Signed Laplacian Graph Neural Networks

Yu Li[1,8*], **Meng Qu**[2,3], **Jian Tang**[2,4,5†], **Yi Chang**[6,7,8†]
[1] College of Computer Science and Technology, Jilin University, China
[2] Mila - Québec AI Institute, Canada
[3] Univesité de Montréal, Canada
[4] HEC Montréal, Canada
[5] CIFAR AI Research Chair, Canada
[6] School of Artificial Intelligence, Jilin University, China
[7] International Center of Future Science, Jilin University, China
[8] Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, Ministry of Education, China
liyu18@mails.jlu.edu.cn, meng.qu@umontreal.ca, jian.tang@hec.ca, yichang@jlu.edu.cn

Abstract

This paper studies learning meaningful node representations for signed graphs, where both positive and negative links exist. This problem has been widely studied by meticulously designing expressive signed graph neural networks, as well as capturing the structural information of the signed graph through traditional structure decomposition methods, e.g., spectral graph theory. In this paper, we propose a novel signed graph representation learning framework, called Signed Laplacian Graph Neural Network (SLGNN), which combines the advantages of both. Specifically, based on spectral graph theory and graph signal processing, we first design different low-pass and high-pass graph convolution filters to extract low-frequency and high-frequency information on positive and negative links, respectively, and then combine them into a unified message passing framework. To effectively model signed graphs, we further propose a self-gating mechanism to estimate the impacts of low-frequency and high-frequency information during message passing. We mathematically establish the relationship between the aggregation process in SLGNN and signed Laplacian regularization in signed graphs, and theoretically analyze the expressiveness of SLGNN. Experimental results demonstrate that SLGNN outperforms various competitive baselines and achieves state-of-the-art performance.

Introduction

A graph, consisting of a set of nodes and links, is an efficient way to encode the naturally occurring interactions between objects (West et al. 2001). In a number of social and economic contexts, interactions between objects can be represented as signed graphs with positive and negative links. Typically, positive and negative links encode opposite relations between objects, and some examples are friendship/enmity, trust/distrust, and similarity/dissimilarity. For instance, users of Epinions review website can express trust or distrust of each other based on their comments. Such graphs have attracted considerable attention in various applications, such as link sign prediction (Leskovec, Huttenlocher, and Kleinberg 2010), recommendation systems (Yoo, Jo, and Kang 2017), and community analysis (Mercado, Tudisco, and Hein 2016).

Signed graph representation learning, which aims to map nodes to low-dimensional representations, is a fundamental problem for signed graph analysis with a variety of downstream applications. Indeed, graph representation learning has been extensively studied in the machine learning literature, and the representative methods are graph neural networks (GNNs) (Kipf and Welling 2017; Veličković et al. 2018; Li et al. 2020a; Liu et al. 2022). GNNs adopt a message passing scheme to obtain its node representations by aggregating representations from neighbors. However, most of the literature focuses on graphs with only positive links, i.e., unsigned graphs. Until fairly recently, studies have shown that GNN and its variants on unsigned graphs only perform low-pass filtering on node representations, i.e., retain low-frequency information, resulting in similar representations of connected nodes (Nt and Maehara 2019). Therefore, GNNs for unsigned graphs cannot be directly applied to signed graphs, as they cannot effectively distinguish between positive and negative links in signed graphs.

In order to model the coexistence of positive and negative links in signed graphs and characterize the differences between them, most existing signed graph representation learning methods (Derr, Ma, and Tang 2018; Li et al. 2020b; Huang et al. 2021; Liu et al. 2021) resort to the balance theory or its variants (Heider 1946; Cartwright and Harary 1956). The balance theory is a well-known socio-psychological theory, which states that social relationships follow four rules: "the friend of my friend is my friend", "the enemy of my friend is my enemy", "the friend of my enemy is my enemy" and "the enemy of my enemy is my friend". However, the balance theory has been proven equivalent to a simple assumption that the nodes can be divided into two disjoint subsets such that there are only positive links within the subsets and only negative links between them (Mercado, Tudisco, and Hein 2016), which is too idealistic for real world signed graphs. This is a distinct limitation when learn-

[*] Work was done during internship at Mila.
[†] Corresponding authors.

ing node representations that can clearly reveal the underlying structure of signed graphs. Fundamentally, the goal of signed graph representation learning is to make the representations of nodes connected by positive links closer and the representations of nodes connected by negative links farther away in the representation space (Qian and Adali 2014). This naturally raises the question: is there another way to achieve this goal and efficiently model positive as well as negative links in signed graphs?

In this paper, we propose a solution based on spectral graph theory (Chung and Graham 1997) and graph signal processing (Dong et al. 2016). Specifically, with the eigendecomposition of the graph Laplacian matrix, we can theoretically extract information at any frequency by designing a suitable function of the signal frequency (eigenvalues). Particularly, the low-frequency information retains the similarity between connected nodes (Nt and Maehara 2019). In contrast, the high-frequency information highlights the dissimilarity between connected nodes (Park et al. 2019). While in signed graphs, positive and negative links are strongly correlated with similarity and dissimilarity, respectively (Qian and Adali 2014; Tang et al. 2016). Therefore, we can naturally bridge the connection between graph signal processing and signed graph modeling.

Motivated by these analyses, we introduce a Signed Laplacian Graph Neural Network (SLGNN) for signed graph representation learning, which can effectively capture the structural information of signed graphs. To flexibly model positive and negative links, we first divide the signed graph into two subgraphs with a common set of nodes, which contain only positive links and negative links, respectively. Then, we design different graph convolution filters on each subgraph to extract low-frequency and high-frequency information, and combine them into a unified message passing framework. However, low-frequency and high-frequency information contribute differently to modeling positive and negative links, and the tie-strengths of different links are also different. To effectively capture the structural information of signed graphs, a self-gating mechanism is proposed to quantify the impacts of low-frequency and high-frequency information during message passing. From the perspective of numerical optimization, SLGNN is an extension of signed Laplacian regularization in signed graphs. We theoretically analyze the expressiveness of SLGNN and experimentally evaluate its effectiveness. The main contributions of our work can be summarized as follows:

- We propose a novel Signed Laplacian Graph Neural Network framework, named SLGNN, for signed graph representation learning.
- We mathematically establish the relationship between SLGNN and signed Laplacian regularization in signed graphs, and theoretically analyze the expressiveness of SLGNN.
- We validate the effectiveness of SLGNN on real-world signed graphs, showing that SLGNN outperforms various baseline methods and achieves new state-of-the-art performance.

Related Work

In line with the focus of our work, we briefly review the most related work on graph signal processing and signed graph representation learning.

Graph signal processing mathematically bridges the gap between signal processing and spectral graph theory (Chung and Graham 1997), enabling researchers to define graph convolution filters in the spectral domain to process signals in graphs. The work (Bruna et al. 2013) was the first to generalize the convolution operations in CNN on images to graph convolution filters based on the spectrum of the graph Laplacian. Later on, various GNNs were proposed either in the spectral domain or the spatial domain (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Veličković et al. 2018). The work (Balcilar et al. 2020) further establishes the connection between spectral-based and spatial-based GNNs. Nevertheless, most of them have been shown to be low-pass graph convolution filters (Nt and Maehara 2019), which will cause the learned node representations to be too similar to be discernible, i.e., over-smoothing issue. Recently, some works (Bo et al. 2021; Chien et al. 2021) propose to mitigate over-smoothing and graph heterophily issues with adaptive frequency-pass convolution filters. However, these methods are specifically designed for unsigned graphs and therefore cannot be directly applied to signed graphs.

