Learning Signed Network Embedding via Graph Attention

Yu Li,[1,5] **Yuan Tian,**[2,4*] **Jiawei Zhang,**[3] **Yi Chang**[2,5]
[1]College of Computer Science and Technology, Jilin University, China
[2]School of Artificial Intelligence, Jilin University, China
[3]IFM Lab, Department of Computer Science, Florida State University, USA
[4]Key Laboratory of Bionic Engineering, Ministry of Education, China
[5]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China
liyu18@mails.jlu.edu.cn, yuantian@jlu.edu.cn, jiawei@ifmlab.org, yichang@jlu.edu.cn

Abstract

Learning the low-dimensional representations of graphs (i.e., network embedding) plays a critical role in network analysis and facilitates many downstream tasks. Recently graph convolutional networks (GCNs) have revolutionized the field of network embedding, and led to state-of-the-art performance in network analysis tasks such as link prediction and node classification. Nevertheless, most of the existing GCN-based network embedding methods are proposed for unsigned networks. However, in the real world, some of the networks are signed, where the links are annotated with different polarities, e.g., positive vs. negative. Since negative links may have different properties from the positive ones and can also significantly affect the quality of network embedding. Thus in this paper, we propose a novel network embedding framework SNEA to learn Signed Network Embedding via graph Attention. In particular, we propose a masked self-attentional layer, which leverages self-attention mechanism to estimate the importance coefficient for pair of nodes connected by different type of links during the embedding aggregation process. Then SNEA utilizes the masked self-attentional layers to aggregate more important information from neighboring nodes to generate the node embeddings based on balance theory. Experimental results demonstrate the effectiveness of the proposed framework through signed link prediction task on several real-world signed network datasets.

Introduction

Network embedding, aiming to learn low-dimensional embedding of nodes in networks, plays a critical role in network analysis and has received much attention from data mining and machine learning communities. Since graphs have been a popular way to model structured data, network embedding enables many downstream network analysis tasks such as link prediction, node classification and community detection (Tian et al. 2014; Zhang et al. 2018; Zhang and Chen 2018; Shao et al. 2019) (Please note that the terms graph and network are used interchangeably in this paper). Traditional network embedding methods predominantly focus on the high-order proximity approximation and network properties preservation, and employ linear modeling approaches to obtain the node embeddings (Yang et al. 2017; Lian et al. 2018; Qiu et al. 2018). Recent research, however, has pivoted to learn the node embeddings using graph convolution networks (GCNs) (Kipf and Welling 2017; Chen, Ma, and Xiao 2018; Abu-El-Haija et al. 2018; Derr, Ma, and Tang 2018; Abu-El-Haija et al. 2019), which aim to aggregate information from the neighbors for node embeddings. These GCN-based network embedding methods have revolutionized the field of network embedding and achieved the state-of-the-art performance in network analysis tasks. Nevertheless, most of the GCN-based network embedding methods are only proposed for unsigned networks (consisting of only positive links). However, in the real word, some networks are signed with the links are annotated with different polarities, i.e., positive vs. negative. Such different link polarities can convey very different physical meanings and information (Kunegis, Preusse, and Schwagereit 2013), which should be effectively incorporated in network representation learning. Since the methods proposed for unsigned networks can not distinguish the different properties of positive and negative links, and fail to exploit additional information from negative links, therefore they cannot be directly applied on signed networks.

Some signed network embedding methods have been proposed in recent years (Kunegis et al. 2010; Hsieh, Chiang, and Dhillon 2012; Chiang, Whang, and Dhillon 2012; Zheng and Skillicorn 2015; Yuan, Wu, and Xiang 2017; Wang et al. 2017; Kim et al. 2018; Derr, Ma, and Tang 2018). Some of these methods employ linear modeling approaches to learn the node embedding by spectral analysis or matrix factorization, while the other methods treat neighboring nodes equally without considering the different contributions of different nodes when aggregating and propagating information in networks. Recently, for the target nodes, many works have demonstrated that impacts of different neighbors can be different, and quantifying such different impacts can significantly improve the performance of unsigned network analysis tasks with deep learning models (Zhao et al. 2017; Vaswani et al. 2017; Veličković et al. 2018). However, in signed networks, negative links have different properties from positive links, therefore signed net-

*Corresponding author: yuantian@jlu.edu.cn

works will generate more complex relationships than unsigned networks.

Based on the above analysis, when designing a neural network architecture with attention mechanism for signed network, we need to address the following issues. In a signed network, nodes can be connected by different types of links (i.e., positive links and negative links). In other words, nodes can have different sets of neighbors connected by links with different polarities. Therefore, how to distinguish the difference of neighbors and efficiently aggregate more important information from the neighboring nodes is required in signed networks.

In this paper, we propose a novel network embedding framework SNEA to learn Signed Network Embeddings via graph Attention. Instead of directly applying mean-pooling strategy when aggregating information from neighboring nodes, SNEA proposes a graph attentional layer, which utilizes a masked self-attention mechanism to compute the importance coefficients for the neighbors. The importance coefficients effectively quantify the impacts among the neighbors in node representation learning. Afterwards, based on the balance theory, SNEA stacks multiple graph attentional layers to aggregate node information from neighboring nodes with different importance coefficients. Therefore, the node embeddings learned by SNEA can extract not only the local structural information, but also the global embedding in the signed networks, whose effectiveness will be evaluated with experiments on real-world signed network datasets. Our main contributions are listed as follows:

- we propose a graph attentional layer, which utilizes a masked self-attention mechanism to estimate the importance coefficient for pair of nodes connected by different type of links for embedding aggregation process.
- we propose a novel signed network embedding framework, namely SNEA, which leverages the graph attentional layers to aggregate more important information from neighboring nodes based on balance theory;
- we design an objective function for both framework optimization and node representation learning;
- we evaluate the effectiveness of the proposed framework SNEA on several real-world signed network datasets through the signed link prediction task.

Related Work

In recent years, signed network analysis has attracted more and more attention from data mining and machine learning communities, as many systems can be expressed as signed graphs or signed networks. Since the network analysis methods proposed for unsigned networks (Wang, Cui, and Zhu 2016; Zhang et al. 2018; Zhang and Chen 2018; Lian et al. 2018) can not distinguish the different properties of positive and negative links, many kinds of signed network analysis methods have been developed for tasks such as node clustering, node classification, signed link prediction and so on.

