed directed network/graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}^+$, $\mathcal{E}^-$ represent the sets of positive links and negative links, respectively. A node $v \in \mathcal{V}$ represents an entity in the graph, and a link $e_{u\to v} \in \{\mathcal{E}^+ \cup \mathcal{E}^-\}$ represents a directed link from u to v associated with a positive or negative sign. More precisely, $+1$ represents a positive link and -1 denotes a negative link. As nodes typically represent users in online social networks, we use the terms "node" and "user/entity," "links/edges" and "relationships" interchangeably in this paper. To differentiate direction information between any two nodes, we define the source node as the creator of a link, while the target node as the receiver of a link. This reflects real-world application domains, such as rating (upvote and downvote) web content on Reddit[1] and voting for adminship on Wikipedia (WikiRfA) [28], where both ratings and votes are created by the source nodes of the links.

For any node $u \in \mathcal{V}$, let $N_O(u)$ be the set of out-neighbors of node u, and $N_I(u)$ be the set of in-neighbors of u. In this sense, we can define $|N_I(u)|$ and $|N_O(u)|$ to represent in-degree and out-degree of u, respectively. The mathematical notations used in this paper are summarized in Table 1.

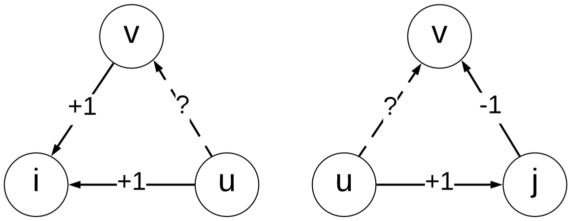
Signed Network Embedding or *Representation Learning:* Given a signed directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$, the task of signed network embedding is to learn a mapping function $f : u \to S[u]$, where $S[u] \in \mathcal{R}^d$ is the learned representation of node u with dimension d. The transformation

[1] https://www.reddit.com

function f preserves the complex structural and semantic information in a signed graph, such that any representations of nodes, in the embedding space, are effective for downstream signed network analysis, e.g., link sign prediction and node ranking. An example of a signed directed network is shown in Fig. 2, which will be used throughout this paper.

TABLE 1: Notations

Notation	Descriptions
$N_O(u)$	the set of out-neighbors of u
$N_I(u)$	the set of in-neighbors of u
G, R	the generative status and receptive status
$S_g[u]$	the latent factor of generative status propagated from the out-neighbors of user u
$S_r[u]$	the latent factor of receptive status propagated from in-neighbors of user u
$S[u]$	the latent factor of status for user u
$\text{sign}_{u\to v}$ $\tilde{\text{sign}}_{u\to v}$	observed sign of link $e_{u\to v}$ predicted sign of link $e_{u\to v}$
$\otimes$	the concatenation operator of two vectors
$\oplus$	mean aggregator
σ	non-linear activation functions, e.g., $\text{sigmoid}(\cdot)$
W, b	the model parameters of *S-GNN*



(a) Predicted as positive (b) Predicted as negative

Fig. 3: Contradictory predictions of $e_{u\to v}$ with balance theory in our example.

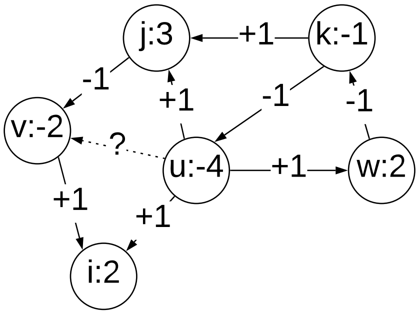


Fig. 4: A simplified status modeling in our example: u has a status of -4 and v has a status of -2. Status theory would, therefore, suggest a positive link of $e_{u\to v}$.

5 Our proposed framework: *S-GNN*

In this section, we first introduce the limitations of balance theory on signed directed networks. Then, we introduce our proposed status-aware graph neural network, called *S-GNN*, a dedicated effort towards the construction of graph neural networks specialized for signed graph embedding.

In particular, we examine the percentage of triangles satisfying balance theory on real-world datasets. There are 92.4% triangles satisfying balance theory on Epinions, 91.7% on Slashdot, while WikiRfA contains 73.6% and WikiElec has 71.2% balanced triangles, respectively. This indicates that signed graph convolution networks based on balance theory may not work very well on signed directed networks. Let's see an example in our signed directed graph, originally shown in Fig. 2. User v links positively to user i and u links positively to user i; balance theory would suggest a positive link of $e_{u\to v}$, as shown in Fig. 3a. However, as shown in Fig. 3b, user u links positively to user j and j links negatively to user v; balance theory would suggest a negative link of $e_{u\to v}$. The predicted sign of link $e_{u\to v}$ based on balance theory with triangles is contradictory.

Before formalizing our status-aware graph neural network, we provide some intuitions behind our construction. Both positive and negative links need to be jointly considered for signed graph embedding in signed networks. The notion of homophily becomes not applicable, which can also be explained by the theory of status. According to the theory of status, the positive in-neighbors and the negative out-neighbors of a user increase its status [2]. In contrast, the positive out-neighbors and negative in-neighbors decrease its status. In this sense, signed graph embedding desires new principles to perform status aggregation and propagation, where the status of each node is determined by its local topological connections.

To illustrate the modeling with status theory in our example, we simplify the status modeling as follows: each user was initially assigned a status of 0. If user u links negatively to another user, or another user links positively to user u, we increase user u a status of 1; otherwise, we decrease the status of u by 1. With the same rule, the status value of each user is assigned as shown in Fig. 4. As u has a status of -4 and v has a status of -2, status theory [2] would suggest a positive link of $e_{u\to v}$. It is worth mentioning that status is propagated and aggregated throughout the networks, which is more complex in the real situation. In what follows, we detail our proposed framework for representation learning on signed networks, the architecture of which is illustrated in Fig. 6.

5.1 Modeling Social Status of Nodes

In this paper, we consider the richness of interactions between entities within signed networks as *social status*. Social status can be represented in many different ways depending on the interpretations of positive and negative links, which are typically distinct across different application domains. In the context of signed social media, status theory allows deriving the relationship of any two users based on their status in the social graph [2].

In particular, to be able to capture the social status of each user, we first separate pairwise interactions of users into two groups: *receptive interactions* and *generative interactions*. Taking the Epinions, a who-trust-whom dataset, as

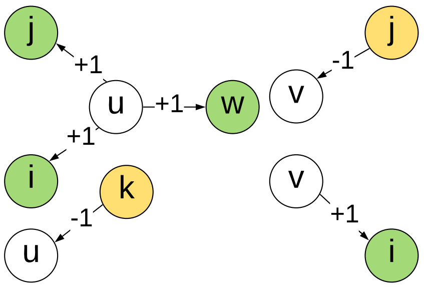


Fig. 5: The receptive interactions (in yellow) and generative interactions (in green): an example.

an example, receptive interactions can be interpreted as the measurement of a user's popularity. In this context, some users are more likely trusted by others, such as the officials, who are referred to as having higher popularity in a society. Therefore, the social status of these users is relatively higher. According to the theory of status, receptive interactions with positives increase the social status of the receiver of the links. In contrast, those with negatives decrease the social status of the receiver.