Signed graph representation learning can be traced back to the eigendecomposition of signed Laplacian (Hou, Li, and Pan 2003) and matrix factorization-based methods (Hsieh, Chiang, and Dhillon 2012). Afterwards, most efforts resort to the balance theory or its variants to model signed graphs. For instance, SIDE (Kim et al. 2018) and SIGNET (Islam, Prakash, and Ramakrishnan 2018) propose to adopt the random walk strategy to maintain the structural balance. SGCN (Derr, Ma, and Tang 2018) proposes the first version of signed GCN based on the balance theory. After that come SiGAT (Huang et al. 2019) and SNEA (Li et al. 2020b), they adopt attention mechanism (Veličković et al. 2018) to distinguish the importance of different neighbors. SGDNET (Jung, Yoo, and Kang 2020) proposes to diffuse node representations based on the balance theory. GS-GNN (Liu et al. 2021) further generalizes the balance theory to the k-group theory and learns both local and global node representations. Nevertheless, due to the limitation of the balance theory, these works still suffer from the problem of decreased expressiveness.

In contrast, SLGNN is based on spectral graph theory and graph signal processing as well as GNNs. Thus, SLGNN enjoys the expressive power of GNNs and can capture the structural information of the signed graph with the help of spectral graph theory. From a spectral point of view, the low-frequency and high-frequency information essentially preserve the similarity and dissimilarity between connected nodes, respectively, which is the key to distinguish positive and negative links. By appropriately modeling the similarity and dissimilarity of the low-frequency and high-frequency information through a self-gating mechanism, SLGNN can effectively capture the structural information of the signed graph, resulting in expressive node representations.

Preliminary

Let a signed network be $\mathcal{G} = (\mathcal{G}^+, \mathcal{G}^-)$, where $\mathcal{G}^+ = (\mathcal{V}, \mathcal{E}^+)$ and $\mathcal{G}^- = (\mathcal{V}, \mathcal{E}^-)$ encode positive and negative links, with a common node set $\mathcal{V} = \{v_1, v_2, ..., v_n\}$, and $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$. For each node $v_i \in \mathcal{V}$, its positive and negative neighbors are denoted as $\mathcal{N}_i^+ = \{v_j | (v_i, v_j) \in \mathcal{E}^+\}$ and $\mathcal{N}_i^- = \{v_k | (v_i, v_k) \in \mathcal{E}^-\}$, respectively. Let $\mathcal{A}^+$ and $\mathcal{A}^-$ be the adjacency matrices of $\mathcal{G}^+$ and $\mathcal{G}^-$, then the diagonal matrix of the degrees of $\mathcal{G}^+$, $\mathcal{G}^-$ and $\mathcal{G}$ are denoted as $\mathcal{D}_{ii}^+ = \sum_{j=1}^n \mathcal{A}_{ij}^+$, $\mathcal{D}_{ii}^- = \sum_{j=1}^n \mathcal{A}_{ij}^-$, and $\mathcal{D} = \mathcal{D}^+ + \mathcal{D}^- + \mathcal{I}$, where $\mathcal{I}$ represents the identify matrix. The graph Laplacians of $\mathcal{G}^+$ and $\mathcal{G}^-$ are denoted as $\mathcal{L}^{\mathcal{G}^+} = \mathcal{D}^+ - \mathcal{A}^+$ and $\mathcal{L}^{\mathcal{G}^-} = \mathcal{D}^- - \mathcal{A}^-$. In addition, the symmetric normalized adjacency-, degree-, and Laplacian- matrices of $\mathcal{G}^+$ and $\mathcal{G}^-$ normalized by $\mathcal{D}$ are denoted as $\mathcal{A}_{sym}^{\mathcal{G}^+}$, $\mathcal{D}_{sym}^{\mathcal{G}^+}$, $\mathcal{L}_{sym}^{\mathcal{G}^+}$ and $\mathcal{A}_{sym}^{\mathcal{G}^-}$, $\mathcal{D}_{sym}^{\mathcal{G}^-}$, $\mathcal{L}_{sym}^{\mathcal{G}^-}$.

Spectral Convolution on Graphs. According to spectral graph theory (Chung and Graham 1997), signal on graphs can be filtered using the eigendecomposition of graph Laplacian: $\mathcal{L} = \mathbf{U}\lambda\mathbf{U}^T$, where $\mathbf{U}$ consists of all the eigenvectors of $\mathcal{L}$ and $\lambda = \text{diag}(\lambda_1, ..., \lambda_n)$ is a diagonal matrix consisting of the corresponding eigenvalues as its diagonal. In graph signal processing, λ_i is the signal frequency, and $\mathbf{U}_i$ is the transform basis corresponding to λ_i. Given a signal h, the graph Fourier transform of h can be defined as $\hat{h} = \mathbf{U}^T h$, and its inverse operation as $h = \mathbf{U}^T \hat{h}$. Afterwards, the spectral convolution filter $\mathcal{F}$ can be formulated as: $\mathcal{F} * h = \mathbf{U}\mathbf{f}(\lambda)\mathbf{U}^T h$, where $\mathbf{f}(\lambda)$ is the graph convolution filter function corresponding of the signal frequency λ. By defining the graph convolution kernel as:

$$\mathcal{F} = \mathbf{U}\mathbf{f}(\lambda)\mathbf{U}^T, \tag{1}$$

(Balcilar et al. 2020) connects the spectral-based and spatial-based GNNs in a unified formula:

$$\mathbf{H}^{(l+1)} = \sigma\Big(\sum_{k=1}^{K} \mathcal{F}^k \mathbf{H}^{(l)} \mathbf{W}^{(l,k)}\Big), \tag{2}$$

where σ denotes the nonlinear activation function (such as ReLU), $\mathcal{F}^k$ denotes the k-th graph convolution kernel, $\mathbf{H}^{(l)}$ is the node representations at l-th layer, and $\mathbf{W}^{(l,k)}$ is the learnable parameter matrix.

Proposed Model

In this section, we propose a novel Signed Laplacian Graph Neural Network framework, SLGNN, for signed graph representation learning. Basically, signed graph representation learning entails the following challenges: (1) How to effectively distinguish positive and negative links in the representation space? (2) How to effectively estimate the tie-strengths of different links in the underlying structure of a signed graph? In this work, we propose to address these challenges based on spectral graph theory and graph signal processing. Specifically, we design different graph convolution filters to extract low-frequency and high-frequency information on positive and negative links, respectively. The low-frequency and high-frequency information can effectively preserve the similarity and dissimilarity between connected nodes, which is the key for distinguishing positive links from negative links. After combining the graph convolution filters into a unified message passing framework, in order to effectively capture the structural information of signed graphs, we propose a self-gating mechanism to quantify the impacts of low-frequency and high-frequency information as well as estimate the tie-strengths of links between nodes. With the expressive power of GNNs and the ability of spectral graph theory to encode structural information in signed graphs, SLGNN can learn expressive node representations. The framework overview of SLGNN is shown in Figure 1. In the following sections, we will describe our framework in detail.