In (Kunegis et al. 2010; Chiang, Whang, and Dhillon 2012; Zheng and Skillicorn 2015), signed Laplacian embedding methods adopt signed variants of the graph Laplacian to cluster nodes in signed networks. In (Hsieh, Chiang, and Dhillon 2012; Tang, Aggarwal, and Liu 2016), matrix factorization based signed network embedding methods are proposed for node classification and signed link prediction tasks. The probabilistic methods are also used for node representation learning. For example, SNE (Yuan, Wu, and Xiang 2017) adopts a log-bilinear model and optimizes a Skip-Gram-like (Mikolov et al. 2013) objective function by the maximum likelihood estimation during node embedding learning. SIDE (Kim et al. 2018) utilizes truncated random walks and optimizes likelihood over different type of network links to derive the node embeddings.

Not only these linear modeling methods, but the deep learning based signed network embedding methods are also proposed for signed networks. SiNE (Wang et al. 2017) optimizes an objective function guided by social theory in signed networks to generate the node embeddings using a deep learning framework. SGCN (Derr, Ma, and Tang 2018) generalizes the GCNs to signed networks and applies a mean-pooling strategy to aggregate information from neighboring nodes according to social theory.

With the increasing investigation of the attention mechanism on unsigned networks (Zhao et al. 2017; Abu-El-Haija et al. 2018; Veličković et al. 2018; Wang et al. 2019; Yang et al. 2019), it has been demonstrated that impacts of different neighboring nodes can be different, and quantifying such different impacts can significantly improve the performance of network analysis tasks. Thus researchers have begun to turn their attention to signed networks (Huang et al. 2019). SiGAT (Huang et al. 2019) first introduces the GAT (Veličković et al. 2018) to signed networks and designs a motif-based graph neural network model based on balance theory and status theory. Different from SiGAT, our method SNEA proposes a graph attentional layer, which provides a more universal way to aggregate and propagate more important information through both positive and negative links based on balance theory.

Preliminary

For the convenience of presentation, we first introduce the main notations used in this paper. Boldface uppercase letters (e.g., $\mathbf{A}$) denote matrices, boldface lowercase letters (e.g., $\mathbf{w}$) denote vectors. Calligraphic math font (e.g., $\mathcal{V}$) denotes set, and $|\mathcal{V}|$ is the cardinality of $\mathcal{V}$. In this way, a signed network can be expressed as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ is the set of *n* nodes and $\mathcal{E} = \{e_{ij}\}_{v_i, v_j \in \mathcal{V}}$ is the set of links. Note that $\mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$ and $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$, where $\mathcal{E}^+$ and $\mathcal{E}^-$ denote the sets of positive and negative links, respectively. Positive and negative neighbors of v_i are denoted as $\mathcal{N}_i^+$ and $\mathcal{N}_i^-$ respectively, in addition, $\hat{\mathcal{N}}_i^+ = \mathcal{N}_i^+ \cup \{v_i\}$, $\hat{\mathcal{N}}_i^- = \mathcal{N}_i^- \cup \{v_i\}$ and $\mathcal{N}_i = \mathcal{N}_i^+ \cup \hat{\mathcal{N}}_i^-$.

As the balance theory (Heider 1946; Cartwright and Harary 1956) implies that "the friend of my friend is my friend" and "the foe of my friend is my foe", for the target node, there exists a set of "friend" nodes and a set of "foe" nodes, which are called balanced node set and unbalanced node set.

Definition 1 ***Balanced/Unbalanced node set*** *(Derr, Ma, and Tang 2018). For the target node v_i in signed network, the balanced (unbalanced) node set is defined as a set of nodes (e.g., v_j) that have even (odd) negative links along a path connecting v_i and v_j.*

More specifically, the balanced node set $\mathcal{B}_i$ and unbalanced node set $\mathcal{U}_i$ with respect to v_i can be defined as:

When $l = 1$

$$\mathcal{B}_i(1) = \{v_j | v_j \in \mathcal{N}_i^+\}$$

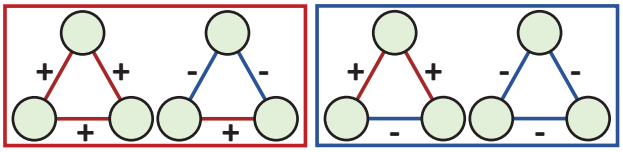
$$\mathcal{U}_i(1) = \{v_j | v_j \in \mathcal{N}_i^-\}$$

For $l > 1$

$$\mathcal{B}_i(l) = \{v_j | v_k \in \mathcal{B}_i(l-1) \; and \; v_j \in \mathcal{N}_k^+\} \cup \{v_j | v_k \in \mathcal{U}_i(l-1) \; and \; v_j \in \mathcal{N}_k^-\}$$

$$\mathcal{U}_i(l) = \{v_j | v_k \in \mathcal{U}_i(l-1) \; and \; v_j \in \mathcal{N}_k^+\} \cup \{v_j | v_k \in \mathcal{B}_i(l-1) \; and \; v_j \in \mathcal{N}_k^-\}$$

where l denotes the length of path between pair of nodes.



(a) balanced (b) unbalanced

Figure 1: Four types of triangles in a signed network.

However, for each node, the balanced node set and the unbalanced node set maybe overlap. As shown in Figure 1, the two triangles on the left are balanced triangles according to balance theory, while the two triangles on the right are unbalanced triangles. The pairs of nodes in the unbalanced triangles are both "friend" and "foe". As known, the information from the balanced node set ("friends") and the unbalanced node set ("foes") may convey different characteristics. There exists problems if we represent each node as only one representation. To tackle this issue, in this paper, we represent each node as two embeddings, i.e., balanced embedding and unbalanced embedding, respectively.

The Proposed Framework

In this section, we propose a novel network embedding framework - SNEA for signed network. The remaining part of this section will be organized as follows. At the beginning, a graph attentional layer, which utilizes a masked self-attention mechanism to handle the problem of relevance of node embeddings in signed network will be introduced. After that, we will learn the node embeddings by aggregating information from balanced and unbalanced node sets by stacking the graph attentional layers. Finally, we will introduce how to train this model and handle practical tasks in signed networks.