Similarly, generative interactions are to measure the engagement of a user. In the context of Epinions, users are more likely to trust others with higher social status; hence their social status is relatively lower than those users they trust. Generative interactions with negatives increase the social status of the creator of the link, while those with positives lower the social status of the creator.

5.2 Status Convolutional Layers

Accordingly, we consider two types of status aggregation to characterize the receptive-based status and generative-based status, represented as $S_r[u]$ and $S_g[u]$, respectively. We use a mean-aggregator to aggregate its associated interactions with its neighbors for each of them. It is worth mentioning that, mean-aggregator is the main operation of aggregating information from local graph neighborhoods [10], [13].

Let's see an example in our example social network graph, shown in Fig. 2. With our link modeling, the receptive interactions of user u and v are depicted in yellow in Fig. 5, while the generative interactions are shown in green. More specifically, for user u, there are three outgoing neighbors. The generative-based status of u is, therefore, 1 by averaging over its outgoing links, and the receptive-based status of u is -1. Similarly, the receptive-based status of v is -1 while its engagement is 1.

***R**eceptive-Based Status Aggregation* (R). Intuitively, the incoming connections and associated semantics information of these links provide direct evidence of a user's popularity in online social networks. We build upon this basis to aggregate and propagate the receptive-based status between connected users.

In particular, to model the semantic information of the links, we first use one-hot encoding to represent positive and negative links, respectively. More specifically, we model positive and negative links as the following one-hot representations: $[1,0]^T$, and $[0,1]^T$. Then *S-GNN* employs a linear transformation to convert the one-hot encodings into the dense vectors through Eq. (1) and Eq. (4). For a link with sign $\text{sign}_{u\leftarrow v}$ (v is the creator of the link), we model the receptive-based status of u created by v as a combination of v's feature vector $x[v]$ and the corresponding latent factors of link $e_{\text{sign}_{u\leftarrow v}}$.

$$e_{\text{sign}_{u\leftarrow v}} = W_{u\leftarrow v} \cdot \text{sign}_{u\leftarrow v}, \tag{1}$$

$$\mathrm{R}_{u\leftarrow v} = x[v] \otimes e_{\text{sign}_{u\leftarrow v}}, \tag{2}$$

where $W_{u\leftarrow v} \in \mathcal{R}^{D_e\times 2}$ is a trainable transformation matrix, $\otimes$ denotes the concatenation operation between two vectors.

We now take the element-wise mean of the vectors in $\{\mathrm{R}_{u\leftarrow v}, \forall v \in N_I(u)\}$. This mean-based aggregator is a linear approximation of a localized spectral convolution [13], as the following function:

$$S_r[u] = \frac{1}{N_I(u)} \cdot \sum_{v\in N_I(u)} \mathrm{R}_{u\leftarrow v}. \tag{3}$$

***G**enerative-Based Status Aggregation* (G). Accordingly, we characterize a user's engagement through its outgoing social connections and associated semantics of generative links. We build upon this basis to perform the propagation and aggregation of the generative-based status between the connected users. Thus, the generative-based status of user u can be captured by the following functions:

$$e_{\text{sign}_{u\rightarrow v}} = W_{u\rightarrow v} \cdot \text{sign}_{u\rightarrow v}, \tag{4}$$

$$\mathrm{G}_{u\rightarrow v} = x[v] \otimes e_{\text{sign}_{u\rightarrow v}}, \tag{5}$$

$$S_g[u] = \frac{1}{N_O(u)} \cdot \sum_{v\in N_O(u)} \mathrm{G}_{u\rightarrow v}, \tag{6}$$

where $\mathrm{G}_{u\rightarrow v}$ denotes the engagement of user u to user v in a signed graph.

Learning Status Latent Factors of Nodes. In order to learn better latent factors of nodes for downstream signed network analysis, the receptive-based status and generative-based status are needed to be considered jointly. Here, we propose to combine these two types of status through a standard fully connected (FC) layer, where $S_r[u]$ and $S_g[u]$ are concatenated before feeding into the FC. Formally, the status latent factor of node u, $S[u]$, can be characterized as follows:

$$S[u] = W \cdot (S_r[u] \otimes S_g[u]) + b, \tag{7}$$

where W is a trainable transformation matrix, b is a learnable bias, and $\otimes$ represents the concatenation operator. The advantage of using concatenation lies in its simplicity and expressiveness, which have been shown in a recent work of graph convolutional neural networks [10].

Higher-Order Status Propagation. By stacking l status convolutional layers, a user is capable of receiving the status (the generative-based status and receptive-based status) propagated from its l-hop neighbors. In the l-th step, the

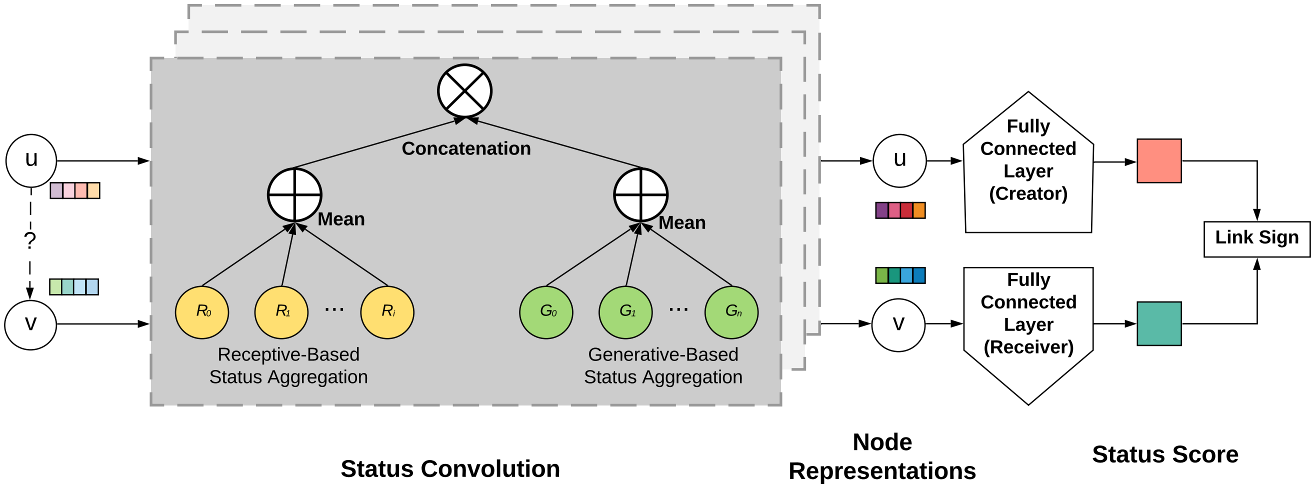

u
?
v
Concatenation
Mean
Mean
R_0
R_1
R_i
G_0
G_1
G_n
Receptive-Based Status Aggregation
Generative-Based Status Aggregation
u
v
Fully Connected Layer (Creator)
Fully Connected Layer (Receiver)
Link Sign
Status Convolution
Node Representations
Status Score