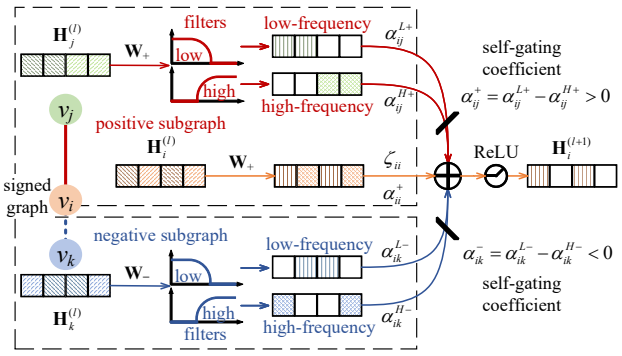

Figure 1: The framework overview of SLGNN.

Graph Convolution Kernels

In signed graphs, the negative links have different properties from the positive ones. In order to flexibly model positive and negative links, we divide the signed graph into two subgraphs, which contain only positive links and negative links, respectively. The following theorem states the connection between the original signed graph and the two subgraphs from the perspective of spectral graph theory.

Theorem 1. *Let* $\mathbf{u}$ *be an eigenvector of* $\mathcal{L}_{sym}^{\mathcal{G}^+}$ *and* $\mathcal{L}_{sym}^{\mathcal{G}^-}$ *with eigenvalues* $\lambda^{\mathcal{G}^+}$ *and* $\lambda^{\mathcal{G}^-}$*, respectively. Then,* $\mathbf{u}$ *is an eigenvector of* $\mathcal{L}_{sym}^{\mathcal{G}}$ *with eigenvalue* $\lambda^{\mathcal{G}^+} + \lambda^{\mathcal{G}^-}$.

Proof. Using the identities $\mathcal{L}_{sym}^{\mathcal{G}^+}\mathbf{u} = \lambda^{\mathcal{G}^+}\mathbf{u}$ and $\mathcal{L}_{sym}^{\mathcal{G}^-}\mathbf{u} = \lambda^{\mathcal{G}^-}\mathbf{u}$, we have $\mathcal{L}_{sym}^{\mathcal{G}}\mathbf{u} = (\mathcal{L}_{sym}^{\mathcal{G}^+} + \mathcal{L}_{sym}^{\mathcal{G}^-})\mathbf{u} = (\lambda^{\mathcal{G}^+} + \lambda^{\mathcal{G}^-})\mathbf{u}$, which concludes the proof. □

With the theorem, we can flexibly model positive and negative links by designing different custom low-pass and high-pass graph convolution filters on each subgraph. In order to avoid the expensive computation of the eigendecomposition of graph Laplacians, we design the low-pass and high-pass filter functions within the linear form of the signal frequency (eigenvalues).

We first design the low-pass filter functions for $\mathcal{G}^+$ and $\mathcal{G}^-$ as:

$$\mathbf{f}_{\mathcal{G}^+}^L(\lambda^{\mathcal{G}^+}) = \zeta + \mathcal{D}_{sym}^{\mathcal{G}^+} - \lambda^{\mathcal{G}^+} \tag{3}$$

$$\mathbf{f}_{\mathcal{G}^-}^L(\lambda^{\mathcal{G}^-}) = \mathcal{D}_{sym}^{\mathcal{G}^-} - \lambda^{\mathcal{G}^-} \tag{4}$$

where ζ is the diagonal matrix with learnable ζ_{ii} in $(0, 1)$, and $\lambda^{\mathcal{G}^+}$ and $\lambda^{\mathcal{G}^-}$ denote the eigenvalues of $\mathcal{L}^{\mathcal{G}^+}_{sym}$ and $\mathcal{L}^{\mathcal{G}^-}_{sym}$, respectively.

Inserting Eq. (3) and Eq. (4) into Eq. (1), the low-pass graph convolution kernels for $\mathcal{G}^+$ and $\mathcal{G}^-$ are derived as:

$$\mathcal{F}^L_{\mathcal{G}^+} = \zeta + \mathcal{D}^{\mathcal{G}^+}_{sym} - \mathcal{L}^{\mathcal{G}^+}_{sym} = \zeta + \mathcal{A}^{\mathcal{G}^+}_{sym} \tag{5}$$

$$\mathcal{F}^L_{\mathcal{G}^-} = \mathcal{D}^{\mathcal{G}^-}_{sym} - \mathcal{L}^{\mathcal{G}^-}_{sym} = \mathcal{A}^{\mathcal{G}^-}_{sym} \tag{6}$$

From the prospective of spatial-based methods, the low-pass filter aggregates information from the node itself and neighboring nodes, making the node representation of each node similar to those of its neighboring nodes.

Different from low-pass filters, high-pass filters amplify the high-frequency signals and attenuate the low-frequency signals. Thus, we design the high-pass filter functions for $\mathcal{G}^+$ and $\mathcal{G}^-$ as:

$$\mathbf{f}^H_{\mathcal{G}^+}(\lambda^{\mathcal{G}^+}) = \zeta - \mathcal{D}^{\mathcal{G}^+}_{sym} + \lambda^{\mathcal{G}^+} \tag{7}$$

$$\mathbf{f}^H_{\mathcal{G}^-}(\lambda^{\mathcal{G}^-}) = -\mathcal{D}^{\mathcal{G}^-}_{sym} + \lambda^{\mathcal{G}^-} \tag{8}$$

Then, the high-pass graph convolution kernels for $\mathcal{G}^+$ and $\mathcal{G}^-$ are derived by inserting Eq. (7) and Eq. (8) into Eq. (1):

$$\mathcal{F}^H_{\mathcal{G}^+} = \zeta - \mathcal{D}^{\mathcal{G}^+}_{sym} + \mathcal{L}^{\mathcal{G}^+}_{sym} = \zeta - \mathcal{A}^{\mathcal{G}^+}_{sym} \tag{9}$$

$$\mathcal{F}^H_{\mathcal{G}^-} = -\mathcal{D}^{\mathcal{G}^-}_{sym} + \mathcal{L}^{\mathcal{G}^-}_{sym} = -\mathcal{A}^{\mathcal{G}^-}_{sym} \tag{10}$$

Viewing from a spatial perspective, the high-pass filter highlights the information difference between node itself and neighboring nodes, making each node far away from its neighbors in the representation space.