Signed Graph Attentional Layers

Before aggregating information from neighbors for each node, we should notice that negative links have different properties from positive links and the impacts from different type of neighbors are different. Here we introduce a masked self-attention mechanism, which is used to learn the importance of neighbors for each node in signed network when aggregating and propagating information through positive and negative links.

As we mentioned above, each node can be represented as balanced embedding $\mathbf{h}^{\mathcal{B}}$ and unbalanced embedding $\mathbf{h}^{\mathcal{U}}$, and these two types of embeddings have different characteristics. Therefore, for each embedding type of nodes (e.g., node with embedding type $\mathcal{R}$, $\mathcal{R} \in \{\mathcal{B}, \mathcal{U}\}$), we leverage the self-attention mechanism to learn the importance of neighbors for each node during the information aggregation process.

Given nodes $v_i \in \mathcal{V}$ and $v_j \in \mathcal{N}_i$, let $\mathbf{h}_i$ and $\mathbf{h}_j$ denote the input embeddings of node v_i and v_j. The importance of v_j to v_i during the information aggregation process for embedding type $\mathcal{R}$ can be formulated as follows:

$$\begin{aligned} e_{ij}^{\mathcal{R}} &= a\left(\mathbf{h}_i\mathbf{W}^{\mathcal{R}}, \mathbf{h}_j\mathbf{W}^{\mathcal{R}}, \mathcal{R}\right) \\ &= \tanh(\mathbf{b}^{\mathcal{R}}(\mathbf{h}_i\mathbf{W}^{\mathcal{R}} \| \mathbf{h}_j\mathbf{W}^{\mathcal{R}})^T) \end{aligned} \tag{1}$$

where the attention mechanism a denotes a single-layer feedforward neural network, parameterized by a shared attentional parameter vector $\mathbf{b}^{\mathcal{R}} \in \mathbb{R}^{1\times 2d_{out}}$, and applying a $\tanh$ function to make the attention model nonlinearity. $\mathbf{W}^{\mathcal{R}} \in \mathbb{R}^{d_{in}\times d_{out}}$ denotes a weight matrix for the linear transformation of node embeddings, and d_{in} and d_{out} denote the input and output embedding dimensions, respectively. In addition, $\cdot^T$ represents transposition and $\|$ denotes the concatenation operator.

After obtaining the importance between pair of nodes, we can normalize them to make the attention score $\alpha_{ij}^{\mathcal{R}}$ comparable across different nodes, as shown in Figure 2(a). However, since there are two type of links and two type of embeddings for each node in signed network, the normalization process is more complex than unsigned networks. Therefore, we will introduce the normalization process detailedly in the following embedding aggregation process.

Since there is only one type of initial embedding for each node in signed network, we first generate the balanced embedding $\mathbf{h}^{\mathcal{B}(1)}$ and unbalanced embedding $\mathbf{h}^{\mathcal{U}(1)}$, we treat it as the first aggregation process. If we denote the initial embedding of v_i as $\mathbf{h}_i^{(0)} \in \mathbb{R}^{1\times d_{in}}$, the first aggregation layer can be defined as:

$$\mathbf{h}_i^{\mathcal{B}(1)} = \tanh\left(\sum_{j\in\hat{\mathcal{N}}_i^+} \alpha_{ij}^{\mathcal{B}(1)}\mathbf{h}_j^{(0)}\mathbf{W}^{\mathcal{B}(1)}\right) \tag{2}$$

$$\mathbf{h}_i^{\mathcal{U}(1)} = \tanh\left(\sum_{j\in\hat{\mathcal{N}}_i^-} \alpha_{ij}^{\mathcal{U}(1)}\mathbf{h}_j^{(0)}\mathbf{W}^{\mathcal{U}(1)}\right) \tag{3}$$

where $\alpha_{ij}^{\mathcal{B}(1)}$ and $\alpha_{ij}^{\mathcal{U}(1)}$ are the attention scores of nodes from balanced node set and unbalanced node set with respect to v_i. Please note that, we add the self-loop in the aggregation

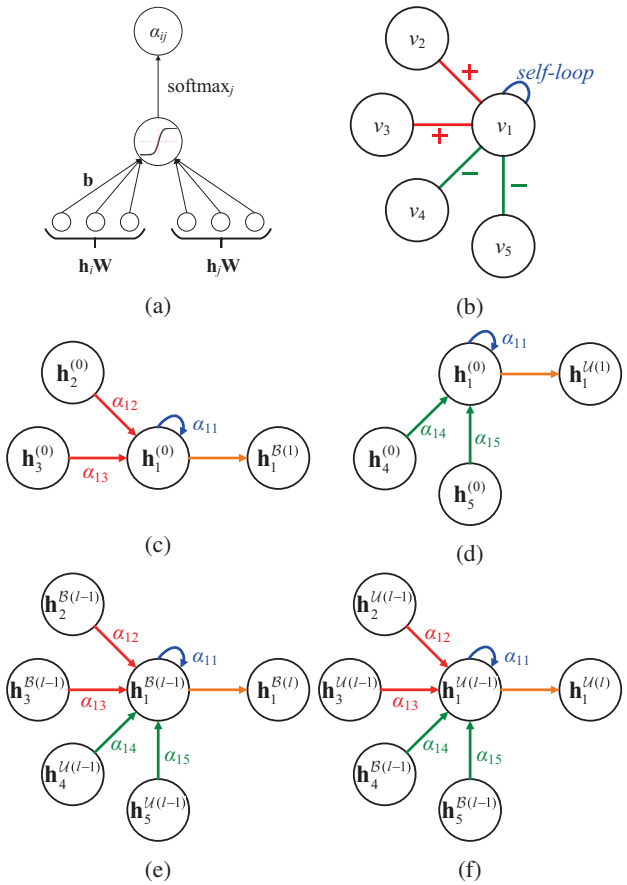


Figure 2: An illustration of the attention mechanism and how SNEA aggregates information from neighboring nodes. (a): The attention mechanism $\tanh(\mathbf{b}(\mathbf{h}_i\mathbf{W}\|\mathbf{h}_j\mathbf{W})^T)$, parameterized by shared weight vector $\mathbf{b}$, applying a $\tanh$ activation. (b): a demo subnetwork for aggregation process. (c) - (d): aggregation processes for the first layer. (e) - (f): aggregation processes for deeper layers (i.e., $l > 1$).

to make sure the embedding will not be lost in the aggregation process. In addition, $\mathbf{W}^{\mathcal{B}(1)}$, $\mathbf{W}^{\mathcal{U}(1)} \in \mathbb{R}^{d_{in}^{(1)} \times d_{out}^{(1)}}$ denote the linear transformation matrices responsible for the information aggregated from $\hat{\mathcal{N}}_i^+$ and $\hat{\mathcal{N}}_i^-$ respectively, and $d_{out}^{(1)}$ denotes the dimension of the output embedding. Taking Figure 2(b) as an example, the aggregation processes for balanced embedding $\mathbf{h}_i^{\mathcal{B}(1)}$ and unbalanced embedding $\mathbf{h}_i^{\mathcal{U}(1)}$ are shown in Figure 2(c) and 2(d).

