


## Article

# Learning Embedding for Signed Network in Social Media with Hierarchical Graph Pooling

Jiawang Chen \* and Zhenqiang Wu \* 

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

\* Correspondence: jiawangc@snnu.edu.cn (J.C.); zqiangwu@snnu.edu.cn (Z.W.)

**Abstract:** Signed network embedding concentrates on learning fixed-length representations for nodes in signed networks with positive and negative links, which contributes to many downstream tasks in social media, such as link prediction. However, most signed network embedding approaches neglect hierarchical graph pooling in the networks, limiting the capacity to learn genuine signed graph topology. To overcome this limitation, this paper presents a unique deep learning-based Signed network embedding model with Hierarchical Graph Pooling (SHGP). To be more explicit, a hierarchical pooling mechanism has been developed to encode the high-level features of the networks. Moreover, a graph convolution layer is introduced to aggregate both positive and negative information from neighbor nodes, and the concatenation of two parts generates the final embedding of the nodes. Extensive experiments on three large real-world signed network datasets demonstrate the effectiveness and excellence of the proposed method.

**Keywords:** signed network; network representation learning; balance theory; hierarchical graph pooling



**Citation:** Chen, J.; Wu, Z. Learning Embedding for Signed Network in Social Media with Hierarchical Graph Pooling. *Appl. Sci.* **2022**, *12*, 9795. <https://doi.org/10.3390/app12199795>

Academic Editor: Giacomo Fiumara

Received: 5 September 2022

Accepted: 26 September 2022

Published: 28 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Many social relationships could be represented by assigned networks with positive or negative sign edges. Positive edges in a signed network represent positive ties, such as friends, trust, and support, whereas negative edges represent negative relationships, such as enemies, mistrust, and resistance [1]. In addition, complicated systems exist where positive and negative interactions are implicit. For example, a hyperlink on a website may convey acceptance or disapproval of the target page, depending on the semantics of both pages. It has been proven that such invisible negative links are predictable, and they can convey significantly different feature information than positive links [2]. It is increasingly recognized that mining adversarial relationships in social systems has a wide range of applications in social network analysis, such as social sentiment analysis [3–5] and relationship recommendation [6].

Network embedding refers to learning a fixed-length, low-dimensional vector for each node. It is essential for a variety of social network analysis tasks, such as link prediction, node classification, etc., and it has gained interest in data mining and social computing [7–12]. Recently, network embedding has been greatly improved by graph neural networks (GNNs) because of their tremendous end-to-end modeling capabilities. By combining graph theory and the convolution theorem, Bruna [13] developed the first graph convolutional neural (GCN) network. However, this is hampered by a high level of temporal complexity. In order to address this issue, ChebNet [14] and GCN suggest parametrizing the convolution kernel in spectral approaches to minimize temporal complexity. Despite being spectral approaches, these two have started spatially defining node weight matrices. Inspired by these major approaches, spatial methods began to represent node weights using the attention mechanism, sample mechanism, and aggregate mechanism.

However, traditional GCNs cannot be used on signed networks [15,16] since there exist specific principles led by social theory, such as “the enemy of my enemy is my

friend”, which demonstrates that negative links can have a big impact on the quality of node embeddings. For the purpose of accurately locating the role of the negative links, many tremendous efforts have been devoted to designing network embedding methods, especially for signed networks. For example, SiNE [17] (Signed Network Embedding) learned a low-dimensional vector embedding by using structure balance theory in signed networks. SGCN [18] expanded GCN to signed networks and designed both positive and negative aggregators for the purpose of generating node embeddings based on balance theory. SiGAT [19] integrated attention mechanisms to signed directed networks and built a motifs-based graph neural network model. However, few known approaches take the hierarchical graph pooling of the networks into consideration, which is critical for improving the representation power of a graph, in light of the fact that hierarchical graph pooling can enhance representation learning with high-order structural information.

Motivated by these analyses, a novel signed graph representation framework SHGP to learn Signed network embedding with hierarchical graph pooling is proposed. Instead of only applying aggregators when aggregating information from neighboring nodes, SHGP employs a pooling operator to learn the hierarchical graph pooling of real signed graphs and use the learned hierarchical graph pooling also as part of the node representation.

Our main contributions are threefold:

- A graph pooling layer is introduced to learn effective signed network embedding, which utilizes a pooling mechanism to perform standard global pooling. Further, the essence of this pooling mechanism is to pick a subset of critical nodes to enable the encoding of high-level features and the hierarchical graph pooling of signed networks.
- An objective function that considers signed link prediction and structural balance theory is designed to optimize the framework and learn node representations.
- Extensive experiments on three real-world signed network datasets show the effectiveness of the proposed SHGP framework through the signed link prediction task.

This paper’s structure continues below. Section 2 reviews signed network embedding research. Section 3 provides an explanation of the notations and a priori information of the research content. Section 4 outlines the SHGP framework. Section 5 is dedicated to analyzing the proposed method. Finally, we provide the paper’s conclusion.