Graph Convolution

In order to filter low-frequency and high-frequency information in signed graphs, we efficiently design multiple graph convolution kernels. However, increasing the number of graph convolution kernels increases the number of training parameters as in Eq. (2), and excessive reduction of training parameters will reduce the expressiveness of graph neural networks. To solve this issue, we propose to share training parameters between graph convolution kernels in the same subgraph, and use two coefficients (α^L_{it} and α^H_{it}, s.t., $\alpha^L_{it} \geq 0$, $\alpha^H_{it} \geq 0$, and $\alpha^L_{it} + \alpha^H_{it} = 1$) to quantify the impacts of low-frequency and high-frequency information that each node aggregates from its neighbors. Combining all the graph convolution kernels of $\mathcal{G}^+$ and $\mathcal{G}^-$ into Eq. (2), we reach to our proposed graph convolution in its spatial form:

$$\begin{aligned}\mathbf{H}^{(l+1)} =& \sigma\Big((\alpha^{L+(l)}\mathcal{F}^L_{\mathcal{G}^+} + \alpha^{H+(l)}\mathcal{F}^H_{\mathcal{G}^+})\mathbf{H}^{(l)}\mathbf{W}^{(l)}_+ \\ &+ (\alpha^{L-(l)}\mathcal{F}^L_{\mathcal{G}^-} + \alpha^{H-(l)}\mathcal{F}^H_{\mathcal{G}^-})\mathbf{H}^{(l)}\mathbf{W}^{(l)}_-\Big) \\ =& \sigma\Big((\zeta + \alpha^{L+(l)}\mathcal{A}^{\mathcal{G}^+}_{sym} - \alpha^{H+(l)}\mathcal{A}^{\mathcal{G}^+}_{sym})\mathbf{H}^{(l)}\mathbf{W}^{(l)}_+ \\ &+ (\alpha^{L-(l)}\mathcal{A}^{\mathcal{G}^-}_{sym} - \alpha^{H-(l)}\mathcal{A}^{\mathcal{G}^-}_{sym})\mathbf{H}^{(l)}\mathbf{W}^{(l)}_-\Big)\end{aligned} \tag{11}$$

where $\alpha^{L+(l)}$ ($\alpha^{L-(l)}$) and $\alpha^{H+(l)}$ ($\alpha^{H-(l)}$) denote the coefficient matrices for low-frequency and high-frequency information at l-th layer in $\mathcal{G}^+$ ($\mathcal{G}^-$), respectively.

For each node v_i, its corresponding node representation $\mathbf{H}_i$ at (l+1)-th layer is formulated as:

$$\begin{aligned}\mathbf{H}^{(l+1)}_i =& \sigma\Big(\zeta_{ii}\mathbf{H}^{(l)}_i\mathbf{W}^{(l)}_+ \\ &+ \sum_{v_j \in \mathcal{N}^+_i} \frac{\alpha^{L+(l)}_{ij} - \alpha^{H+(l)}_{ij}}{\Delta^{sym}_{ij}}\mathbf{H}^{(l)}_j\mathbf{W}^{(l)}_+ \\ &+ \sum_{v_k \in \mathcal{N}^-_i} \frac{\alpha^{L-(l)}_{ik} - \alpha^{H-(l)}_{ik}}{\Delta^{sym}_{ik}}\mathbf{H}^{(l)}_k\mathbf{W}^{(l)}_-\Big)\end{aligned} \tag{12}$$

where Δ^{sym}_{ij} and Δ^{sym}_{ik} are the normalization items, whose values are $\sqrt{\mathcal{D}_{ii}\mathcal{D}_{jj}}$ and $\sqrt{\mathcal{D}_{ii}\mathcal{D}_{kk}}$, respectively. In this part, we omit the superscript (l) in the coefficient notations for brevity. For each node v_i, α^{L+}_{ij} (α^{H+}_{ij}) controls the contribution of similar (dissimilar) node representation from node v_j to make the nodes pair (v_i, v_j) become *similar* in $\mathcal{G}^+$, while α^{L-}_{ik} (α^{H-}_{ik}) controls the contribution of similar (dissimilar) node representation from node v_k to make the nodes pair (v_i, v_k) become *dissimilar* in $\mathcal{G}^-$. From the inherent properties of $\mathcal{G}^+$ and $\mathcal{G}^-$, it can be concluded that $\alpha^{L+}_{ij} > \alpha^{H+}_{ij}$ and $\alpha^{L-}_{ik} < \alpha^{H-}_{ik}$. Let $\alpha^+_{ij} = \alpha^{L+}_{ij} - \alpha^{H+}_{ij}$ and $\alpha^-_{ik} = \alpha^{L-}_{ik} - \alpha^{H-}_{ik}$, we can easily derive that $0 \leq \alpha^+_{ij} \leq 1$, and $-1 \leq \alpha^-_{ik} \leq 0$.

For each node pair (v_i, v_t), s.t., $v_t \in \mathcal{N}^+_i \cup \mathcal{N}^-_i$, it is impractical to manually set the coefficients α^+_{it} or α^-_{it}, since we have no prior knowledge about the tie-strength of the node pair. Therefore, we propose to learn these coefficients in a data driven manner and parameterize them as a function of the node representations of connected nodes by designing a self-gating mechanism as:

$$\alpha^\tau_{it} = \varphi^\tau_{it} \cdot \mathrm{sigmoid}(f(\mathbf{H}_i\mathbf{W}_\tau \| \mathbf{H}_t\mathbf{W}_\tau)) \tag{13}$$

where $\tau \in \{+, -\}$. If $v_t \in \mathcal{N}^+_i$, then $\tau = +$ and $\varphi^\tau_{it} = 1$; otherwise, if $v_t \in \mathcal{N}^-_i$, then $\tau = -$ and $\varphi^\tau_{it} = -1$. Further, $\|$ represents the concatenation operator, $f(\cdot)$ is a function modeled as a single layer fully-connected neural network to transform a vector to a scalar, and $\mathrm{sigmoid}(\cdot)$ is the sigmoid function, which maps the output scalar from $f(\cdot)$ to [0, 1].

If we regard the learnable parameter ζ_{ii} as the self-loop coefficient, it can also be learned by Eq. (13) as α^+_{ii}. We further normalize α^+_{ii} to $\alpha^+_{ii}/\Delta^{sym}_{ii}$ to ensure numerical stability in model optimization. By setting the normalized coefficients $\alpha^+_{ii}/\Delta^{sym}_{ii}$, $\alpha^+_{ij}/\Delta^{sym}_{ij}$ and $\alpha^-_{ik}/\Delta^{sym}_{ik}$ as $\alpha^{+'}_{ii}$, $\alpha^{+'}_{ij}$ and $\alpha^{-'}_{ik}$, Eq. (12) can be reformulated as:

$$\begin{aligned}\mathbf{H}^{(l+1)}_i =& \sigma\Big(\alpha^{+'(l)}_{ii}\mathbf{H}^{(l)}_i\mathbf{W}^{(l)}_+ + \sum_{v_j \in \mathcal{N}^+_i} \alpha^{+'(l)}_{ij}\mathbf{H}^{(l)}_j\mathbf{W}^{(l)}_+ \\ &+ \sum_{v_k \in \mathcal{N}^-_i} \alpha^{-'(l)}_{ik}\mathbf{H}^{(l)}_k\mathbf{W}^{(l)}_-\Big),\end{aligned} \tag{14}$$

where $\alpha^{+'(l)}_{ii} > 0$, $\alpha^{+'(l)}_{ij} \geq 0$ and $\alpha^{-'(l)}_{ik} \leq 0$.

Multi-Head Gating

In order to stabilize the learning process and improve the robustness of our model, we use a multi-head gating mechanism to learn the self-gating coefficients like multi-head attention mechanism (Veličković et al. 2018). Concretely, given M independent self-gating heads, each self-gating head m learns the node representations $\mathbf{H}_{i,m}^{(l+1)}$, and these node representations are then concatenated to generate the updated node representations:

$$\mathbf{H}_i^{(l+1)} = \sigma\Big(\Big\|_{m=1}^{M} \mathbf{H}_{i,m}^{(l+1)}\Big). \tag{15}$$

In the final layer L of our model, we average the node representations from multiple heads to obtain the final node representations:

$$\mathbf{H}_i^{(L)} = \sigma\Big(\frac{1}{M}\sum_{m=1}^{M} \mathbf{H}_{i,m}^{(L)}\Big). \tag{16}$$

Objective Function

In signed graphs, link signs reveal the relationships between connected nodes. Thus, we adopt the link sign prediction problem as the training objective for SLGNN.