Fig. 6: Illustration of *S-GNN* framework.

representation of user u is recursively formulated as Eq. (8) - Eq. (12):

$$\mathrm{R}^l_{u\leftarrow v} = S^{l-1}[v] \otimes \{W^l_{u\leftarrow v} \cdot \mathrm{sign}_{u\leftarrow v}\}, \tag{8}$$

$$\mathrm{G}^l_{u\rightarrow v} = S^{l-1}[v] \otimes \{W^l_{u\rightarrow v} \cdot \mathrm{sign}_{u\rightarrow v}\}, \tag{9}$$

$$S^l_r[u] = \frac{1}{N_I(u)} \cdot \sum_{v\in N_I(u)} \mathrm{R}^l_{u\leftarrow v}, \tag{10}$$

$$S^l_g[u] = \frac{1}{N_O(u)} \cdot \sum_{v\in N_O(u)} \mathrm{G}^l_{u\rightarrow v}, \tag{11}$$

$$S^l[u] = W^l \cdot (S^l_r[u] \otimes S^l_g[u]) + b^l, \tag{12}$$

where $S^0[u] = x[u]$ is the feature vector of node u, $\mathrm{sign}_{u\rightarrow v}$ and $\mathrm{sign}_{u\leftarrow v}$ are the observed signed link, and $W^l_{u\leftarrow v}$, $W^l_{u\rightarrow v}$, W^l, and b^l are the model trainable parameters, to be optimized in an end-to-end fashion with *S-GNN*. Note that, by stacking multiple status convolutional layers, we enrich user embedding with its receptive-based status and generative-based status in signed social networks and allow controlling the range of status propagation throughout the graph by adjusting l.

5.3 Modeling Relationships of Users Based on Status Theory

In the theory of status, a signed link, from creator u to receiver v, can be interpreted as the intention of u in creating the link to v [19]. Status theory characterizes the semantics of the links from the relative difference of their status scores.

To further model the status score for each user, we fit the latent factor of the user status to fully connected (FC) layers. In particular, we use two different fully-connected layers to learn the status scores for the creator and receiver, respectively. Formally, the status scores are formulated as Eq. (13) and Eq. (14), where W_{src} and W_{rev} are trainable weight matrices defined in two FC layers, and b_{src} and b_{rev} are corresponding biases. The FC layers lead to more effective representations of user status, as this step explicitly injects the receptive-based status and generative-based status of individual users in a collaborative fashion.

$$S_{\mathrm{src}}[u] \leftarrow W_{\mathrm{src}} \times S^L[u] + b_{\mathrm{src}}, \tag{13}$$

$$S_{\mathrm{rev}}[v] \leftarrow W_{\mathrm{rev}} \times S^L[v] + b_{\mathrm{rev}}. \tag{14}$$

With status theory, the link semantics can be captured by its relative differential of the creator and receiver. Formally, it is formulated as:

$$S_{\Delta_{u\rightarrow v}} = S_{\mathrm{src}}[u] - S_{\mathrm{rev}}[v]. \tag{15}$$

Therefore, the sign of link $e_{u\rightarrow v}$ can be computed as:

$$\tilde{\mathrm{sign}}_{e_{u\rightarrow v}} = \begin{cases} -1 & \text{if } \sigma(S_{\Delta_{u\rightarrow v}}) > 0.5 \\ +1 & \text{otherwise} \end{cases},$$

where σ is a sigmoid function to normalize the status differential to the range of $(0,1)$, defined as $\mathrm{sigmoid}(x) = \frac{1}{1+\exp(-x)}$. The detailed forward propagation algorithm of *S-GNN* is shown as **procedure S-GNN**. Specifically, as illustrated in line 2 of **procedure S-GNN**, the node features are treated as the initial embedding that will be updated during the forward propagation. Lines $4-11$ of the procedure show how nodes are updated through L status convolutional layers. Lines $12-13$ describe the process of generating status scores through fully connected layers and obtaining the link sign based on the status differential of two nodes.

5.4 Model Training

We define an objective function to learn the model parameters in *S-GNN*. In particular, the objective function contains two terms, both of which are to encourage the representations to be able to understand associated status so as to obtain the semantics of links based on status theory. More

```
procedure S-GNN: REPRESENTATION GENERATION (I.E. FORWARD PROPAGATION)
    $S^0[u] \leftarrow x[u]$, for all $u \in \mathcal{V}$, where $X$ is node feature matrix
                                          ▷ Status latent factors of observed nodes
    for all $u \in \mathcal{V}$ do
        for $l = 1 \cdots L$ do
                                          ▷ The latent factors of receptive status
            for $i \in N_I(u)$ do
                $\mathrm{R}^l_{u\leftarrow i} = S^{l-1}[i] \otimes \{W^l_{u\leftarrow i} \cdot \mathrm{sign}_{u\leftarrow i}\}$
            $S^l_r[u] = \frac{1}{N_I(u)} \cdot \sum_{i \in N_I(u)} \mathrm{R}^l_{u\leftarrow i}$
                                          ▷ The latent factors of generative status
            for $i \in N_O(u)$ do
                $\mathrm{G}^l_{u\rightarrow i} = S^{l-1}[i] \otimes \{W^l_{u\rightarrow i} \cdot \mathrm{sign}_{u\rightarrow i}\}$
            $S^l_g[u] = \frac{1}{N_O(u)} \cdot \sum_{i \in N_O(u)} \mathrm{G}^l_{u\rightarrow i}$
            $S^l[u] = W^l \cdot [S^l_r[u] \otimes S^l_g[u]] + b^l$
                                          ▷ Status score
    for all $\langle u, v\rangle \in \mathcal{W}$ do
        $S_{\mathrm{src}}[u] \leftarrow W_{\mathrm{src}} \times S^L[u] + b_{\mathrm{src}}$
        $S_{\mathrm{rev}}[v] \leftarrow W_{\mathrm{rev}} \times S^L[v] + b_{\mathrm{rev}}$
        $S_{\Delta_{u\rightarrow v}} = \sigma(S_{\mathrm{src}}[u] - S_{\mathrm{rev}}[v])$
```

specifically, the first term minimizes the status differential of positive links, while the second term maximizes the status differential of negative links. Our framework can be regarded as a binary classifier for the link prediction task. Therefore, the first two terms together are the cross-entropy for training the model parameters. The overall objective function is formalized as follows:

$$\begin{aligned}\mathcal{L} = \frac{1}{|\mathcal{E}^+|} \sum_{e_{u\rightarrow v} \in \mathcal{E}^+} (\mathrm{sign}_{e_{u\rightarrow v}}) \cdot \sigma(S_{\Delta_{u\rightarrow v}}) \\ + \frac{1}{|\mathcal{E}^-|} \sum_{e_{u\rightarrow v} \in \mathcal{E}^-} (\mathrm{sign}_{e_{u\rightarrow v}}) \cdot \sigma(S_{\Delta_{u\rightarrow v}}) \\ + \lambda \cdot ||\Theta||_2^2,\end{aligned} \tag{16}$$

where $\mathrm{sign}_{e_{u\rightarrow v}}$ represents the ground-truth (Eq. 17), and $\Theta = \{\{W^l_{u\leftarrow v}, W^l_{u\rightarrow v}, W^l\}_{l=1}^L, W_{\mathrm{src}}, W_{\mathrm{rev}}\}$ denotes all trainable model parameters, and λ controls the L_2 regularization strength to prevent over-fitting. In particular, we adopt Adam [29] as the optimizer in our implementation, as it has been shown to be effective in updating the model parameters [10].