## 2. Related Work

### 2.1. Signed Network Embedding

In recent years, there has been a rise in the amount of focus placed on network embedding, which is the process of learning a fixed-length, low-dimension vector representation for each node. It is essential in network analysis and has sparked a great deal of interest in the data mining and machine learning areas [20,21]. However, many of the interactions that exist in the real world might be represented as signed networks with positive or negative sign edges. It has been demonstrated that such negative ties contain substantially different information and meanings than positive ones do [22]. Since traditional network representation learning methods are unable to fully capture the distinct semantic content of positive and negative edge representations, various signed network embedding approaches have been developed in recent years. [23,24]. Tang et al. [25], for instance, suggested a NeLp approach employing the soft edge support vector to handle the problem of negative connection prediction in signed networks. Wang et al. [21] confirmed the status theory of users in trust relationships and calculated the status of users in social networks using the PageRank algorithm. SNE [26] is a signed network node representation learning technique that generates a series of nodes using a random walk strategy. The algorithm of SiNE [17] utilized a multilayer neural network for learning the embedding representation of nodes without using softmax as well as log-likelihood by maximizing the probability of node co-occurrence. SiNE designs a refined strategy based on triangle relations to extract similarities and dissimilarities between nodes, which efficiently and accurately extracts the structural properties of the network.

These methods combine classic machine learning with sociological theories of signed networks (e.g., balance theory) to obtain low-dimensional node representations.

## 2.2. Graph Neural Networks

GNNs seek to implement deep neural networks to structured data. Due to the characteristics of the non-Euclidean data structure, traditional deep neural networks, such as CNNs, cannot be directly used in the graph domains [27]. To solve this problem, GNNs ignore the input order of nodes and propagate the information on each node separately. They can also do the propagation guided by graph structure instead of using it as part of node features. Graph convolution network (GCN) is a classic graph neural network. With the help of the Laplacian matrix, GCN uses feature decomposition and Fourier transform to obtain a convolution kernel to perform convolution on the graph network. GraphSage is an inductive framework that uses vertex feature information to construct embeddings for previously unseen vertices in an efficient manner. GAT assumes that neighbor nodes have different contributions to the target node, and attention is utilized to determine the relevance of each neighboring node to the target node.

With continuous research on graph convolution, researchers have also proposed some signed network embedding methods based on deep graph convolution networks [28]. Signed Graph Convolutional Network (SGCN) [18] first expanded GCN to signed networks and designed both positive and negative aggregators to generate “friend expression” and “enemy expression” for each node in signed networks based on balance theory.

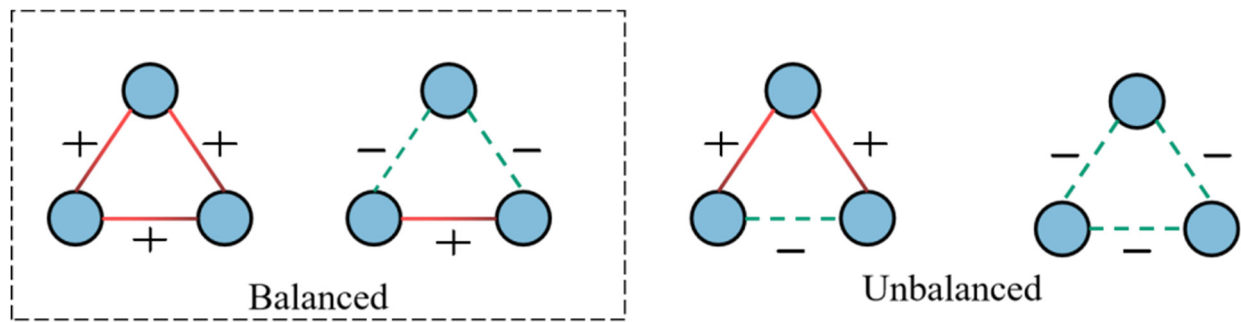
However, the majority of existing approaches ignore the high-order information of the signed graphs. SHGP employs a graph pooling layer, which enables the collection and transmission of more meaningful high-order information to guide the representation learning process.

## 3. Problem Definition

To facilitate presenting, we begin by introducing the primary notations and definitions in this paper. Given a signed network  $G = (V, E)$ , which is constructed of a set  $V = \{v_1, v_2, \dots, v_N\}$  of  $N$  nodes, the set of positive links between nodes can be expressed as  $E^+$ , and the set of negative links can be denoted as  $E^-$ . Note that  $E = E^+ \cup E^-$ . The set of positive and negative neighbors of a node can be expressed as  $N_i^+$  and  $N_i^-$ , respectively.

Similarly, we provide the definition of the set of balance (unbalance) neighbor nodes: in a signed network, if node  $v_i$  and  $v_j$  are connected by an  $L$ -hop path, then  $v_j$  belongs to the set of balance (unbalance) neighbor nodes if there are an odd (even) number of negative links on the  $L$ .