For each node pair (v_i, v_t), we first construct the link representation by applying an interactive concatenation to the pair of node representations:

$$\mathbf{s}_{it} = [\mathbf{H}_i, \mathbf{H}_i \times \mathbf{H}_t, \mathbf{H}_i - \mathbf{H}_t, \mathbf{H}_t], \tag{17}$$

where $\mathbf{H}_i \times \mathbf{H}_t$ and $\mathbf{H}_i - \mathbf{H}_t$ encode a notion of relative position of two nodes (Lipman, Rustamov, and Funkhouser 2010), which can be also regarded as distance encodings in the representation space (Li et al. 2020a).

A multi-layer perceptron (MLP) is then applied to $\mathbf{s}_{it}$ and produces the probability of node v_i and node v_t forming a positive link:

$$p(v_i, v_t) = \mathrm{Sigmoid}(\mathrm{MLP}(\mathbf{s}_{it}; \theta)) \in \mathbb{R}^1 \tag{18}$$

Finally, we employ the binary cross entropy loss to train our model w.r.t this objective, i.e.,

$$\mathcal{L}oss = \frac{-1}{|\mathcal{E}^+ \cup \mathcal{E}^-|}\Big(\sum_{(v_i,v_t)\in\mathcal{E}^+} \log p(v_i, v_t) + \sum_{(v_i,v_t)\in\mathcal{E}^-} \log(1 - p(v_i, v_t))\Big) \tag{19}$$

Theoretical Analysis

In this section, we establish the connection between SLGNN and signed Laplacian regularization from the perspective of numerical optimization, and theoretically analyze the expressiveness of SLGNN.

Definition 1. (Signed Laplacian Regularization) *Given the noisy node representation $\mathbf{X} \in \mathbb{R}^{N\times d}$ on a signed graph $\mathcal{G}$, the objective of signed graph regularization is to recover an expressive node representation $\mathbf{H} \in \mathbb{R}^{n\times d}$ that can ensure node pairs connected by positive links to be similar while node pairs connected by negative links to be dissimilar, by solving the following optimization problem (Gallier 2016):*

$$\begin{aligned}
&\arg\min_{\mathbf{H}} \mathcal{L}_{reg} \\
&= \sum_{v_i\in\mathcal{V}} \|\mathbf{H}_i - \mathbf{X}_i\|_2^2 + \frac{1}{2}\sum_{v_i\in\mathcal{V}}\sum_{v_j\in\mathcal{N}_i^+} c_{ij}^+ \|\mathbf{H}_i - \mathbf{H}_j\|_2^2 \\
&\quad + \frac{1}{2}\sum_{v_i\in\mathcal{V}}\sum_{v_k\in\mathcal{N}_i^-} c_{ik}^- \|\mathbf{H}_i + \mathbf{H}_k\|_2^2,
\end{aligned}$$

where the constants $c_{ij}^+ > 0$ and $c_{ik}^- > 0$.

The gradient of $\mathcal{L}_{reg}$ with respect to $\mathbf{H}$ focusing on node v_i can be rewritten as:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{reg}}{\partial \mathbf{H}_i} =& 2(\mathbf{H}_i - \mathbf{X}_i) + \sum_{v_j\in\mathcal{N}_i^+} (c_{ij}^+ + c_{ji}^+)(\mathbf{H}_i - \mathbf{H}_j) \\
&+ \sum_{v_k\in\mathcal{N}_i^-} (c_{ik}^- + c_{ki}^-)(\mathbf{H}_i + \mathbf{H}_k).
\end{aligned} \tag{20}$$

Then, the gradient at $\mathbf{X}$ is

$$\begin{aligned}
\frac{\partial \mathcal{L}_{reg}}{\partial \mathbf{H}_i}\Big|_{\mathbf{H}=\mathbf{X}} =& \sum_{v_j\in\mathcal{N}_i^+} (c_{ij}^+ + c_{ji}^+)(\mathbf{X}_i - \mathbf{X}_j) \\
&+ \sum_{v_k\in\mathcal{N}_i^-} (c_{ik}^- + c_{ki}^-)(\mathbf{X}_i + \mathbf{X}_k).
\end{aligned} \tag{21}$$

Thus, the one step of gradient descent starting from $\mathbf{X}$ with the adaptive stepsize b_i can be formulated as:

$$\begin{aligned}
\mathbf{H}_i &\leftarrow \mathbf{X}_i - b_i \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{H}_i}\Big|_{\mathbf{H}=\mathbf{X}} \\
&= \Big(1 - \sum_{v_j\in\mathcal{N}_i^+} b_i(c_{ij}^+ + c_{ji}^+) - \sum_{v_k\in\mathcal{N}_i^-} b_i(c_{ik}^- + c_{ki}^-)\Big)\mathbf{X}_i \\
&\quad + \sum_{v_j\in\mathcal{N}_i^+} b_i(c_{ij}^+ + c_{ji}^+)\mathbf{X}_j - \sum_{v_k\in\mathcal{N}_i^-} b_i(c_{ik}^- + c_{ki}^-)\mathbf{X}_k
\end{aligned} \tag{22}$$

By setting $\hat{\alpha}_{ij}^+ = b_i(c_{ij}^+ + c_{ji}^+)$, $\hat{\alpha}_{ik}^- = b_i(c_{ik}^- + c_{ki}^-)$, and $\hat{\alpha}_{ii}^+ = 1 - \sum_{v_j\in\mathcal{N}_i^+} \hat{\alpha}_{ij}^+ - \sum_{v_k\in\mathcal{N}_i^-} \hat{\alpha}_{ik}^-$, we can get the following update formula:

$$\mathbf{H}_i \leftarrow \hat{\alpha}_{ii}^+ \mathbf{X}_i + \sum_{v_j\in\mathcal{N}_i^+} \hat{\alpha}_{ij}^+ \mathbf{X}_j + \sum_{v_k\in\mathcal{N}_i^-} -\hat{\alpha}_{ik}^- \mathbf{X}_k, \tag{23}$$

which can derive a message aggregation operation similar to that in Eq. (14) by matching the learnable coefficients and transforming the node representation. From a spectral point of view, since $\hat{\alpha}_{ij}^+ > 0$ and $-\hat{\alpha}_{ik}^- < 0$, the message aggregation operation in Eq. (23) essentially aggregates low-frequency information from positive neighbors and high-frequency information from negative neighbors to make nodes connected by positive links become similar and nodes connected by negative links become dissimilar. Thus, SLGNN is equivalent to the extension of signed Laplacian regularization in signed graphs.