In the first aggregation layer, the importance of v_j to v_i for balanced embedding $\mathbf{h}_i^{\mathcal{B}(1)}$ and unbalanced embedding $\mathbf{h}_i^{\mathcal{U}(1)}$ can be calculated by:

$$e_{ij}^{\mathcal{B}(1)} = a\left(\mathbf{h}_i^{(0)}\mathbf{W}^{\mathcal{B}(1)}, \mathbf{h}_j^{(0)}\mathbf{W}^{\mathcal{B}(1)}, \mathcal{B}\right) \tag{4}$$

$$e_{ij}^{\mathcal{U}(1)} = a\left(\mathbf{h}_i^{(0)}\mathbf{W}^{\mathcal{U}(1)}, \mathbf{h}_j^{(0)}\mathbf{W}^{\mathcal{U}(1)}, \mathcal{U}\right) \tag{5}$$

By normalizing the importance between pair of nodes, we can get the normalized attention scores $\alpha_{ij}^{\mathcal{B}(1)}$ and $\alpha_{ij}^{\mathcal{U}(1)}$ via softmax function:

$$\alpha_{ij}^{\mathcal{B}(1)} = \frac{\exp(e_{ij}^{\mathcal{B}(1)})}{\sum_{t\in\hat{\mathcal{N}}_i^+}\exp(e_{it}^{\mathcal{B}(1)})} \tag{6}$$

$$\alpha_{ij}^{\mathcal{U}(1)} = \frac{\exp(e_{ij}^{\mathcal{U}(1)})}{\sum_{t\in\hat{\mathcal{N}}_i^-}\exp(e_{it}^{\mathcal{U}(1)})} \tag{7}$$

For the deeper aggregation layers ($l > 1$), the attention-guided aggregation layers can be recursively defined as:

$$\mathbf{h}_i^{\mathcal{B}(l)} = \tanh\left(\sum_{j\in\hat{\mathcal{N}}_i^+, k\in\mathcal{N}_i^-} \alpha_{ij}^{\mathcal{B}(l)}\mathbf{h}_j^{\mathcal{B}(l-1)}\mathbf{W}^{\mathcal{B}(l)} + \alpha_{ik}^{\mathcal{B}(l)}\mathbf{h}_k^{\mathcal{U}(l-1)}\mathbf{W}^{\mathcal{B}(l)}\right) \tag{8}$$

$$\mathbf{h}_i^{\mathcal{U}(l)} = \tanh\left(\sum_{j\in\hat{\mathcal{N}}_i^+, k\in\mathcal{N}_i^-} \alpha_{ij}^{\mathcal{U}(l)}\mathbf{h}_j^{\mathcal{U}(l-1)}\mathbf{W}^{\mathcal{U}(l)} + \alpha_{ik}^{\mathcal{U}(l)}\mathbf{h}_k^{\mathcal{B}(l-1)}\mathbf{W}^{\mathcal{U}(l)}\right) \tag{9}$$

where $\mathbf{W}^{\mathcal{B}(l)}, \mathbf{W}^{\mathcal{U}(l)} \in \mathbb{R}^{d_{in}^{(l)} \times d_{out}^{(l)}}$ denote the shared linear transformation matrices. Taking Figure 2(b) as an example again, the aggregation processes for balanced embedding $\mathbf{h}_i^{\mathcal{B}(l)}$ and unbalanced embedding $\mathbf{h}_i^{\mathcal{U}(l)}$ are shown in Figure 2(e) and 2(f).

Calculating the importance between pair of nodes for deeper layers are more complex than the first aggregation layer. Therefore, we propose to use the following formulas to calculate the importance for different pair of nodes for deeper layers:

$$e_{ij}^{\mathcal{B}(l)} = a\left(\mathbf{h}_i^{\mathcal{B}(l-1)}\mathbf{W}^{\mathcal{B}(l)}, \mathbf{h}_j^{\mathcal{B}(l-1)}\mathbf{W}^{\mathcal{B}(l)}, \mathcal{B}\right) \tag{10}$$

$$e_{ik}^{\mathcal{B}(l)} = a\left(\mathbf{h}_i^{\mathcal{U}(l-1)}\mathbf{W}^{\mathcal{B}(l)}, \mathbf{h}_k^{\mathcal{U}(l-1)}\mathbf{W}^{\mathcal{B}(l)}, \mathcal{B}\right) \tag{11}$$

$$e_{ij}^{\mathcal{U}(l)} = a\left(\mathbf{h}_i^{\mathcal{U}(l-1)}\mathbf{W}^{\mathcal{U}(l)}, \mathbf{h}_j^{\mathcal{U}(l-1)}\mathbf{W}^{\mathcal{U}(l)}, \mathcal{U}\right) \tag{12}$$

$$e_{ik}^{\mathcal{U}(l)} = a\left(\mathbf{h}_i^{\mathcal{B}(l-1)}\mathbf{W}^{\mathcal{U}(l)}, \mathbf{h}_k^{\mathcal{B}(l-1)}\mathbf{W}^{\mathcal{U}(l)}, \mathcal{U}\right) \tag{13}$$