$$\mathrm{sign}_{e_{u\rightarrow v}} = \begin{cases} -1 & \text{if } e_{u\rightarrow v} \in \mathcal{E}^- \\ +1 & \text{if } e_{u\rightarrow v} \in \mathcal{E}^+ \end{cases}. \tag{17}$$

Time Complexity Analysis. As we can see, the key computational operations of our framework are the notion of localized graph convolutions [10], [30], [31]. More specifically, each status convolutional layer performs node-wise aggregation of information from neighbor nodes (i.e., every computation requires only the immediate neighbors of a node). In other words, the computation cost of our framework main comes from the localized graph convolutions, of which the complexity is coming from the model parameter complexity. And the parameters of our proposed status convolutional layers are shared across all nodes, making the parameter complexity of *S-GNN* independent of the input graph size. Specifically, the computation cost for each node is $\mathcal{O}(\mathcal{N}^L D^2)$, where $\mathcal{N}$ denotes the number of neighbors, L is the number of status convolutional layers, and D is the node hidden features.

As our framework inherits the inductive property of graph convolutional networks [10], at the representation generation step, we are able to compute embeddings for nodes that were not in the training set. This allows us to train on a subgraph to obtain model parameters and then generate embeddings for nodes that have not been observed during training. Sec. 6 empirically verified the efficiency and scalability of our model.

TABLE 2: Statistical Description of the Datasets.

DATASET	# OF NODES	# OF EDGES	+ EDGES (%)	- EDGES (%)
EPINIONS	131,828	841,372	85.30	14.70
SLASHDOT	82,140	549,202	77.40	22.60
WIKIRFA	11,258	179,418	77.92	22.08
WIKIELEC	7,126	104,167	78.78	21.22

6 EXPERIMENTAL EVALUATION

6.1 Description of Datasets Used

To evaluate the effectiveness and efficiency of *S-GNN*, we conduct experiments on four benchmark datasets: Epinions, Slashdot [2], WikiRfA [28], and WikiElec [32], which are publicly accessible signed social network datasets. The details of how the signed edges are defined on these datasets can refer to the website[2]. Table 2 presents the statistics of these datasets.

Epinions: This is a who-trust-whom online social network, where users create signed directed relations to each other indicating trust (corresponding to positive links) or distrust relationships (represented as negative links).

Slashdot: Slashdot is a technology-related news website known for its specific user community. Users in the network designate others as “friends” (positive links) or “foes” (negative links).

WikiRfA: This network is defined by votes for Wikipedia administrator candidates. Any member can cast a supporting, neutral or opposing vote for a Wikipedia editor. We discard neutral votes and construct a signed directed network as did in BESIDE [5] and [28]. **WikiElec:** is the elections and voting data of Wikipedia administrator, of which the definition is similar to that of WikiRfA.

6.2 Link Sign Prediction Based on Learned Node Representations

Alternative baselines. To demonstrate the effectiveness of learned embedding, we first conduct experiments on the task of link sign prediction. In particular, we compared *S-GNN* against the state-of-the-art methods on signed network embedding, and below we will detail them. We do not include unsigned methods (e.g., LINE [22], Node2Vec [21])

[2] https://snap.stanford.edu/data/index.html

TABLE 3: Sign Prediction Results

Dataset	Metric	SNE	SiNE	SIDE	SigNet	SGCN-UD	SGCN-D	SiGAT	SNEA	BESIDE-tri	BESIDE	S-GNN
Epinions	AUC	0.8282	0.8589	0.8768	0.9071	0.8997	0.8926	0.9359	0.8277	0.9304	**0.9437**	0.9397
	F1-binary	0.9206	0.9082	0.9487	0.9484	0.9486	0.9528	0.9586	0.9239	0.9601	0.9637	**0.9640**
	F1-macro	0.7634	0.6968	0.7817	0.8031	0.8212	0.8079	0.8441	0.7856	0.8478	0.8661	**0.8722**
	F1-micro	0.8529	0.8332	0.9094	0.9104	0.9119	0.9171	0.9282	0.8748	0.9306	0.9368	**0.9381**
Slashdot	AUC	0.6447	0.7736	0.7180	0.8759	0.8444	0.8409	0.8629	0.8030	0.8769	**0.9108**	0.8975
	F1-binary	0.8726	0.8671	0.8675	0.8994	0.8918	0.8942	0.8978	0.8802	0.9039	**0.9148**	0.9118
	F1-macro	0.4663	0.6341	0.5360	0.7579	0.7023	0.6904	0.7374	0.7723	0.7592	**0.7990**	0.7957
	F1-micro	0.7740	0.7654	0.7728	0.8406	0.8230	0.8245	0.8354	0.8234	0.8460	**0.8657**	0.8616
WikiRfA	AUC	0.6954	0.8610	0.6824	0.7903	0.8512	0.8393	0.6180	0.7765	0.8931	0.8969	**0.8979**
	F1-binary	0.8823	0.8730	0.8748	0.9012	0.9024	0.9001	0.8755	0.8615	0.9081	0.9101	**0.9136**
	F1-macro	0.6503	0.7365	0.4713	0.7454	0.7411	0.7285	0.4446	0.7405	0.7679	0.7740	**0.7848**
	F1-micro	0.8095	0.7746	0.7793	0.8407	0.8408	0.8369	0.7789	0.7969	0.8526	0.8560	**0.8619**
WikiElec	AUC	0.8169	0.7214	0.6682	0.7274	0.8523	0.8534	0.6386	0.7778	0.8981	0.9003	**0.9008**
	F1-binary	0.8960	0.8787	0.8836	0.8777	0.9058	0.9073	0.8818	0.8686	0.9142	0.9145	**0.9190**
	F1-macro	0.6803	0.6393	0.4833	0.5985	0.7446	0.7433	0.4574	0.7413	0.7723	0.7735	**0.7899**
	F1-micro	0.8259	0.7836	0.7924	0.7596	0.8463	0.8480	0.7894	0.8039	0.8608	0.8612	**0.8692**

and spectral clustering algorithms based on signed Laplacian matrix (e.g., SSE [33]), since previous signed network embedding work (BESIDE [6] and SGCN [14]) has shown their superiority over these methods.