In addition to this, we propose to use the squared Euclidean distance as the pair-wise distance of a node pair in the representation space to more intuitively evaluate the expressive power of SLGNN. More specifically, given a node pair (v_i, v_t) and its corresponding node representations $\mathbf{H}_i$ and $\mathbf{H}_t$, the distance between node v_i and node v_t is defined by $\mathbf{dis}_{it} = \|\mathbf{H}_i - \mathbf{H}_t\|_2^2$. Let $\mathbf{dis}_{it}^+$ and $\mathbf{dis}_{it}^-$ be the distance of node representations after one-step message aggregation through positive and negative links, respectively, and set $\zeta_{ii} = \zeta_{tt} = c$, we have

$$\mathbf{dis}_{it}^+ = \|(c\mathbf{H}_i + \mathbf{H}_t) - (c\mathbf{H}_t + \mathbf{H}_i)\|_2^2 = |1 - c|\mathbf{dis}_{it}$$
$$\mathbf{dis}_{it}^- = \|(c\mathbf{H}_i - \mathbf{H}_t) - (c\mathbf{H}_t - \mathbf{H}_i)\|_2^2 = |1 + c|\mathbf{dis}_{it}$$

We can easily observe that $\mathbf{dis}_{it}^+ < \mathbf{dis}_{it} < \mathbf{dis}_{it}^-$, which means SLGNN can make nodes connected by positive links become similar, while nodes connected by negative links become dissimilar.

Note that these analyses are under the assumption that node representations are transformed by the same learnable matrix. In fact, using different learnable transformation matrices for the information propagating on positive and negative links can significantly improve the expressiveness, which will be demonstrated in the experiments.

Experiments

In this section, we conduct experiments to demonstrate the effectiveness of SLGNN on link sign prediction task.

Experimental Settings

Datasets. We evaluate SLGNN on four popular signed graphs:

- **BitcoinAlpha** and **BitcoinOTC** are who-trusts-whom networks of people who trade on Bitcoin platforms and tag the others trust or distrust.
- **Slashdot** is a friendship network of people who tag each other as friends or foes on Slashdot technology-related news website.
- **Epinions** is a who-trust-whom network of people give trust or distrust tags on Epinions consumer review site.

The summary statistics for these signed graph datasets are provided in Table 1. We can see that the number of positive and negative links in these signed graphs is highly imbalanced, with significantly more positive links than negative links. Since these signed graphs do not contain initial node features, as in previous works (Jung, Yoo, and Kang 2020; Li et al. 2020b), we use Truncated-SVD (Halko, Martinsson, and Tropp 2011) to generate their initial node features.

Datasets	#nodes	#pos.links	#neg.links	#pos/#neg
Bit.Alpha	3775	12721	1399	9.09:1
Bit.OTC	5875	18230	3259	5.59:1
Slashdot	37626	313543	105529	2.97:1
Epinions	45003	513851	102180	5.03:1

Table 1: Statistics of the signed graph datasets.

Baselines & Parameter Settings. We compare SLGNN with eight representative signed graph representation learning methods: SIGNET (Islam, Prakash, and Ramakrishnan 2018), SLF (Xu et al. 2019), SGCN (Derr, Ma, and Tang 2018), SiGAT (Huang et al. 2019), SNEA (Li et al. 2020b), SGDNET (Jung, Yoo, and Kang 2020), SDGNN (Huang et al. 2021), and GS-GNN (Liu et al. 2021):

- **SIGNET** proposes a novel random walk strategy to maintain the structural balance in signed graphs.
- **SLF** proposes a signed latent factor model to capture the relationships between nodes in signed graphs.
- **SGCN** first generalizes GCN model to signed graphs based on the balance theory.
- **SiGAT** incorporates graph motifs in signed graphs into GAT model based on the balanced theory and status theory.
- **SNEA** introduces an attention mechanism based on the balance theory to learn the importance of different neighbors in information propagation.
- **SGDNET** designs a random walk technique based on the balance theory to diffuse node representations in signed graphs.
- **SDGNN** proposes to simultaneously reconstruct link signs, link directions and signed directed triangles to model the signed graphs.
- **GS-GNN** generalizes the balance theory to the k-group theory and proposes a dual GNN architecture to learn both the global and the local node representations.

We use the default parameter settings in the officially released codes of the baseline methods. *For a fair comparison*, we change the balanced logistic regression in SGCN and SNEA to unbalanced logistic regression as the link sign predictor, since all the other baselines use unbalanced settings and unbalanced logistic regression for SGCN and SNEA shows better results on the evaluation metrics. For our proposed method SLGNN, we set the numbers of self-gating mechanism to $M = 4$ for BitcoinAlpha, Slashdot and Epinions, and $M = 2$ for BitcoinOTC, and employ 2 message aggregation layers, with a node representation dropout rate of 0.5, a link coefficient dropout rate of 0.5, and the hidden representation dimension of 64. We use AdaGrad (Duchi, Hazan, and Singer 2011) to optimize SLGNN with a learning rate of 0.01, a weight decay of 0.001.

Evaluation Metrics. For each signed graph, we randomly select 20% of the positive and negative links as the test set, while ensuring that the residual signed graph is still *connected* and used as the training set. To comprehensively evaluate the performance of SLGNN, we adopt four types of F1 scores: F1-micro, F1-macro, F1-weighted, F1-binary (abbreviated as F1-MI, F1-MA, F1-WT, F1-BI) and two types of the Area Under the Curve (AUC): AUC-P and AUC-L (taking the estimated target probability and estimated target label as inputs, respectively) as the evaluation metrics. We repeat the experiments 5 times with different randomly split signed graphs, and report the averaged performance.