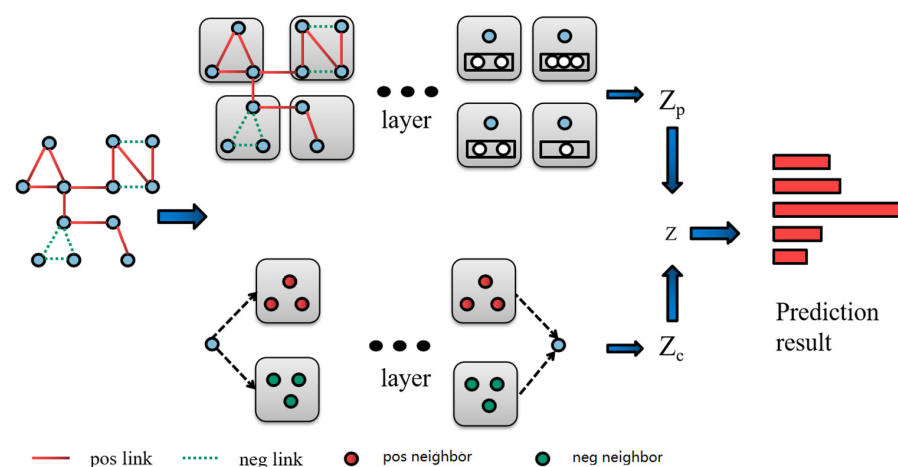
The two triangles on the left of Figure 1. are structural balance triangles. In contrast, in the unbalanced structural triangles on the right, the relationship between a pair of nodes is both friend and foe. Moreover, structural balance in signed networks allows nodes from different node sets to express different features in the embedding space, which means the information aggregated from a balanced node set conveys different characteristics than an unbalanced node set. To solve this problem, we use balance embedding and unbalance embedding to represent one node. In other words, two different types of embedding are leveraged to aggregate the features from two sets of nodes.



**Figure 1.** Balance and unbalance relations in a signed network.

#### 4. Proposed Signed Network Embedding with Hierarchical Graph Pooling

In this part, SHGP is proposed to learn signed network embeddings. Before formalizing the SHGP, we explain its construction. Firstly, traditional GCNs (with only positive links) generate a node representation through aggregate node local neighbors' information and then use the aggregation function to combine the original feature to update the feature of the current node. However, signed networks have valence signs (+, −, 0) on their edges, which carry different information from positive ones. Therefore, different aggregation functions are needed to aggregate features from different types of neighbors. Moreover, hierarchical graph pooling plays a crucial role in network embedding, which has been proven to boost numerous network analysis tasks such as link prediction. As shown in Figure 2, two different aggregators (the red one is the friend relation aggregator while the green one is the foe aggregator) are utilized to combine information from nodes in neighboring neighborhoods. The top of the diagram indicates the hierarchical graph pooling aggregation process, and the white nodes indicate the higher-order nodes involved in the aggregation. The global vector  $Z_p$  together with the convolution vector  $Z_c$  form the final node embedding vector  $Z$ . In addition, a pooling operation is utilized to aggregate hierarchical graph pooling from the signed graph, which computes the various significance coefficients of the neighbors (including high-order neighbors) and generates graph-level embeddings of the nodes. Finally, the signed link prediction results are obtained by the logistic regression classifier. The body of this section is structured as follows: we begin by discussing how to sample and propagate information from a signed network. Then, we explain in depth how hierarchical graph pooling may be utilized to manage the relevance of node embeddings. We conclude with an explanation of how to train this model and perform practical tasks in the signed networks.



**Figure 2.** The framework of SHGP. Given a signed graph with positive and negative links, the lower part of the diagram shows the process of local feature convolution through two types of aggregators, red for the friend relationship aggregator and green for the enemy relationship aggregator.

#### 4.1. Local Signed Graph Convolution

In general, most graph neural networks can be considered message-passing neural networks (MPNNs). The core of MPNNs is the definition of aggregation function and update function between nodes. To begin, the local structural expression of each node is obtained by applying the aggregation function to it and its neighboring nodes. Second, the current node's representation is updated using the update function and the local structural representation. The general expression of the MPNNs can be expressed as follows:

$$m_i^{t+1} = \sum_{j \in N(i)} M_t(h_j^t, e_{i,j}), h_i^{t+1} = U_t(h_i^t + m_i^{t+1}) \quad (1)$$

where  $h_i^t$  denotes the hidden layer representation of node  $i$  at  $t$ -th steps,  $e_{i,j}$  is the features of a given link,  $M_t$  represents the aggregate function at  $t$ -th steps,  $m_i^{t+1}$  means the local structure representation of node  $i$  after aggregating, and  $U_t$  stands for the update function. By designing appropriate sampling and aggregation functions, such as weighted aggregation or mean aggregation, the target node accepts the features passed from its own neighborhood nodes and completes an update of its own features through feature fusion of the local structure to obtain a new feature representation.