The normalized attention scores can be calculated by the following formulas:

$$\alpha_{ij}^{\mathcal{B}(l)} = \frac{\exp(e_{ij}^{\mathcal{B}(l)})}{\sum_{t\in\hat{\mathcal{N}}_i^+\cup\mathcal{N}_i^-}\exp(e_{it}^{\mathcal{B}(l)})} \tag{14}$$

$$\alpha_{ik}^{\mathcal{B}(l)} = \frac{\exp(e_{ik}^{\mathcal{B}(l)})}{\sum_{t\in\hat{\mathcal{N}}_i^+\cup\mathcal{N}_i^-}\exp(e_{it}^{\mathcal{B}(l)})} \tag{15}$$

$$\alpha_{ij}^{\mathcal{U}(l)} = \frac{\exp(e_{ij}^{\mathcal{U}(l)})}{\sum_{t\in\hat{\mathcal{N}}_i^+\cup\mathcal{N}_i^-}\exp(e_{it}^{\mathcal{U}(l)})} \tag{16}$$

$$\alpha_{ik}^{\mathcal{U}(l)} = \frac{\exp(e_{ik}^{\mathcal{U}(l)})}{\sum_{t\in\hat{\mathcal{N}}_i^+\cup\mathcal{N}_i^-}\exp(e_{it}^{\mathcal{U}(l)})} \tag{17}$$

It's worth noting that when we calculate the attention scores, we use the same type of embedding as the input of the attention mechanism to calculate the importance coefficients, and use the shared weight matrices to build the connections between balanced and unbalanced embeddings. This is because the balanced embedding and unbalanced embedding represent different physical meanings from each other, as nodes from balanced node set and unbalanced node set mean "friends" and "foes" respectively according to balance theory. Therefore, using the same type of embeddings to calculate the importance coefficients can estimate the correlation between pair of nodes more accuracy.

The logic behind the aggregation processes for $\mathbf{h}_i^{\mathcal{B}(l)}$ and $\mathbf{h}_i^{\mathcal{U}(l)}$ are same to the definitions of $\mathcal{B}_i(l)$ and $\mathcal{U}_i(l)$. When we generate balanced embedding $\mathbf{h}_i^{\mathcal{B}(l)}$ for v_i, we aggregate the balanced embeddings from its balanced node set and the unbalanced embeddings from its unbalanced node set. For the unbalanced embedding $\mathbf{h}_i^{\mathcal{U}(l)}$, we aggregate the balanced embeddings from its unbalanced node set and the unbalanced embeddings from its balanced node set. Furthermore, the sum of the attention scores in each graph attentional layer equals to 1, in other words, $\sum_{j\in\hat{\mathcal{N}}_i^+,k\in\mathcal{N}_i^-}\alpha_{ij}^{\mathcal{B}(l)} + \alpha_{ik}^{\mathcal{B}(l)} = \sum_{j\in\hat{\mathcal{N}}_i^+,k\in\mathcal{N}_i^-}\alpha_{ij}^{\mathcal{U}(l)} + \alpha_{ik}^{\mathcal{U}(l)} = 1$, which means that attention scores are comparable across nodes from both the balanced and unbalanced node sets.

After the initial embedding propagates through several aggregation layers mentioned above, we can obtain the balanced and unbalanced embedding $\mathbf{h}_i^{\mathcal{B}(l)}$ and $\mathbf{h}_i^{\mathcal{U}(l)}$, respectively. Then by incorporating $\mathbf{h}_i^{\mathcal{B}(l)}$ and $\mathbf{h}_i^{\mathcal{U}(l)}$, we can obtain the final node embedding for v_i as:

$$\mathbf{h}_i = \tanh([\mathbf{h}_i^{\mathcal{B}(l)} \| \mathbf{h}_i^{\mathcal{U}(l)}]\mathbf{W}^{\mathcal{M}}) \tag{18}$$

where $\mathbf{W}^{\mathcal{M}}$ is the linear transformation matrix responsible for the concatenation of $\mathbf{h}_i^{\mathcal{B}(l)}$ and $\mathbf{h}_i^{\mathcal{U}(l)}$. With the aggregation layers defined above, the embedding generation process of SNEA can be summarized in Algorithm 1.

Algorithm 1 : Embedding Generation Process of SNEA.

Input: Signed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; initial embedding $\{\mathbf{h}_i^0, \forall v_i \in \mathcal{V}\}$; number of aggregation layers L; weight matrices $\mathbf{W}^{\mathcal{M}}$, $\mathbf{W}^{\mathcal{B}(l)}$ and $\mathbf{W}^{\mathcal{U}(l)}$; attentional parameter vectors $\mathbf{b}^{\mathcal{B}(l)}$ and $\mathbf{b}^{\mathcal{U}(l)}$; $l \in \{1, ..., L\}$;
Output: Node embedding $\mathbf{h}_i, \forall v_i \in \mathcal{V}$;

for $v_i \in \mathcal{V}$ **do**
  Calculate $\alpha_{ij}^{\mathcal{B}(1)}$ using Eq. (6);
  Update $\mathbf{h}_i^{\mathcal{B}(1)}$ using Eq. (2);
  Calculate $\alpha_{ij}^{\mathcal{U}(1)}$ using Eq. (7);
  Update $\mathbf{h}_i^{\mathcal{U}(1)}$ using Eq. (3);
end for
if $L > 1$ **then**
  for $l = 2, ..., L$ **do**
    for $v_i \in \mathcal{V}$ **do**
      Calculate $\alpha_{ij}^{\mathcal{B}(l)}$ and $\alpha_{ik}^{\mathcal{B}(l)}$ using Eq. (14) and (15);
      Update $\mathbf{h}_i^{\mathcal{B}(l)}$ using Eq. (8);
      Calculate $\alpha_{ij}^{\mathcal{U}(l)}$ and $\alpha_{ik}^{\mathcal{U}(l)}$ using Eq. (16) and (17);
      Update $\mathbf{h}_i^{\mathcal{U}(l)}$ using Eq. (9);
    end for
  end for
end if
Update $\mathbf{h}_i$ using Eq. (18);