SNE [9] adopted a log-bilinear model and used random walk sampling to generate samples. SNE was designed without any specific theories of signed networks.

SiNE [8] proposed a multi-layer neural network to learn the embeddings by optimizing an objective function satisfying structural balance theory. SiNE only concentrated on the immediate neighborhoods rather than the global balance structure.

SIDE [7] provided a linearly scalable approach with regard to the number of nodes. SIDE aggregated the direction and signed information of the links along the paths based on structural balance theory. It was proposed to optimize the likelihood over both directed and undirected connections.

SIGNet [6] was built upon the traditional word2vec family of embedding approaches. It leveraged a targeted node sampling strategy to maintain structural balance in higher-order neighborhoods.

SGCN [14] was a graph convolutional network (GCN) specialized for signed network analysis. Balance theory was leveraged to aggregate and propagate the information of signed networks across signed GCN layers.

SNEA [15] was built upon graph attention networks to capture balance theory. In particular, it leveraged masked self-attention layers to aggregate the rich information from neighboring nodes.

SiGAT [26] incorporated both balance theory and status theory to model signed directed networks based on motifs. Specifically, they defined 38 motifs, including directed edges, signed edges, and triangles. The graph convolution layer of SiGAT consists of 38 GAT aggregators, each corresponding to a neighborhood under a motif definition.

BESIDE [5] is the most relevant work to us in the literature, as both BESIDE [5] and our proposed approach can be used for link sign prediction and node ranking tasks simultaneously. Its core is to incorporate both balance theory and status theory for signed network embedding. By incorporating both triangles and "bridge" edges, BESIDE can learn embeddings for nodes and edges on signed directed networks. In particular, **BESIDE-tri** is a component of BESIDE, which only uses triangles with balance theory to learn node embeddings.

Among the baselines, SiNE can only deal with undirected signed networks, and SGCN only evaluated their model on undirected networks. In contrast, the others were designed in the context of directed signed networks. To be comparable, we follow the sampling method of [7] to generate associated undirected networks of the benchmark datasets. Then we evaluated SiNE over the undirected signed networks. As SGCN works for both undirected and directed networks, we run SGCN over undirected (SGCN-UD) and directed (SGCN-D) networks, respectively.

Evaluation metrics. Four standard metrics are used to measure link-sign prediction accuracy, including AUC, binary-F1, macro-F1, and micro-F1. Note that larger values of these metrics indicate better prediction accuracy. As commonly done in the literature [5], all experiments were run 5 times to obtain the average values. We used the average time over 5 runs for efficiency and scalability.

All the experiments are performed on a computer with Intel Core i7-9700K 8-core 3.6GHz CPU, GeForce GTX 1660 Ti GPU, 32GB RAM, and 500GB SSD.

Parameter settings. We implemented our proposed framework in PyTorch[3] [4]. We split each dataset into two parts: 80% edges for training and 20% for test. For each run, we run with different train-test splits. Because there are no node attributes in the datasets, we randomly initialize the node embeddings with 64 dimensions. In terms of hyperparameters, we applied a grid search for hyperparameters: the learning rate was tuned amongst $\{0.001, 0.005, 0.01, 0.05\}$, the coefficient of L_2 normalization was searched in $\{10^{-5}, 10^{-4}\}$. The model parameters are initialized using Xavier initializer [34]. In addition, the maximum epoch is set as 600, and early stopping strategy was performed, *i.e.*, premature stopping if training loss does not increase for 10 successive epochs. Without specification, we report the results of three status convolutional layers $[32, 64, 32]$, learning rate of 0.01 and normalization coefficient of 10^{-5}.

To evaluate the performance of learned node representations on the task of link sign prediction, we follow the method in BESIDE [5] to get link latent features through concatenation, *i.e.*, the feature of $e_{u\rightarrow v}$ is $[S_{\text{src}}[u] : S_{\text{rev}}[v]]$. Then these link latent features are used to train a logistic regression for evaluating the performance of the associate method. We used the released source code for SNE[5], SiNE[6], SIDE[7], SIGNet[8], SGCN[9], SNEA[10] SiGAT[11] and BESIDE[12], of which parameters were initialized as in the corresponding papers.

Prediction accuracy. Table 3 reports the performance comparison results, where the bold scores underlined are the best and the ones underlined are the second best. We have the following observations:

SNE achieves poor performance over four datasets, demonstrating the importance of social-physiological theories on signed network analysis. The performance of SiNE is reported on undirected networks. SGCN-UD consistently outperforms SiNE across all datasets, indicating that graph convolutional neural networks can effectively capture the complex relationships between entities on signed networks.

Compared to SIDE and SIGNet, the experimental results of SIGNet verify that maintaining structural balance in higher-order neighborhoods can improve the expressiveness of node representations. SGCN-D generally performs better than SIGNet in Epinions and WikiElec, while slightly worse in Slashdot and WikiRfA. Overall, the prediction performance of SGCN and SIGNet are comparable. It makes sense since SGCN introduces structural balance in higher-order connectivity by stacking multiple graph convolutional layers. THEREFORE, both SGCN and SIGNet demonstrate the importance of maintaining structural balance theory in higher-order neighborhoods.

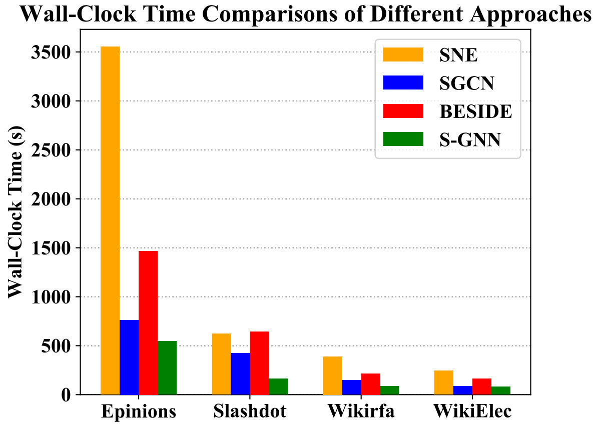


Fig. 7: Time Comparisons.

Both SiGAT and SNEA were built upon graph attention networks, and we observed that these two approaches could not achieve optimum performance. We conjecture that they could not encode comprehensive information of signed graphs as they do not explicitly consider the direction information of links. BESIDE achieves the best performance among the alternatives in all cases. Such improvements might be attributed to the cooperation of structural balance theory and status theory, which incorporates both triangle and "bridge" edges in a complementary manner. This indicates the benefits of applying status theory in directed signed networks.

S-GNN generally yields the best performance as compared with all of the baselines. Note that a slight improvement in the reported evaluation metrics implies a significant increase in the performance of link sign prediction. In particular, compared to BESIDE-tri (the one that only considers structural balance theory), *S-GNN* consistently performs better on all datasets, indicating that status theory can better capture the complex relationships in a directed signed network. Compared to BESIDE, which incorporates structural balance theory and status theory, our framework still provides better performance gains for some datasets, including fewer triangles satisfying balance theory (e.g., WikiRfA and WikiElec).