A signed network is a specialized type of network that contains type information on its edges. It not only includes two types of connected edges (positive links and negative links) but also has special sociological properties such as structural balance. In particular, the fact that my enemy's enemy is my friend (a foe node two hops from the central node is a friend) makes it infeasible to define aggregation functions only based on edge type. As a result, we use two different GNN aggregators in this paper to aggregate different information from  $\hat{N}_i^+$  and  $\hat{N}_i^-$ . In the first aggregation layer, given the initial feature  $h_i^{(0)}$  of node  $i$ , we can generate the balanced embedding  $h_p^{B(1)}$  and unbalanced embedding  $h_n^{U(1)}$ :

$$h_p^{B(1)} = \sigma\left(\sum_{j \in \hat{N}_i^+} \tilde{F}_{agg} h_i^{(0)} h_j^{(0)}, W^{B(1)}\right) \quad (2)$$

$$h_n^{U(1)} = \sigma\left(\sum_{j \in \hat{N}_i^-} \tilde{F}_{agg} h_i^{(0)} h_j^{(0)}, W^{U(1)}\right) \quad (3)$$

where  $\sigma()$  is the nonlinear activation function,  $\tilde{F}_{agg}$  is the aggregate operation for aggregating feature information from node pairs,  $W^{B(1)}$ ,  $W^{U(1)}$  refers to the linear transformation matrices responsible for the information aggregated from  $\hat{N}_i^+$  and  $\hat{N}_i^-$ . Due to the fact that the first layer of the model can only portray first-order neighbors, there is no structural balance, and friends or enemies can be obtained by direct aggregation. However, from the second layer of the model, the friend representation of node  $i$  will be acquired by the aggregator from its own friends, its own friends' friends, and its own enemies' enemies based on balance theory.

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

$$h_p^{B(l)} = \sigma\left(\sum_{j \in \hat{N}_i^+, k \in \hat{N}_i} \tilde{y}_{ij}^{B(l)} h_j^{B(l-1)} w^{B(l)} + \tilde{y}_{ik}^{B(l)} h_k^{U(l-1)} w^{B(l)}\right) \quad (4)$$

$$h_n^{U(l)} = \sigma\left(\sum_{j \in \hat{N}_i^+, k \in \hat{N}_i^-} \tilde{y}_{ij}^{U(l)} h_j^{U(l-1)} w^{U(l)} + \tilde{y}_{ik}^{U(l)} h_k^{B(l-1)} w^{U(l)}\right) \quad (5)$$

where  $w^{B(l)}$ ,  $w^{U(l)}$  is the shared weight matrix. When the number of aggregation layers is greater than two, the balanced embedding of node  $v_i$  should aggregate the information not only from the balanced node set but also nodes from the unbalanced node set, whose relationship is enemy's enemy.

#### 4.2. Global Signed Graph Aggregation

Hierarchical graph pooling improves network embedding through graph pooling techniques. Convolutional neural networks for Euclidean data use pooling layers. They

minimize the feature graph size and broaden the perceptual field, improving feature generalization and extraction. Graph pooling layers, which are similar to standard convolutional neural networks, are proposed for the generation of graph-level representations. Early graph pooling layers were typically constructed to be flat. They built graph-level representations straight from node-level representations in a single operation. As an illustration, a graph neural network might apply either an average pooling or a maximum pooling to each feature channel in order to create a graph-level representation. Hierarchical graph pooling was later created with the goal of capturing the information contained in a graph by gradually coarsening the original graph, which was developed later. Inspired by the above theory, we designed a hierarchical graph pooling layer on signed networks, which is applied to learn the importance of different nodes from different node sets (balanced node set and unbalanced node set). The operation selects a subset of nodes to form a new but small graph in a signed network when aggregating and propagating information. As shown in Figure 3, given the input graph with four nodes, each of them has three features. Then we can obtain the input feature matrix  $X^\ell \in \mathbb{R}^{4 \times 3}$ . By adding the trainable projection vector  $P$ , the features of the nodes can be mapped to 1D. Then, the top- $k$  nodes with high scores are selected with the help of a sigmoid function, and the index information is recorded. Finally, the index selection process preserves the position order information in the original graph and the adjacency matrix of the new graph  $A^{\ell+1}$  and  $X^{\ell+1}$  can be obtained through the selected index.