Objective Function and Training

With the structure of our framework as mentioned above, the embedding of each node in signed networks can be generated by incorporating the balanced and unbalanced embeddings as introduced in last subsection. In this subsection, we will introduce the objective function of the proposed framework and the concrete training details. Labels are extremely lacking in the real-world signed networks, however at the same time link type, as an important property, reveals the relationships between nodes. Recall that there are three type of links in signed networks: positive link, negative link and no link, denoted as $\mathcal{S} = \{+, -, ?\}$ - of which "no link" means there exists no link between the pair of nodes. Therefore, to model the framework easier, we transform the optimization problem as a classification problem. To be specific, we construct a mini-batch of nodes $\mathcal{V}_t$, and a set of link triplets $\mathcal{T}$. $\mathcal{T}$ consists of triplets of the form (v_i, v_j, s_{ij}), where $v_i \in \mathcal{V}_t$ or $v_j \in \mathcal{V}_t$, and $s_{ij} \in \mathcal{S}$ denotes which type of link exists between v_i and v_j. We can denote the one-hot coding vector of s_{ij} as $\mathbf{s}_{ij} \in \{0, 1\}^{|\mathcal{S}|}$, and evaluate the cross-entropy error over $\mathcal{T}$ as:

$$\mathcal{L}_{entropy} = \frac{1}{|\mathcal{T}|} \sum_{(v_i, v_j, s_{ij})\in\mathcal{T}} loss(\mathbf{h}_i, \mathbf{h}_j, \mathbf{s}_{ij}) \tag{19}$$

where

$$\begin{aligned} &loss(\mathbf{h}_i, \mathbf{h}_j, \mathbf{s}_{ij}) \\ &= -w_{s_{ij}} \sum_{k=1}^{|\mathcal{S}|} \mathbf{s}_{ij}(k) \log \frac{\exp([\mathbf{h}_i \| \mathbf{h}_j]\boldsymbol{\theta}_k^{src})}{\sum_{s=1}^{|\mathcal{S}|} \exp([\mathbf{h}_i \| \mathbf{h}_j]\boldsymbol{\theta}_s^{src})}, \end{aligned} \tag{20}$$

$\boldsymbol{\theta}^{src}$ denotes the parameters of softmax regression classifier. $w_{s_{ij}}$ denotes the weight associated with link type s_{ij} subject to $\sum_{s_{ij}\in\mathcal{S}} w_{s_{ij}} = 1$. Due to the sparsity of signed networks and the imbalance of positive and negative links, we define different weight $w_{s_{ij}}$ for each link type $s_{ij} \in \mathcal{S}$ according to the number of positive and negative links, and generate "no links" as described in (Derr, Ma, and Tang 2018).

The extended structural balance theory (Qian and Adali 2013; 2014) suggests that nodes are more likely to be more similar to the node with a positive link than a node with a negative link. In addition to this, nodes are more likely to be more similar to the node with a positive link than a node with no link, while nodes are more likely to be more dissimilar to the node with a negative link than a node with no link. More specifically, for v_i, v_j, v_k, $v_t \in \mathcal{V}$ subject

to $(v_i, v_j, +)$, $(v_i, v_k, -)$, $(v_i, v_t, ?) \in \mathcal{T}$. The constraints $\|\mathbf{h}_i - \mathbf{h}_j\|_2^2 < \|\mathbf{h}_i - \mathbf{h}_t\|_2^2$ and $\|\mathbf{h}_i - \mathbf{h}_t\|_2^2 < \|\mathbf{h}_i - \mathbf{h}_k\|_2^2$ should be satisfied. To achieve this goal, for each $v_i \in \mathcal{V}$, we consider the following four cases: (1) if v_i is more similar to v_j than v_t in the embedding space, i.e., $\|\mathbf{h}_i - \mathbf{h}_j\|_2^2 - \|\mathbf{h}_i - \mathbf{h}_t\|_2^2 < 0$, we should not penalize this case; while (2) if v_i is more similar to v_t than v_j in the embedding space, i.e., $\|\mathbf{h}_i - \mathbf{h}_j\|_2^2 - \|\mathbf{h}_i - \mathbf{h}_t\|_2^2 > 0$, we should add a penalty to pull the embedding of v_i be more closer to v_j than v_t; in the similar way, (3) if v_i is more similar to v_t than v_k in the embedding space, i.e., $\|\mathbf{h}_i - \mathbf{h}_t\|_2^2 - \|\mathbf{h}_i - \mathbf{h}_k\|_2^2 < 0$, we should not penalize this case; while (4) if v_i is more similar to v_k than v_t in the embedding space, i.e., $\|\mathbf{h}_i - \mathbf{h}_t\|_2^2 - \|\mathbf{h}_i - \mathbf{h}_k\|_2^2 > 0$, we should add a penalty to pull the embedding of v_i be more closer to v_t than v_k. Based on the aforementioned analysis, we propose the following minimization terms to force v_i closer to v_j than v_k and force v_i closer to v_t than v_k in the embedding space:

$$\mathcal{L}_{pos\_no} = \min \max(0, \|\mathbf{h}_i - \mathbf{h}_j\|_2^2 - \|\mathbf{h}_i - \mathbf{h}_t\|_2^2) \quad (21)$$

$$\mathcal{L}_{neg\_no} = \min \max(0, \|\mathbf{h}_i - \mathbf{h}_t\|_2^2 - \|\mathbf{h}_i - \mathbf{h}_k\|_2^2) \quad (22)$$

Therefore, for all the links in the triplet set $\mathcal{T}$, the objective of extended structural balance theory can be mathematically defined as:

$$\begin{aligned}\mathcal{L}_{structure} = &\frac{1}{|\mathcal{T}_{(+,?)}|} \sum_{(v_i,v_j,+),(v_i,v_t,?)\in\mathcal{T}_{(+,?)}} \mathcal{L}_{pos\_no} \\ &+ \frac{1}{|\mathcal{T}_{(-,?)}|} \sum_{(v_i,v_k,-),(v_i,v_t,?)\in\mathcal{T}_{(-,?)}} \mathcal{L}_{neg\_no}\end{aligned} \quad (23)$$

where $\mathcal{T}_{(+,?)}$ ($\mathcal{T}_{(-,?)}$) is the set of triplets with $s_{ij} \in \{+, ?\}$ ($s_{ij} \in \{-, ?\}$) from $\mathcal{T}$.

To incorporate the extended structural balance theory into signed network embedding, we learn the node embeddings by jointly training the objective function including the objectives of classification task and extended structural balance theory, which can be defined as:

$$\mathcal{L} = \mathcal{L}_{entropy} + \lambda \mathcal{L}_{structure} + \mathcal{L}_{regularizer} \quad (24)$$

where λ denotes the weight of extended structural balance theory objective to balance between classification task and extended structural balance theory, and $\mathcal{L}_{regularizer}$ denotes the variable regularizer of our proposed framework.

When training the neural network, we use Xavier initialization (Glorot and Bengio 2010) to generate values for its parameter matrices, and apply a variant of stochastic gradient descent - AdaGrad (Duchi, Hazan, and Singer 2011) to train the neural network with a mini-batch setting.