	Metrics	SIGNET	SLF	SGCN	SiGAT	SNEA	SGDNET	SDGNN	GS-GNN	SLGNN
BitcoinAlpha	F1-MI	92.82±0.31	91.92±0.17	92.65±0.30	92.08±0.24	92.99±0.29	92.30±0.22	92.55±0.32	92.84±0.51	**94.28±0.51**
	F1-MA	74.91±1.33	74.09±0.92	75.67±1.46	72.93±0.92	76.46±1.19	75.22±0.92	78.04±1.03	79.86±1.18	**83.61±0.94**
	F1-WT	91.92±0.40	91.33±0.24	91.98±0.41	91.20±0.27	92.29±0.36	91.72±0.25	92.36±0.34	92.83±0.45	**94.22±0.42**
	F1-BI	96.11±0.17	95.58±0.09	95.99±0.16	95.70±0.13	96.19±0.16	95.79±0.12	95.89±0.17	96.03±0.29	**96.83±0.29**
	AUC-P	92.02±0.56	87.20±0.69	92.31±0.82	87.71±0.78	92.70±0.58	89.41±0.34	91.88±0.52	89.45±1.07	**95.08±0.34**
	AUC-L	70.29±1.26	71.23±1.18	71.99±1.74	69.05±0.97	72.31±1.20	72.14±1.25	76.75±1.21	79.80±1.46	**82.88±0.82**
BitcoinOTC	F1-MI	90.68±0.29	90.91±0.28	91.72±0.25	90.54±0.26	92.28±0.27	91.80±0.47	92.26±0.28	92.73±0.51	**94.48±0.33**
	F1-MA	79.62±0.71	81.31±0.57	82.37±0.57	79.11±0.87	83.45±0.66	83.54±0.90	84.57±0.65	85.87±0.95	**88.99±0.56**
	F1-WT	90.09±0.33	90.64±0.29	91.32±0.27	89.88±0.35	91.87±0.30	91.67±0.46	92.16±0.31	92.73±0.50	**94.41±0.31**
	F1-BI	94.63±0.16	94.70±0.17	95.21±0.14	94.56±0.13	95.54±0.15	95.20±0.28	95.47±0.16	95.72±0.30	**96.77±0.20**
	AUC-P	90.78±0.73	90.61±0.52	93.21±0.28	90.47±0.58	93.84±0.21	92.97±0.67	94.19±0.26	93.37±0.80	**96.87±0.39**
	AUC-L	76.50±0.77	79.67±0.55	79.67±0.63	75.85±1.12	80.46±0.81	82.49±0.87	83.73±0.89	85.87±1.05	**87.95±0.27**
Slashdot	F1-MI	83.28±0.10	84.10±0.07	83.17±0.17	82.93±0.09	84.74±0.10	84.22±0.12	84.05±0.18	87.79±0.05	**87.83±0.09**
	F1-MA	76.22±0.15	77.69±0.12	75.73±0.38	75.73±0.11	78.45±0.12	77.31±0.32	77.98±0.26	**83.27±0.08**	**83.27±0.13**
	F1-WT	82.65±0.11	83.63±0.08	82.40±0.23	82.29±0.08	84.23±0.09	83.53±0.18	83.72±0.19	87.58±0.05	**87.61±0.10**
	F1-BI	89.18±0.07	89.65±0.04	89.17±0.10	88.95±0.06	90.09±0.07	89.82±0.06	89.54±0.12	91.96±0.04	**92.01±0.06**
	AUC-P	88.42±0.08	88.62±0.04	87.75±0.30	87.97±0.07	90.00±0.10	88.98±0.15	88.59±0.11	92.59±0.04	**93.22±0.05**
	AUC-L	74.62±0.15	76.30±0.13	73.89±0.48	74.17±0.10	76.87±0.11	75.43±0.46	76.95±0.27	**82.26±0.12**	82.15±0.16
Epinions	F1-MI	91.25±0.05	91.41±0.14	92.34±0.03	90.75±0.07	92.48±0.05	91.76±0.04	92.71±0.04	94.22±0.03	**94.40±0.08**
	F1-MA	82.29±0.09	83.14±0.44	85.12±0.05	81.74±0.17	85.42±0.08	83.93±0.19	86.02±0.09	89.02±0.09	**89.44±0.18**
	F1-WT	90.71±0.05	91.03±0.19	92.04±0.03	90.31±0.08	92.20±0.05	91.43±0.06	92.48±0.05	94.07±0.04	**94.28±0.09**
	F1-BI	94.89±0.03	94.94±0.07	95.48±0.02	94.56±0.04	95.56±0.03	95.14±0.03	95.69±0.02	96.58±0.01	**96.68±0.04**
	AUC-P	93.19±0.06	92.92±0.39	95.17±0.06	93.00±0.09	95.30±0.07	94.42±0.06	95.06±0.06	96.48±0.05	**97.02±0.06**
	AUC-L	78.99±0.08	80.65±0.68	82.75±0.09	79.16±0.20	83.08±0.09	81.54±0.52	84.00±0.15	87.13±0.22	**87.79±0.36**

Table 2: Link sign prediction results (%, mean±std). The best and second best are bolded and underlined, respectively.

Results & Discussion. Table 2 summarizes the experimental results on link sign prediction. We observe that SLGNN outperforms all the baseline methods on almost all the datasets and evaluation metrics, and GS-GNN generally achieves suboptimal results, followed by SNEA and SDGNN. On BitcoinAlpha and BitcoinOTC, SLGNN significantly outperforms GS-GNN; while on Slashdot and Epinions, GS-GNN performs close to SLGNN, but still underperforms SLGNN. The reason is that SLGNN can adaptively quantify the impacts of low-frequency and high-frequency information in modeling positive and negative links, while effectively capturing the structural information of signed graphs, not only limited to the latent community structure claimed by GS-GNN. For other baseline methods, SNEA and SDGNN are superior to the remaining baseline methods. Specifically, SNEA performs better on F1-micro, F1-binary and AUC-P, while SDGNN reports better results on F1-macro, F1-weighted and AUC-L. To sum up, the significant performance gain of SLGNN indicates that SLGNN can learn more expressive node representations and effectively distinguish positive and negative links in signed graphs.

Ablation Study

We conduct an ablation study on the effectiveness of key components of SLGNN, and name SLGNN without different components as follows - **w/o gating**: SLGNN without self-gating mechanism, i.e., setting $\alpha_{ij}^{+} = 1$ and $\alpha_{ik}^{-} = -1$; **w/o dual**: SLGNN without different learnable transformation matrices for information from positive and negative links, i.e., replacing $\mathbf{W}_{+}^{(l)}$ and $\mathbf{W}_{-}^{(l)}$ with $\mathbf{W}^{(l)}$. Due to space limitation, we only report results on BitcoinAlpha and BitcoinOTC, as we have similar observations on other datasets. The performance comparison of SLGNN and its variants is shown in Table 3. We have the following observations: the performance of SLGNN without self-gating mechanism decreases compared to SLGNN, which implies that adaptively quantifying the impacts of low-frequency and high-frequency information is more conducive to modeling positive and negative links and capturing structural information of signed graphs; and when we only use the same learnable parameter matrix to transform the node information, the performance of SLGNN (w/o dual) significantly reduces, which suggests that using different learnable transformation matrices for the information from positive and negative links can significantly improve the expressiveness of SLGNN.

Visualization of Link Coefficients

To provide an intuitive understanding of the impacts of low-frequency information ($\alpha_{it}^{L\tau}$) and high-frequency information ($\alpha_{it}^{H\tau}$) on positive and negative links from the perspective of the spectral domain, we visualize all the link coefficients ($\alpha_{it}^{\tau} = \alpha_{it}^{L\tau} - \alpha_{it}^{H\tau}$, Eq (13)) from the last layer of SLGNN, as shown in Figure 2. Similar to the ablation study section, we only use BitcoinAlpha and BitcoinOTC. We can

	Metrics	SLGNN	w/o gating	w/o dual
BitcoinAlpha	F1-MI	**94.28**±**0.51**	93.67±0.75	92.26±0.49
	F1-MA	**83.61**±**0.94**	82.55±1.50	79.66±0.78
	F1-WT	**94.22**±**0.42**	93.73±0.64	92.50±0.38
	F1-BI	**96.83**±**0.29**	96.48±0.43	95.67±0.29
	AUC-P	**95.08**±**0.34**	94.04±0.60	90.60±0.82
	AUC-L	**82.88**±**0.82**	82.69±0.69	81.69±0.97
BitcoinOTC	F1-MI	**94.48**±**0.33**	93.85±0.27	93.49±0.34
	F1-MA	**88.99**±**0.56**	87.98±0.40	86.98±0.64
	F1-WT	**94.41**±**0.31**	93.84±0.24	93.40±0.34
	F1-BI	**96.77**±**0.20**	96.38±0.16	96.19±0.21
	AUC-P	**96.87**±**0.39**	96.55±0.23	94.37±0.50
	AUC-L	**87.95**±**0.27**	87.78±0.34	85.96±0.64

Table 3: Ablation study of SLGNN.

observe that the coefficients of negative links are mostly between (-1.0, -0.7) on BitcoinAlpha and BitcoinOTC, which implies that high-frequency information dominates the information aggregation from negative neighbors. Conversely, low-frequency information plays a decisive role in the information aggregation from positive neighbors, as the coefficients of positive links on BitcoinAlpha and BitcoinOTC are distributed in the range of (0.4, 0.9) and (0.4, 0.6), respectively.