As previously stated, node embeddings from the local graph convolution function are classified into two types: positive embedding  $hp$  and negative embedding  $hn$ . These two types of embeddings have different characteristics. Similarly, we performed separate pooling operations to extract the corresponding hierarchical graph pooling from each of the two types of node sets. In the pooling layer, the importance measure of the nodes is learned from the input node features  $F^{(in)}$ :

$$y = \frac{F^{(in)}p}{\|p\|} \quad (6)$$

where  $F^{(in)}$  refers to the input matrix of node features, and  $p$  represents the learning vector. After obtaining the important score  $y$ , we sort all of the nodes and select the top  $K$  importance node, it can be formulated as:

$$idx, \mathcal{P} = rank(y, N_{op}, R) \quad (7)$$

$$idx, \mathcal{N} = rank(y, N_{op}, R) \quad (8)$$

As far as we achieve the  $idx$  of the selected node, the graph structure and node features for the pooled graph are constructed on the basis of this information, and it can be determined by deducing the graph structure of the input graph based on these as follows:

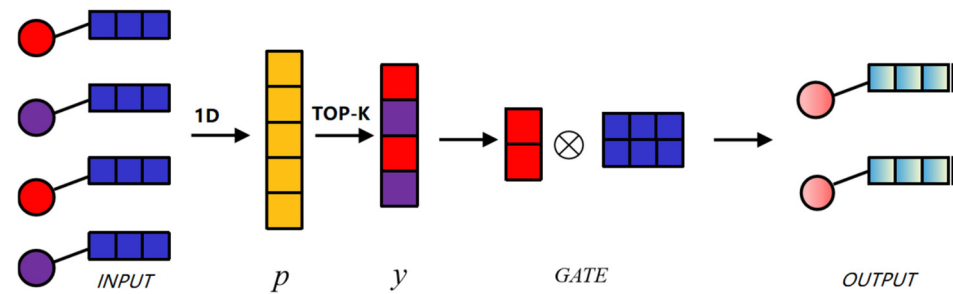
$$\begin{aligned} \tilde{y} &= \sigma(y(idx)) \\ \tilde{F} &= F^{(in)}(idx, :) \\ F_p &= \tilde{F} \odot (\tilde{y} 1_{d_{ip}}^\top) \end{aligned} \quad (9)$$

where  $\sigma()$  denotes the Sigmoid function that maps importance scores to  $(0,1)$ .  $1_{d_{ip}}^\top$  is a vector with all elements being 1.

In this case, it is important to note that we utilize the same kind of nodes as input for the pooling mechanism, and we employ a common matrix to build the link between the balance and unbalance embedded representations. This is due to the fact that balanced embeddings and unbalanced embeddings have distinctly different physical meanings, respectively. Since different types of node sets imply antagonistic relationships such as “trust” or “distrust” [26]. As a result, using the same type of embedding to estimate the



importance ranking can provide a more appropriate estimation of the association between a pair of nodes.



**Figure 3.** An illustration of the proposed signed graph pooling layer with  $k = 2$ . We consider four nodes from balanced node set, and each node has 3-dimension features. The yellow block is the trainable projection vector. After that, we can select the top two nodes. After the gate stage, we can achieve the new output.

#### 4.3. The Objective Function

This section describes the objective function as well as the training details of SHGP. Positive links, negative links, and no links are all represented as  $s = \{+, -, ?\}$ , where “no links” implies that there are no links between node  $v_i$  and  $v_j$ . In the hidden space, we choose to reduce the distance between positive node pairs as much as possible while increasing the distance between negative node pairs. The optimization problem is then transformed into a three-classification problem. We utilize the node mini-batch training approach to build a collection of edge triples, which we then use to test our hypothesis  $\mathcal{T}$ .  $\mathcal{T}$  consists of triplets of the form  $(v_i, v_j, s_{ij})$ , where  $s_{ij} \in s$  refers to which type of link exists between  $v_i$  and  $v_j$ . We use one hot to encode the type triples of the edges  $s_{ij}$  as  $s_{ij} \in \{0, 1\}^S$ , and then the cross-entropy error over  $\mathcal{T}$  can be defined as follow:

$$\mathcal{L}_{entropy} = \frac{1}{\mathcal{T}} \sum_{(v_i, v_j, s_{ij}) \in \mathcal{T}} \text{loss}(h_i, h_j, s_{ij}) \quad (10)$$

$$\text{loss}(h_i, h_j, s_{ij}) = -w_{s_{ij}} \sum_{k=1}^S s_{ij}(k) \log \frac{\exp([h_i \parallel h_j] \theta_k^S)}{\sum_{s=1}^S \exp([h_i \parallel h_j] \theta_s^S)} \quad (11)$$

where  $\theta_k^S$  denotes the parameters of the softmax regression classifier.

We construct a distinct weight for each link type  $s_{ij} \in s$ , depending on the amount of positive and negative connections in the signed networks, and produce “no link”, as stated in [16]. This is undertaken because signed networks are sparse and contain an unbalance of positive and negative links.

According to the extended structural balancing theory, nodes with positive links are close, nodes with negative links are far, and node pairs without ties are in the middle. The triad-based objective function can be mathematically defined as follows:

$$\mathcal{L}_{pos} = \frac{1}{\mathcal{T}_{(+,?)}} \sum_{(v_i, v_j, v_k)} \max\left(0, \left(\|h_i - h_j\|_2^2 - \|h_i - h_k\|_2^2\right)\right) \quad (12)$$

$$\mathcal{L}_{neg} = \frac{1}{\mathcal{T}_{(-,?)}} \sum_{(v_i, v_j, v_k)} \max\left(0, \left(\|h_i - h_k\|_2^2 - \|h_i - h_j\|_2^2\right)\right) \quad (13)$$

where  $\mathcal{T}_{(+,?)}$  and  $\mathcal{T}_{(-,?)}$  are the sets for node pairs from  $\mathcal{T}$ .