Experiments

In this section, we present experiments to evaluate the effectiveness of the proposed framework SNEA. We begin by introducing experimental settings. Then we will measure the quality of the embedding learned by SNEA on signed link prediction task with comparisons with state-of-the-art baseline methods. Finally, we present the parameter sensitivity analysis of SNEA.

Datasets and Baselines

We conduct experiments on four real-world signed networks to evaluate the effectiveness the proposed framework: Bitcoin-Alpha[1], Bitcoin-OTC[2], Epinions[3] and Slashdot[4]. Bitcoin-Alpha and Bircoin-OTC are trading platforms settled in Bitcoins. Since users of the trading platforms are anonymous, users can label other users as trust (positive) or distrust (negative) user to maintain a trading record to prevent transactions from risky users. Epinions is a general consumer review site where users can create positive (or negative) links, if they trust (or distrust) each other. And Slashdot is a technology news site in which users can create positive (or negative) links. Some additional preprocessing was performed on these lager network datasets (Epinions and Slashdot) by filtering out users (nodes) with only a few links. Some key statistics of the network datasets are summarized in Table 1.

Table 1: The statistics of datasets.

Datasets	Bit.Alpha	Bit.OTC	Slashdot	Epinions
$\lvert\mathcal{V}\rvert$	3775	5875	37626	45003
$\lvert\mathcal{E}^+\rvert$	12721	18230	313543	513851
$\lvert\mathcal{E}^-\rvert$	1399	3259	105529	102180

We compare the proposed framework with the following state-of-the-art baseline methods:

- TSVD (Eckart and Young 1936): It is a singular value decomposition method in the form of $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\Sigma \in \mathbb{R}^{d\times d}$ is the diagonal matrix of singular values in descending order, and $\mathbf{U}$, $\mathbf{V} \in \mathbb{R}^{N\times d}$ are the orthonormal matrices corresponding to the selected singular values. We utilize $\mathbf{U}$ as the node embeddings for TSVD.
- SiNE[5] (Wang et al. 2017): It adopts a deep learning framework guided by the structural balance theory to obtain the node embeddings.
- SIDE[6] (Kim et al. 2018): it optimizes the likelihood over both direct and indirect signed connections to encode structural information into node embeddings learning.
- SGCN[7] (Derr, Ma, and Tang 2018): It utilizes balance theory to aggregate and propagate information through graph convolutional layers to generate the node embeddings.
- SiGAT[8] (Huang et al. 2019): it introduces the GAT (Veličković et al. 2018) to signed networks and designs a motif-based graph neural model to learn the node embeddings.

[1] http://www.btc-alpha.com
[2] http://www.bitcoin-otc.com
[3] http://www.epinions.com
[4] http://www.slashdot.com
[5] http://www.public.asu.edu/%7Eswang187/codes/SiNE.zip
[6] https://datalab.snu.ac.kr/side/resources/side.zip
[7] https://www.cse.msu.edu/%7Ederrtyle/code/SGCN.zip
[8] https://github.com/huangjunjie95/SiGAT

For a fair comparison, we set the final embedding dimension as 64 for all the methods. For SiNE, SIDE, SGCN and SiGAT, we use the suggested hyperparameters and settings in their papers. For SNEA, we set λ as 4. Since there is no node features for all the signed network datasets used in our paper, we use final node embeddings (i.e., $\mathbf{U}$) of TSVD as the initial embeddings of SNEA model.

Signed Link Prediction

In this subsection, we measure the quality of node embeddings learned by SNEA on the most fundamental signed network analysis task - signed link prediction. For signed link prediction task, we randomly select 80% links as training set to learn the node embeddings and utilize the remaining links as test set to evaluate the performance. We derive a link feature by combining two embeddings of the connected nodes. As signed link prediction is regarded as binary classification task, therefore we employ a logistic regression classifier to classify positive and negative links, and the performance will be evaluated with Area Under Curve (AUC) and F1-score metrics. We repeat the process 5 times and report the average performance as shown in Table 2 and Table 3.

Table 2: Signed link prediction results with AUC.

Methods	Bit.Alpha	Bit.OTC	Slashdot	Epinions
TSVD	0.740	0.761	0.740	0.766
SiNE	0.781	0.782	0.785	0.831
SIDE	0.642	0.632	0.554	0.617
SGCN	0.801	0.804	0.786	0.849
SiGAT	0.775	0.796	0.789	0.853
SNEA-1	0.766	0.784	0.726	0.822
SNEA	**0.816**	**0.818**	**0.799**	**0.861**

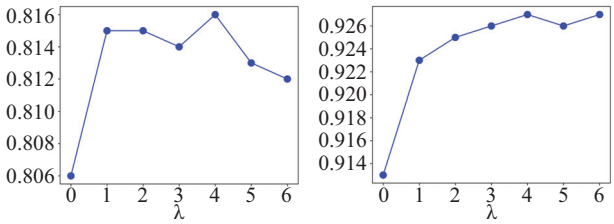
Table 3: Signed link prediction results with F1.

Methods	Bit.Alpha	Bit.OTC	Slashdot	Epinions
TSVD	0.863	0.870	0.804	0.843
SiNE	0.895	0.876	0.850	0.902
SIDE	0.753	0.728	0.624	0.725
SGCN	0.915	0.908	0.859	0.920
SiGAT	0.894	0.903	0.857	0.917
SNEA-1	0.884	0.886	0.801	0.897
SNEA	**0.927**	**0.924**	**0.868**	**0.933**

We denote SNEA-1 as a variant of SNEA that only makes use of the first attentional layer to aggregate informations. We can see that SNEA achieves an apparent performance improvement over SNEA-1 on all the network datasets, which demonstrates the necessity and effectiveness of using balance theory when aggregating and propagating information in signed networks. In comparision with other baseline methods, SNEA outperforms all of them in terms of AUC and F1. Although we use the final node embeddings $\mathbf{U}$ of TSVD as the initial embeddings of SNEA, SNEA achieves a significant improvement over TSVD, which demonstrates the ability of learning node embeddings using SNEA.