Furthermore, to verify the contributions of information from positive and negative links on each individual node from the perspective of the spatial domain, we propose to rank the link coefficients. Specifically, we first convert the link coefficient of a negative link to a positive number by taking its absolute value. Then, for each node with both positive and negative links, we divide all the link coefficients into 10 buckets (labeled from 1 to 10) in the interval between the smallest and largest coefficients, such that coefficients with larger values belong to buckets with higher labels. The density of the number of coefficients of positive and negative links in each bucket is shown in Figure 3. We can observe that the coefficients of negative links are ranked significantly higher than those of positive links, especially on BitcoinOTC. This is because negative links are much sparser than positive links in signed graphs. In order to guarantee that nodes connected by negative links remain dissimilar after information aggregation, node should aggregate as many dissimilar node representations (i.e., high-frequency information) from negative neighbors as possible.

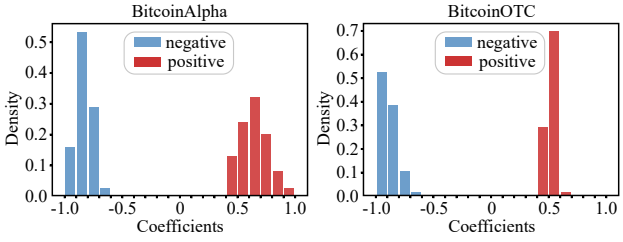

Figure 2: Visualization of link coefficients.

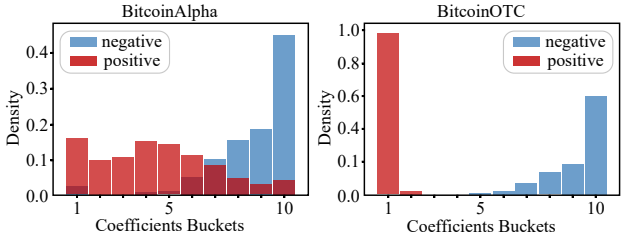

Figure 3: Visualization of link coefficients bucket ranking.

Conclusions

In this work, we propose a novel signed graph representation learning framework - SLGNN, which can learn expressive node representations from signed graphs. Specifically, we design different graph convolution filters to extract low-frequency and high-frequency information on positive and negative links, and combine them into a unified message passing framework. Furthermore, we design a self-gating mechanism to quantify the impacts of low-frequency and high-frequency information for effective capturing the structural information of signed graphs. The expressiveness of SLGNN is theoretically analyzed by proving it being an extension of signed Laplacian regularization in signed graphs. The empirical evaluations on real-world signed graphs show that SLGNN outperforms the baselines and achieves state-of-the-art performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61976102, No.U19A2065), the Science and Technology Development Program of Jilin Province (No.20210508060RQ), the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant, the Canada CIFAR AI Chair Program, collaboration grants between Microsoft Research and Mila, Samsung Electronics Co., Ltd., Amazon Faculty Research Award, Tencent AI Lab Rhino-Bird Gift Fund and a NRC Collaborative R&D Project (AI4DCORE-06). Yu Li is also supported in part by China Scholarship Council (No.202006170168). This work was also partially funded by IVADO Fundamental Research Project grant PRF2019-3583139727.

References

Balcilar, M.; Renton, G.; Héroux, P.; Gauzere, B.; Adam, S.; and Honeine, P. 2020. Bridging the gap between spectral and spatial domains in graph neural networks. *arXiv preprint arXiv:2003.11702*.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the thirty-fifth AAAI Conference on Artificial Intelligence*, 3950–3957.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

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

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive universal generalized pagerank graph neural network. *International Conference on Learning Representations*.

Chung, F. R.; and Graham, F. C. 1997. *Spectral graph theory*, volume 92. American Mathematical Soc.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Derr, T.; Ma, Y.; and Tang, J. 2018. Signed graph convolutional networks. In *Proceedings of the 2018 IEEE International Conference on Data Mining*, 929–934.

Dong, X.; Thanou, D.; Frossard, P.; and Vandergheynst, P. 2016. Learning Laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23): 6160–6173.

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.

Gallier, J. 2016. Spectral theory of unsigned and signed graphs. applications to graph clustering: a survey. *arXiv preprint arXiv:1601.04692*.

Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288.

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

Hou, Y.; Li, J.; and Pan, Y. 2003. On the Laplacian eigenvalues of signed graphs. *Linear and Multilinear Algebra*, 51(1): 21–30.

Hsieh, C.-J.; Chiang, K.-Y.; and Dhillon, I. S. 2012. Low rank modeling of signed networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 507–515.

Huang, J.; Shen, H.; Hou, L.; and Cheng, X. 2019. Signed graph attention networks. In *International Conference on Artificial Neural Networks*, 566–577.

Huang, J.; Shen, H.; Hou, L.; and Cheng, X. 2021. SDGNN: Learning Node Representation for Signed Directed Networks. In *Proceedings of the thirty-fifth AAAI Conference on Artificial Intelligence*, 196–203.

Islam, M. R.; Prakash, B. A.; and Ramakrishnan, N. 2018. Signet: Scalable embeddings for signed networks. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 157–169.

Jung, J.; Yoo, J.; and Kang, U. 2020. Signed Graph Diffusion Network. *arXiv preprint arXiv:2012.14191*.

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

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Leskovec, J.; Huttenlocher, D.; and Kleinberg, J. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, 641–650.

Li, P.; Wang, Y.; Wang, H.; and Leskovec, J. 2020a. Distance Encoding: Design Provably More Powerful Neural Networks for Graph Representation Learning. *Advances in Neural Information Processing Systems*, 33.

Li, Y.; Tian, Y.; Zhang, J.; and Chang, Y. 2020b. Learning signed network embedding via graph attention. In *Proceedings of the thirty-fourth AAAI Conference on Artificial Intelligence*, 4772–4779.

Lipman, Y.; Rustamov, R. M.; and Funkhouser, T. A. 2010. Biharmonic distance. *ACM Transactions on Graphics*, 29(3): 1–11.

Liu, H.; Zhang, Z.; Cui, P.; Zhang, Y.; Cui, Q.; Liu, J.; and Zhu, W. 2021. Signed Graph Neural Network with Latent Groups. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1066–1075.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. 2022. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Mercado, P.; Tudisco, F.; and Hein, M. 2016. Clustering signed networks with the geometric mean of Laplacians. *Advances in neural information processing systems*, 29.

Nt, H.; and Maehara, T. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*.

Park, J.; Lee, M.; Chang, H. J.; Lee, K.; and Choi, J. Y. 2019. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6519–6528.

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.

Tang, J.; Chang, Y.; Aggarwal, C.; and Liu, H. 2016. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3): 1–37.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

West, D. B.; et al. 2001. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River.

Xu, P.; Hu, W.; Wu, J.; and Du, B. 2019. Link prediction with signed latent factors in signed social networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1046–1054.

Yoo, J.; Jo, S.; and Kang, U. 2017. Supervised belief propagation: Scalable supervised inference on attributed networks. In *Proceedings of the 2017 IEEE International Conference on Data Mining*, 595–604.