These observations are aligned with the statistics of balanced triangles in each dataset (see Sec. 5). As for the dataset more aligned with balance theory (e.g., Slashdot), our proposed model performs on par with BESIDE. The experimental results indicate that our proposed model may implicitly capture the property of structural balance theory and status theory even though our loss term for training *S-GNN* is designed based on status theory. In a nutshell, by modeling the heterogeneous relationships with status and stacking multiple status convolutional layers, *S-GNN* can obtain more expressive representations on signed networks in most cases.

Efficiency. For efficiency evaluation, we compared the total runtime of *S-GNN* with the baselines. Because the released codes of SiNE, SIGNet, and SIDE are not available on GPU, comparing them with those that can be run on GPU is not expected. Therefore, we report the comparison results

[3] https://pytorch.org
[4] Our implementation: https://www.dropbox.com/sh/2vmgkr524b497nt/AAAjBBA0pp6NZ6WG3Oc0GJS0a?dl=0
[5] https://bitbucket.org/bookcold/sne-signed-network-embedding/src/master/
[6] http://www.public.asu.edu/~swang187/codes/SiNE.zip
[7] https://datalab.snu.ac.kr/side/
[8] https://github.com/raihan2108/signet
[9] https://github.com/benedekrozemberczki/SGCN
[10] https://github.com/liyu1990/snea
[11] https://github.com/huangjunjie95/SiGAT
[12] https://github.com/yqc01/BESIDE

with the baselines that can run on GPU. The results are shown in Fig. 7. It is worth noting that *S-GNN* consistently outperforms in all cases. In particular, *S-GNN* speeds up the processing time by up to 6.5× as compared with the baselines. Compared with the state-of-the-art literature, BESIDE, *S-GNN* is 4× and 2.7× faster on Slashdot and Epinions, respectively.

Comparing S-GNN with SGCN. We observed that SGCN performs the second-best in terms of run time. This can be explained by the fact that SGCN also inherits the benefits of graph neural networks like our framework: it avoids the parameter explosion in graphs, and it allows for parameter sharing across the graph to avoid overfitting. The incurred overhead of SGCN may come from its complicated way of aggregating and propagating information across the network. Overall, Fig. 7 demonstrates that our proposed status convolutional layer greatly speeds up the sign prediction process and shows its promising that it can be applied to large-scale network applications.

Robustness. We evaluated the approaches with different training and test set ratios to measure their robustness. The portions of the training set were set as 80%, 60%, 40% of the entire dataset. In Table 4, we report the evaluation results on Epinions and WikiRfA, whereas we omit the results on Slashdot and WikiElec as they showed a similar performance trend. Both of *S-GNN* and BESIDE have good robustness. In particular, *S-GNN* has a minor F1-binary decrease of 0.0026 in Epinions and 0.0054 in WikiRfA when the size of the training set is reduced to 40% of the entire graph, while BESIDE has a decrease of 0.0041 and 0.0106, respectively. This indicates that our proposed framework has better robustness to the size of the training set.

Scalability. The scalability of *S-GNN* is evaluated by measuring the wall-clock times with a different number of nodes and a different number of node pairs (links/edges), respectively. Both of the selected nodes and pairs are subgraphs from the main graph of the dataset, and each node in the subgraphs has at least one edge (no singleton node). In Fig. 8, we only report the results on Epinions and WikiRfA, and the results on Slashdot and WikiElec are omitted as they showed a similar performance trend.

Concretely, as Fig. 8a and Fig. 8c show, the wall-clock time of BESIDE increases sharply with the number of nodes while *S-GNN* consistently performs well as the number of nodes increases. This is because the parameters of our proposed status convolutional layers are shared across all nodes, making the parameter complexity of our approach independent of the number of nodes. For the increasing number of node pairs, the time of BESIDE increases dramatically and shows similar trends on both datasets, shown in Fig. 8b and Fig. 8d. We observe that *S-GNN* tends to increase slightly with the input graph size increase. That can be explained by the fact that neighborhood size increases when the number of users increases, leading to more local aggregation computations. In a nutshell, *S-GNN* consistently performs well on all benchmarking datasets, indicating that *S-GNN* is more scalable and can readily be generalized to large-scale network applications.

TABLE 4: Robustness

Dataset	Training Size	metric	SIDE	SGCN	BESIDE	S-GNN
Epinions	0.8	AUC	0.8768	0.8926	**0.9437**	0.9396
		F1-binary	0.9487	0.9528	0.9637	**0.9642**
		F1-macro	0.7817	0.8079	0.8661	**0.8726**
		F1-micro	0.9094	0.9171	0.9368	**0.9385**
	0.6	AUC	0.8788	0.8278	**0.9393**	0.9366
		F1-binary	0.9495	0.9479	0.9615	**0.9616**
		F1-macro	0.7830	0.7598	0.8597	**0.8632**
		F1-micro	0.9108	0.9071	0.9335	**0.9340**
	0.4	AUC	0.8700	0.8215	**0.9340**	0.9295
		F1-binary	0.9481	0.9470	0.9596	**0.9616**
		F1-macro	0.7719	0.7553	0.8519	**0.8634**
		F1-micro	0.9080	0.9054	0.9303	**0.9340**
WikiRfA	0.8	AUC	0.6824	0.8393	0.8969	**0.8979**
		F1-binary	0.8748	0.9001	0.9101	**0.9136**
		F1-macro	0.4713	0.8295	0.7740	**0.7848**
		F1-micro	0.7793	0.8369	0.8560	**0.8619**
	0.6	AUC	0.6801	0.8281	0.8901	**0.8948**
		F1-binary	0.8761	0.8970	0.9068	**0.9122**
		F1-macro	0.4716	0.7220	0.7633	**0.7814**
		F1-micro	0.7813	0.8321	0.8503	**0.8597**
	0.4	AUC	0.6710	0.8069	0.8729	**0.8858**
		F1-binary	0.8750	0.8941	0.8995	**0.9082**
		F1-macro	0.4626	0.7090	0.7394	**0.7717**
		F1-micro	0.7791	0.8268	0.8378	**0.8533**

TABLE 5: Link Sign Prediction Accuracy Based on Status (%)

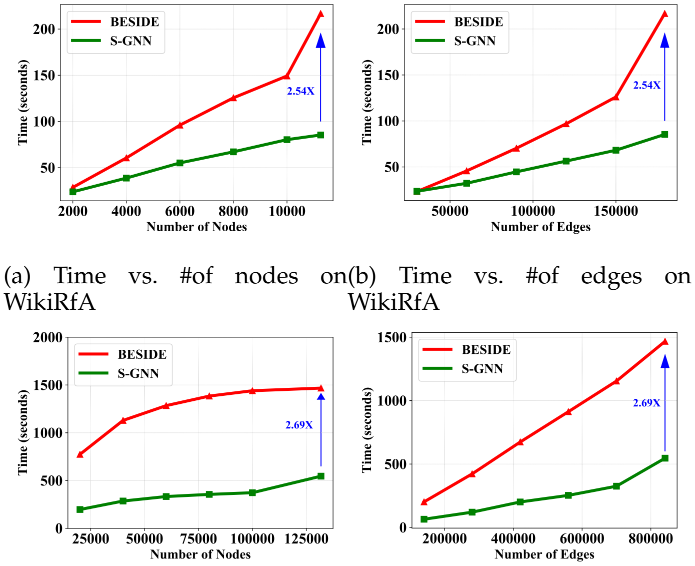
dataset \ method	PageRank	BESIDE-sta	BESIDE	S-GNN
Epinions	65.15	85.28	91.52	**92.40**
Slashdot	62.73	82.77	**86.01**	85.58
WikiRfA	66.37	80.67	82.39	**85.28**
WikiElec	72.83	79.65	81.72	**85.71**