Based on the objectives of the edge signed classification and structural balance theory, the overall objective function can be defined as:

$$\mathcal{L} = \mathcal{L}_{entropy} + \lambda[(\mathcal{L}_{pos} + \mathcal{L}_{neg}) + \mathcal{L}_{reg}] \quad (14)$$

where  $\lambda$  denotes the weight of different loss functions, and  $\mathcal{L}_{reg}$  denotes the variable regularizer of our proposed framework.

## 5. Experiment

In this part, the effectiveness of SHGP is evaluated in multiple phases. The experimental conditions are presented first. Next, we show a sensitivity analysis of SHGP parameters. Finally, the quality of the node embeddings using SHGP to the baseline approach is compared.

### 5.1. Simulation Setup

The SHGP is analyzed using Bitcoin-Alpha, Bitcoin-OTC, and Wikirfa [4]. Bitcoin-Alpha and Bitcoin-OTC accept Bitcoin. When trading Bitcoin on the site, users must maintain a good reputation to avoid fraudulent and hazardous transactions. Members can rate each other from  $-10$  (total distrust) to  $+10$  (full trust) in 1-point increments. Note that we treat the score lower than 0 as negative and beyond 0 as positive. Wikirfa is a dataset that includes the Wikipedia admin election data. For an editor in Wikipedia to be promoted to the position of administrator they must make a request for adminship (RfA), and other Wikipedia members may vote in favor, neutrality, or opposition to their request. In Table 1, a complete description of the dataset is shown.

**Table 1.** Statistic of the datasets.

Datasets	Nodes	Positive Links	Negative Links
Bitcoin-Alpha	3783	22,650	1536
Bitcoin-OTC	5881	32,029	3563
Wikirfa	10,835	138,813	39,283

In trials, two unsigned and three signed network embedding methods were compared to the proposed method to illustrate its superiority. In the unsigned network experiment, we removed the negative links in the training stage since these methods cannot distinguish between positive links and negative links:

Deepwalk [18]: This technique simulates text creation by supplying a succession of nodes using random walk paths on the network;

Node2vec [25]: This approach modifies the DeepWalk algorithm's random walk sequence. It provides width-first and depth-first search by using two parameters,  $p$  and  $q$ ;

SiNE [15]: It uses a multilayer neural network for representation learning of nodes based on triangle relations to extract similarities and dissimilarities between nodes;

SIDE [19]: A strategy of random walk generation of node sequences is used with indirect signed connections to encode structural information into node embeddings learning;

SGCN [16]: It uses balance theory to generate node embeddings by designing two node aggregators to aggregate and propagate information through the graph convolution layer.

The final embedding dimension for all of the approaches is set to 64 in order to make a fair comparison. For SiNE, SIDE, and SGCN, we utilize the hyperparameters and settings that were proposed in the respective articles. After obtaining the embeddings of each node, we synthesize the embedding of the two nodes into a concatenated representation and then put the concatenated representation into a logistic regression classifier to obtain the final result of signed link prediction. Our models are implemented by Pytorch = 1.9.1 with Adam optimizer with the learning rate at 0.0001. For the three real datasets, we choose 80% as the training set and 20% as the validation set to test their quality.

### 5.2. Parameter Analysis

This part analyses the hyperparameters of the experiment, including the relationship between epoch and AUC, F1, and the relationship between the number of aggregated layers and AUC. We performed experiments on all three datasets, and because of the space limitations, we will only discuss the performance of the parameters on the BitcoinOTC



dataset. From Figures 4 and 5, we can notice that our model produced some oscillations in the early stages of training due to the randomization of parameters, and when the epoch was greater than 25, the model training stabilized and converged quickly. After the epoch greater than 100, the predictions are stable and will not change significantly.

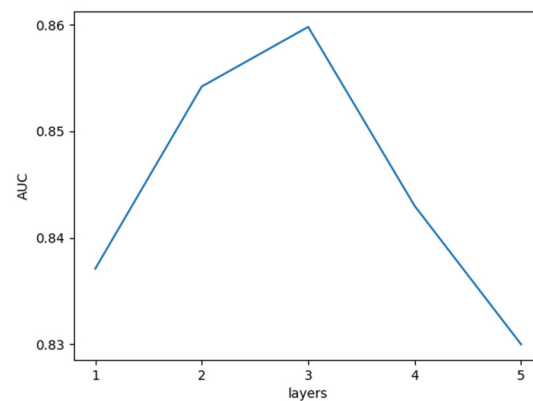


Figure 4. Parameter analysis of embedding layers.