Parameter Study

SNEA has two major parameters - λ and the learning rate ϵ for AdaGrad. As ϵ is used for the optimization process, therefore we do not consider ϵ, and only investigate the impact of parameter λ in this subsection. Due to space limit, we only show the parameter analysis results of Bitcoin-Alpha, as we have similar observations on other datasets. We vary λ from $\{0, 1, 2, 3, 4, 5, 6\}$ and show the performance variations of λ in Figure 3. We can see that when $\lambda > 1$, the performance in terms of AUC varies in a narrow range, and the performance in terms of F1 presents a increase trend with the increase of λ. To make a balance between the AUC and F1 performance, we set $\lambda = 4$ in our paper. Furthermore, when $\lambda = 0$, we have a drastic decrease in performance, which demonstrates the necessity of incorporating the extended structural balance theory into signed network embedding.



(a) Bitcoin-Alpha with AUC. (b) Bitcoin-Alpha with F1.

Figure 3: Parameter sensitivity of SNEA w.r.t. λ.

Conclusion

In this paper, we propose a novel signed network embedding framework - SNEA, which utilizes a neural network architecture to learn node embeddings with graph attention machanism. First, we propose a graph attentional layer, which utilizes a masked self-attention mechanism to compute different importance coefficients for different nodes in a neighborhood for aggregation process. Then, we utilize the graph attentional layer and balance theory to learn more discriminative embeddings. Finally, we design a objective function as the objective for optimizing the proposed framework. Extensive experimental results on several real-world signed network datasets demonstrate the effectiveness of the proposed framework through the signed link prediction task. One future direction is to generalize this framework to heterogenous networks.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (NSFC) through grant 61976102. This work is also partially supported by the National Science Foundation (NSF) through grant IIS-1763365 and by FSU.

References

Abu-El-Haija, S.; Perozzi, B.; Al-Rfou, R.; and Alemi, A. A. 2018. Watch your step: Learning node embeddings via graph attention. In *Proceedings of NIPS*, 9180–9190.

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Steeg, G. V.; and Galstyan, A. 2019. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *Proceedings of ICML*, 21–29.

Cartwright, D., and Harary, F. 1956. Structural balance: a generalization of heider's theory. *Psychological review* 63(5):277.

Chen, J.; Ma, T.; and Xiao, C. 2018. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *ICLR*.

Chiang, K.-Y.; Whang, J. J.; and Dhillon, I. S. 2012. Scalable clustering of signed networks using balance normalized cut. In *Proceedings of CIKM*, 615–624.

Derr, T.; Ma, Y.; and Tang, J. 2018. Signed graph convolutional networks. In *Proceedings of ICDM*, 929–934.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(Jul):2121–2159.

Eckart, C., and Young, G. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1(3):211–218.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS*, 249–256.

Heider, F. 1946. Attitudes and cognitive organization. *The Journal of psychology* 21(1):107–112.

Hsieh, C.-J.; Chiang, K.-Y.; and Dhillon, I. S. 2012. Low rank modeling of signed networks. In *Proceedings of KDD*, 507–515.

Huang, J.; Shen, H.; Hou, L.; and Cheng, X. 2019. Signed graph attention networks. In *Proceedings of ICANN*.

Kim, J.; Park, H.; Lee, J.-E.; and Kang, U. 2018. Side: representation learning in signed directed networks. In *Proceedings of WWW*, 509–518.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kunegis, J.; Schmidt, S.; Lommatzsch, A.; Lerner, J.; De Luca, E. W.; and Albayrak, S. 2010. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of SDM*, 559–570.

Kunegis, J.; Preusse, J.; and Schwagereit, F. 2013. What is the added value of negative links in online social networks?. In *Proceedings of WWW*, 727–736.

Lian, D.; Zheng, K.; Zheng, V. W.; Ge, Y.; Cao, L.; Tsang, I. W.; and Xie, X. 2018. High-order proximity preserving information network hashing. In *Proceedings of KDD*, 1744–1753.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 3111–3119.

Qian, Y., and Adali, S. 2013. Extended structural balance theory for modeling trust in social networks. In *Proceedings of PST*, 283–290.

Qian, Y., and Adali, S. 2014. Foundations of trust and distrust in networks: Extended structural balance theory. *ACM Transactions on the Web* 8(3):13.

Qiu, J.; Dong, Y.; Ma, H.; Li, J.; Wang, K.; and Tang, J. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of WSDM*, 459–467.

Shao, J.; Zhang, Z.; Yu, Z.; Wang, J.; Zhao, Y.; and Yang, Q. 2019. Community detection and link prediction via cluster-driven low-rank matrix completion. In *Proceedings of IJCAI*, 3382–3388.

Tang, J.; Aggarwal, C.; and Liu, H. 2016. Node classification in signed social networks. In *Proceedings of SDM*, 54–62.

Tian, F.; Gao, B.; Cui, Q.; Chen, E.; and Liu, T.-Y. 2014. Learning deep representations for graph clustering. In *Proceedings of AAAI*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of NIPS*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Wang, S.; Tang, J.; Aggarwal, C.; Chang, Y.; and Liu, H. 2017. Signed network embedding in social media. In *Proceedings of SDM*, 327–335.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019. Heterogeneous graph attention network. In *Proceedings of WWW*, 2022–2032.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. In *Proceedings of KDD*, 1225–1234.

Yang, C.; Sun, M.; Liu, Z.; and Tu, C. 2017. Fast network embedding enhancement via high order proximity approximation. In *Proceedings of IJCAI*, 19–25.

Yang, L.; Wu, F.; Wang, Y.; Gu, J.; and Guo, Y. 2019. Masked graph convolutional network. In *Proceedings of IJCAI*, 4070–4077.

Yuan, S.; Wu, X.; and Xiang, Y. 2017. Sne: signed network embedding. In *Proceedings of PAKDD*, 183–195.

Zhang, M., and Chen, Y. 2018. Link prediction based on graph neural networks. In *Proceedings of NIPS*, 5165–5175.

Zhang, Z.; Cui, P.; Wang, X.; Pei, J.; Yao, X.; and Zhu, W. 2018. Arbitrary-order proximity preserved network embedding. In *Proceedings of KDD*, 2778–2786.

Zhao, Z.; Gao, B.; Zheng, V. W.; Cai, D.; He, X.; and Zhuang, Y. 2017. Link prediction via ranking metric dual-level attention network learning. In *Proceedings of IJCAI*, 3525–3531.

Zheng, Q., and Skillicorn, D. B. 2015. Spectral embedding of signed networks. In *Proceedings of SDM*, 55–63.