6.3 Link Sign Prediction Based on Status

Alternative baselines. According to the theory of status, a positive edge indicates that the receiver has a higher status than the creator, which can be interpreted as "I trust people who have higher status than me" and vice versa. For example, if sign$_{e_{u\to v}}$ is $+1$, then $S_{\text{rev}}[v]-S_{\text{src}}[u]$ should be positive. To illustrate the effectiveness of the learned status score of each entity, we select three baselines that can obtain the status scores of nodes. We do not include other methods (e.g.,

TABLE 6: Global Ranking Results on WikiRfA&WikiElec: never elected (red), elected once (black), and elected twice (blue)

Method / Rank	WikiRfA			WikiElec		
	PageRank	BESIDE	S-GNN	PageRank	BESIDE	S-GNN
1	West.andrew.g	Can't sleep...	Nev1	alkivar	sarah_ewart	bd2412
2	Cobi	SarahStierch	Legoktm	cambridgebayweather	khoikhoi	derhexer
3	ProtectionBot	Phaedriel	NCurse	graft	amidaniel	wjbscribe
4	Anomie	DerHexer	Dabomb87	guanaco2	elonka	arjun01
5	Jason Quinn	Alex Bakharev	PeterSymonds	schissel	alex_bakharev	duja
6	RedirectCleanupBot	Werdna	DerHexer	andyz	Can't sleep...	ncurse
7	lustiger seth	HJ Mitchell	Can't sleep...	savidan	ncurse	Can't sleep...
8	Dinoguy1000	Everyking	Phaedriel	sebastiankessel	halibutt	amidaniel
9	Bellhalla	Dabomb87	HJ Mitchell	peruvianllama	phaedriel	newyorkbrad
10	TommyBoy	PeterSymonds	SarahStierch	slambo	derhexer	phaedriel



(a) Time vs. #of nodes on WikiRfA

(b) Time vs. #of edges on WikiRfA

(c) Time vs. #of nodes on Epinions

(d) Time vs. #of edges on Epinions

Fig. 8: Scalability: S-GNN vs. BESIDE.

such as Prestige [35], MPR [36], and Troll-Trust [37]), because BESIDE has shown its superiority over these methods.

PageRank [38] was a classical ranking algorithm for unsigned networks. For this implementation, we follow the same method in BESIDE to obtain the status score of each node, applying it to the positive subgraph (graph contains positive links only) to obtain the global values (status scores) for nodes.

BESIDE [5] uses both triangles and "bridge" edges to train the model and obtain associated node status. **BESIDE-sta** is a component of BESIDE, which only uses "bridge" edges with status theory to learn the status of each node.

We use 80% edges for training and 20% edges for test. The result is obtained by comparing the status differential of two nodes and associated ground-truth (link sign) in the test set. Experiments are performed on four datasets, including Epinions, Slashdot, WikiRfA, and WikiElec. We use accuracy as the evaluation metric, as did in BESIDE. The results are shown in Table 5, in which we have the following observations:

In general, *S-GNN* outperforms all baselines over Epinions, WikiRfA, and WikiElec, while BESIDE performs slightly better on Slashdot. More precisely, *S-GNN* has significantly better results, a 2.89% improvement on WikiRfA and a 3.99% improvement on WikiElec, respectively, which verifies that these two datasets can be better characterized with status. Moreover, *S-GNN* improves the accuracy by up to 7.12% as compared with BESIDE-sta, indicating that our proposed status convolutional layers can capture the status property very well. PageRank obtains poor performance among all methods, implying the importance of both positive and negative links in signed network analysis.

6.4 Global Node Ranking Based on Status

To determine if the status scores are plausible for ranking on the global scale, we also compare the top 10 nodes ranked based on PageRank, BESIDE, and *S-GNN*, respectively. For PageRank, each node has a single ranking score which can be directly used to find the top 10 nodes. As for BESIDE and *S-GNN*, we use the combination equation as did in BESIDE [5] to obtain the ranking score for each node based on the learned status scores.

$$S_v = \sum_{u \in N_I(v)^+} \frac{S_u}{N_o(u)} - \sum_{u \in N_I(v)^-} \frac{S_u}{N_o(u)}. \tag{18}$$

S_v is the status score of node v, $N_I(v)^+$ and $N_I(v)^-$ are two sets of source nodes pointing to the node v with positive or negative edges respectively, S_u is the source status score of node u and $N_o(u)$ is the out-degree of node u.

The experiments are conducted on WikiRfA and WikiElec since they clearly indicate the global ranking. For WikiRfA, among 3949 candidates, 1885 and 18 users are elected once and twice, respectively. For WikiElec, there are 2391 candidates, among which 1223, 11, and 1 users are elected once, twice, and thrice respectively. More times of successful elections indicate higher status. The results are shown in Table 6 with never-elected (in red), once-elected (in black), and twice-elected (in blue), respectively.

We can observe that *S-GNN* and BESIDE both have three twice-elected users on WikiRfA. In particular, the average ranks of these twice-elected users are higher in *S-GNN* than BESIDE. On WikiElec, both PageRank and BESIDE include never-elected candidates, while the top 10 users selected by *S-GNN* are all once-elected. The result indicates *S-GNN* can capture better global ranking features.

- PageRank performs the worst overall: with one and two never-elected candidates for WikiRfA and WikiElec, respectively, indicating that excluding negative links would not effectively rank the candidates.
- Excluding the overlapped candidates selected by BESIDE and *S-GNN* on WikiRfA, the average number of negative links on the rest four candidates is significantly less in *S-GNN*, 8.75 selected by *S-GNN* compared to 123.25 by BESIDE. This result shows that *S-GNN* may give lower scores for candidates with a higher number of negative links.
- For the twice-elected candidates on WikiRfA, "Everyking" has the highest number of negative links—334—which might be the reason why this candidate is not included in the top-10 selected candidates by *S-GNN*.
- BESIDE has "halibutt" as one of its top-10 candidates. This candidate has 69 positive and 28 negative links, respectively. Allowing a high ratio of negative links may result in inaccurate ranking.