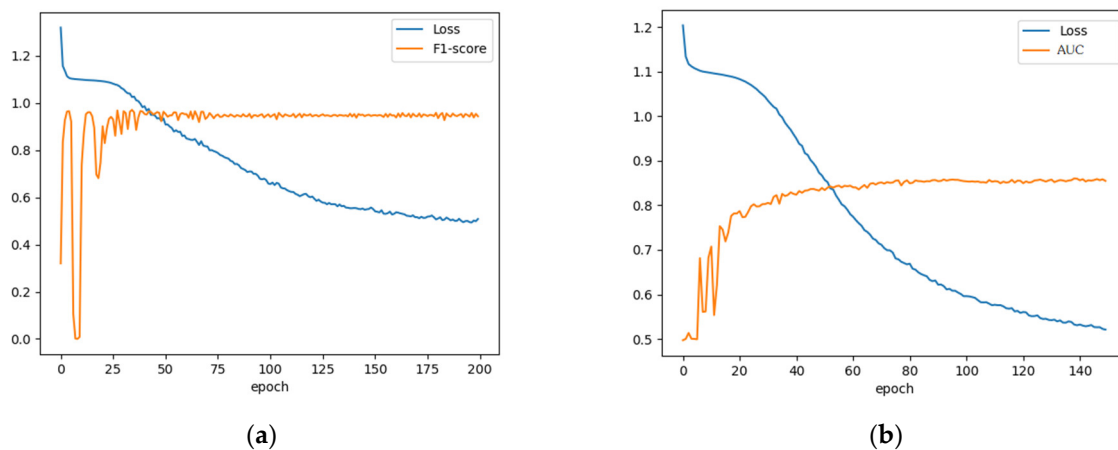


Figure 5. (a) Bitcoin-OTC with train loss and F1-score; (b) Bitcoin-OTC with train loss and AUC.

When discussing the number of aggregation layers, we have normalized the range of values for  $l$  to be from 1 to 5. As can be seen in Figure 5, the AUC gradually increases as the number of layers increases from 1 to 3 and begins to decrease as it increases to 4. This demonstrates that if the network is aggregated lower down, the effect is significantly worse, as the effect of higher-order neighbors on oneself is relatively small. So we set  $L = 3$ . In this work, we set  $k = 3$  since there is a GCN layer before each pooling layer to aggregate information from its first-order neighboring nodes.

### 5.3. Comparison with State-of-the-Art

Signed link prediction is a downstream task of node embedding that we utilize to gauge vector quality. Signed link prediction predicts an edge's sign. We represent link features using two node embedding representations. Signed link prediction can be reduced to categorizing the positive and negative linkages. The performance of the binary classifier will be assessed by the Area Under Curve (AUC) and F1-score metrics. Due to the lack of feature information about the nodes in the real dataset, we use the embeddings obtained in the TVSD method as the initial input for our node features. The results of the AUC and F1-score for the five node embedding methods on three different datasets are shown in Tables 2 and 3, respectively. We can see that:

**Table 2.** Signed Link Prediction Result With AUC.

Dataset	Unsigned Network Embedding		Signed Network Embedding			
	DeepWalk	Node2vec	SiNE	SIDE	SGCN	SHGP
Bitcoin-A	0.641	0.727	0.779	0.757	0.801	0.854
Bitcoin-O	0.614	0.734	0.780	0.764	0.804	0.841
Wikirfa	0.542	0.643	0.740	0.693	0.796	0.812

**Table 3.** Signed Link Prediction Result with F1-score.

Dataset	Unsigned Network Embedding		Signed Network Embedding			
	DeepWalk	Node2vec	SiNE	SIDE	SGCN	SHGP
Bitcoin-A	0.836	0.863	0.895	0.835	0.899	0.936
Bitcoin-O	0.823	0.870	0.876	0.802	0.908	0.925
Wikirfa	0.779	0.787	0.804	0.787	0.895	0.914

For the unsigned network approach, even considering only positive links, the metrics are seen to improve, indicating that network structure plays a crucial role and that Node2vec achieves the best results in this class of methods. After taking the negative links into account, SiNE, SIDE, and SGCN significantly improve the prediction results over other unsigned network methods. These methods combine sociological theory with network embedding and have achieved good results in signed network analysis. SGCN had better performance in the experiment, indicating that graph convolutional neural networks have powerful abilities in feature aggregation.

SHGP achieves a significant performance improvement over these baseline methods on all network datasets. SHGP outperforms all these methods in terms of AUC and F1-score in comparison with other baseline methods. It can be shown that the performance is improved from SGCN on three real-world datasets with 5.3%, 3.7%, and 1.6% when a graph pooling layer is added to the network. This highlights the necessity and efficacy of applying graph pooling procedures as well as balance theory to achieve the desired results.

## 6. Conclusions

In this study, a unique deep-learning-based signed network embedding model with hierarchical graph pooling of the networks is proposed. Specifically, we utilize a pooling layer with a top-k selection process, which can select necessary nodes with higher scores at each update step. Moreover, the high-order relations between the node pairs have been taken into account with the constraint of balance theory. In conclusion, our proposed SHGP model can accurately anticipate the sort of links that exist inside social networks and also be able to accurately forecast how negative associations are utilized within network systems, which helps to influence the design of social computing applications. The proposed model can be able to infer the underlying attitudes of users based on the data from the network and may be used to better recommend friends or favorite items to users in apps on social media. Extensive testing on three large real-world datasets shows the effectiveness of SHGP in signed link prediction tasks. One conceivable approach for the future is to extend the scope of this framework to heterogeneous networks.

**Author Contributions:** Conceptualization, J.C. and Z.W.; methodology, J.C.; software, J.C.; validation, J.C.; formal analysis, J.C.; investigation, J.C.; resources, J.C. and Z.W.; data curation, J.C.; writing—original draft preparation, J.C.; writing—review and editing, J.C. and Z.W.; visualization, J.C.; supervision, J.C.; project administration, J.C.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Sichuan Provinces Key Research and Development Plan Information Secure Sharing and Privacy Protection Technology for IoT Project (2020YFG0292) and the Science and Technology on Communication Security Laboratory of China (6142103190207).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Leskovec, J.; Huttenlocher, D.; Kleinberg, J. Signed networks in social media. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta, GA, USA, 10–15 April 2010.
- Derr, T.; Wang, Z.; Tang, J. Link and Interaction Polarity Predictions in Signed Networks. *Soc. Netw. Anal. Min.* **2020**, *10*, 1–14. [\[CrossRef\]](#)
- Derr, T.; Tang, J. Congressional vote analysis using signed networks. In Proceedings of the IEEE International Conference on Data Mining Workshops, Singapore, 17–20 November 2018.
- Hu, B.; Wang, H.; Wang, L. WSHE: User feedback-based weighted signed heterogeneous information network embedding. *Inf. Sci.* **2021**, *579*, 167–185. [\[CrossRef\]](#)
- Dhelim, S.; Ning, H.; Aung, N. ComPath: User interest mining in heterogeneous signed social networks for Internet of people. *IEEE Internet Things J.* **2020**, *8*, 7024–7035. [\[CrossRef\]](#)
- Ma, H.; Lyu, M.R.; King, I. Learning to recommend with trust and distrust relationships. In Proceedings of the ACM Conference on Recommender Systems, New York, NY, USA, 22–25 October 2009.
- Cui, P.; Wang, X.; Pei, J.; Zhu, W. A survey on network embedding. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 833–852. [\[CrossRef\]](#)
- Berahmand, K.; Nasiri, E.; Rostami, M.; Forouzandeh, S. A modified deepwalk method for link prediction in attributed social network. *Computing* **2021**, *103*, 2227–2249. [\[CrossRef\]](#)
- Liao, L.; He, X.; Zhang, H.; Chua, T.-S. Attributed social network embedding. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 2257–2270. [\[CrossRef\]](#)
- Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015.
- Shao, J.; Zhang, Z.; Yu, Z.; Wang, J.; Zhao, Y.; Yang, Q. Community detection and link prediction via cluster-driven low-rank matrix completion. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019.
- Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; Yang, S. Community preserving network embedding. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
- Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2013**, arXiv:1312.6203.
- Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
- Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
- Mara, A.; Mashayekhi, Y.; Lijffijt, J.; De Bie, T. Csne: Conditional signed network embedding. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020.
- Wang, S.; Tang, J.; Aggarwal, C.; Chang, Y.; Liu, H. Signed network embedding in social media. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2017.
- Derr, T.; Ma, Y.; Tang, J. Signed graph convolutional networks. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018.
- Huang, J.; Shen, H.; Hou, L.; Cheng, X. Signed graph attention networks. In *International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2019.
- Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 April 2014.
- Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
- Zheng, Q.; Skillicorn, D.B. Spectral embedding of signed networks. In Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, 30 April–2 May 2015.
- Beigi, G.; Ranganath, S.; Liu, H. Signed link prediction with sparse data: The role of personality information. In Proceedings of the 2019 World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019.
- Wang, J.; Shen, J.; Li, P.; Xu, H. Online matrix completion for signed link prediction. In Proceedings of the ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017.
- Tang, J.; Chang, S.; Aggarwal, C.; Liu, H. Negative link prediction in social media. In Proceedings of the ACM International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015.

26. Yuan, S.; Wu, X.; Xiang, Y. Sne: signed network embedding. In Proceedings of the PacificAsia Conference on Knowledge Discovery and Data Mining, Jeju, Korea, 23–26 May 2017.
27. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
28. Dhelim, S.; Aung, N.; Kechadi, T.; Ning, H.; Chen, L.; Lakas, A. Trust2Vec: Large-Scale IoT Trust Management System based on Signed Network Embeddings. *IEEE Internet Things J.* **2022**. [[CrossRef](#)]