7 Concluding Remarks

In this paper, we devised a status-aware graph neural network, called *S-GNN*, to learn effective network embedding in a signed heterogeneous network. The key of *S-GNN* is the newly proposed status convolutional layer, which can jointly capture the rich graph structural and semantic information of the links. *S-GNN* is equipped with an objective designed based on *status theory*. Extensive experiments on four benchmark datasets have demonstrated the rationality and effectiveness of our proposed *S-GNN*. Meanwhile, it enjoys high efficiency, scalability, and robustness due to the notion of localized graph convolutions.

References

[1] P. Xu, W. Hu, J. Wu, W. Liu, B. Du, and J. Yang, "Social Trust Network Embedding," in *Proc. International Conference on Data Mining*. IEEE, 2019, pp. 678–687.

[2] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed Networks in Social Media," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 1361–1370.

[3] X. Song, Y. Chi, K. Hino, and B. Tseng, "Identifying Opinion Leaders in the Blogosphere," in *Proc. International Conference on Information and Knowledge Management*. ACM, 2007, pp. 971–974.

[4] J. Tang, Y. Chang, C. Aggarwal, and H. Liu, "A Survey of Signed Network Mining in Social Media," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 42, 2016.

[5] Y. Chen, T. Qian, H. Liu, and K. Sun, ""Bridge" Enhanced Signed Directed Network Embedding," in *Proc. International Conference on Information and Knowledge Management*. ACM, 2018, pp. 773–782.

[6] M. R. Islam, B. A. Prakash, and N. Ramakrishnan, "SIGNet: Scalable Embeddings for Signed Networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 157–169.

[7] J. Kim, H. Park, J.-E. Lee, and U. Kang, "SIDE: Representation Learning in Signed Directed Networks," in *Proc. International World Wide Web Conference*, 2018, pp. 509–518.

[8] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu, "Signed Network Embedding in Social Media," in *Proc. SIAM International Conference on Data Mining*. SIAM, 2017, pp. 327–335.

[9] S. Yuan, X. Wu, and Y. Xiang, "SNE: Signed Network Embedding," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 183–195.

[10] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Proc. International Conference on Neural Information Processing Systems (NeurIPS 2017)*, 2017, pp. 1024–1034.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in *Proc. International Conference on Learning Representations*, 2018.

[12] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" in *Proc. International Conference on Learning Representations*, 2018.

[13] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proc. International Conference on Machine Learning (ICML 2017)*, 2017.

[14] T. Derr, Y. Ma, and J. Tang, "Signed Graph Convolutional Networks," in *Proc. International Conference on Data Mining*. IEEE, 2018, pp. 929–934.

[15] Y. Li, Y. Tian, J. Zhang, and Y. Chang, "Learning Signed Network Embedding via Graph Attention," in *Proc. the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4772–4779.

[16] W. J. Goode, *The Celebration of Heroes: Prestige as a Control System*. University of California Press Berkeley, 1978.

[17] M. Sauder, F. Lynn, and J. M. Podolny, "Status: Insights from Organizational Sociology," *Annual Review of Sociology*, vol. 38, pp. 267–283, 2012.

[18] D. Cartwright and F. Harary, "Structural Balance: a Generalization of Heider's Theory," *Psychological Review*, vol. 63, no. 5, p. 277, 1956.

[19] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of Trust and Distrust," in *Proc. International World Wide Web Conference*. ACM, 2004, pp. 403–412.

[20] M. Zhang and Y. Chen, "Link Prediction Based on Graph Neural Networks," in *Proc. International Conference on Neural Information Processing Systems (NeurIPS 2018)*, vol. 31, 2018, pp. 5165–5175.

[21] A. Grover and J. Leskovec, "Node2Vec: Scalable Feature Learning for Networks," in *Proc. SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 855–864.

[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-Scale Information Network Embedding," in *Proc. International World Wide Web Conference*. ACM, 2015, pp. 1067–1077.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online Learning of Social Representations," in *Proc. SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 701–710.

[24] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "SHINE: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction," in *Proc. International Conference on Web Search and Data Mining*. ACM, 2018, pp. 592–600.

[25] F. S. Rizi and M. Granitzer, "Signed Heterogeneous Network Embedding in Social Media," in *Proc. ACM Symposium on Applied Computing*. ACM, 2020, pp. 1877–1880.

[26] J. Huang, H. Shen, L. Hou, and X. Cheng, "Signed Graph Attention Networks," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 566–577.

[27] P. Xu, W. Hu, J. Wu, and B. Du, "Link Prediction with Signed Latent Factors in Signed Social Networks," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1046–1054.

[28] R. West, H. S. Paskov, J. Leskovec, and C. Potts, "Exploiting Social Network Structure for Person-to-Person Sentiment Analysis," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 297–310, 2014.

[29] D. P. Kingma and J. Ba, "ADAM: A Method for Stochastic Optimization," *Proc. International Conference on Learning Representations*, 2015.

[30] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," in *Proc. SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2018, pp. 974–983.

[31] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[32] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting Positive and Negative Links in Online Social Networks," in *Proc. International World Wide Web Conference*. ACM, 2010, pp. 641–650.

[33] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral Analysis of Signed Graphs for Clustering, Prediction and Visualization," in *Proc. SIAM International Conference on Data Mining*. SIAM, 2010, pp. 559–570.

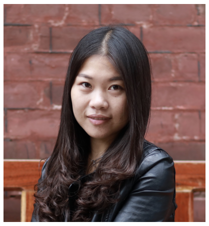
[34] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[35] K. Zolfaghar and A. Aghaie, "Mining Trust and Distrust Relationships in Social Web Applications," in *Proc. International Conference on Intelligent Computer Communication and Processing*. IEEE, 2010, pp. 73–80.

[36] M. Shahriari and M. Jalili, "Ranking Nodes in Signed Social Networks," *Social Network Analysis and Mining*, vol. 4, no. 1, p. 172, 2014.

[37] Z. Wu, C. C. Aggarwal, and J. Sun, "The Troll-Trust Model for Ranking in Signed Networks," in *Proc. International Conference on Web Search and Data Mining*. ACM, 2016, pp. 447–456.

[38] S. Brin and L. Page, "Reprint of: The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.



Wanyu Lin received her Ph.D. degree from the Department of Electrical and Computer Engineering at the University of Toronto. She received her B.Engr. degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology, China and her MPhil. degree from the Department of Computing, The Hong Kong Polytechnic University. Her research interests include data outsourcing security in cloud computing, machine learning in adversarial settings, and model interpretation. She is a member of IEEE.

Baochun Li received his B.Engr. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 1995 and his M.S. and Ph.D. degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, in 1997 and 2000. Since 2000, he has been with the Department of Electrical and Computer Engineering at the University of Toronto, where he is currently a Professor. He holds the Bell Canada Endowed Chair in Computer Engineering since August 2005. His research interests include cloud computing, distributed systems, datacenter networking, and wireless systems. He was the recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems in 2000. In 2009, he was a recipient of the Multimedia Communications Best Paper Award from the IEEE Communications Society, and a recipient of the University of Toronto McLean Award. He is a member of ACM and a Fellow of IEEE.