SigGAN : Adversarial Model for Learning Signed Relationships in Networks

ROSHNI CHAKRABORTY, Indian Institute of Technology Patna, India
RITWIKA DAS, National Institute of Technology Durgapur, India
JOYDEEP CHANDRA, Indian Institute of Technology Patna, India

Signed link prediction in graphs is an important problem that has applications in diverse domains. It is a binary classification problem that predicts whether an edge between a pair of nodes is positive or negative. Existing approaches for link prediction in unsigned networks cannot be directly applied for signed link prediction due to their inherent differences. Further, signed link prediction must consider the inherent characteristics of signed networks, such as, structural balance theory. Recent signed link prediction approaches generate node representations using either generative models or discriminative models. Inspired by the recent success of Generative Adversarial Network (GAN) based models in several applications, we propose a GAN based model for signed networks, SigGAN. It considers the inherent characteristics of signed networks, such as, integration of information from negative edges, high imbalance in number of positive and negative edges and structural balance theory. Comparing the performance with state-of-the-art techniques on 5 real-world datasets validates the effectiveness of SigGAN.

Additional Key Words and Phrases: Signed Networks, Generative Adversarial Networks, Link Prediction, Structural Awareness, Structural Balance

1 INTRODUCTION

The relationships and interactions among users in online social media platforms is characterized as positive, negative or neutral based on the shared opinions and views [8, 59]. Recently, several research works have proposed to represent these relationships through signed networks [75, 76] where the nodes represent the users and the edges are positive or negative depending on the polarity of the interaction/relationship [20, 79]. Several applications of sign prediction between a pair of users [17, 53] include personalized product recommendation [50], understanding of the stance of an user [7, 9], determination of troll or trustworthy posts [72] and identification of malicious users [41]. However, sign prediction has different challenges than the conventional link prediction in traditional unsigned networks [2, 61]. For example, Leskovec et al. [43] have shown that the presence of negatively connected edges must be considered for sign prediction which adds more value than compared to consideration of only positively connected edges. there is a high imbalance between the number of negative and positive edges which makes integration of the information from the negatively connected edges challenging. Additionally, existing theories of social science, like *homophily* and *social influence* [21, 55] does not apply directly to signed networks. While two users who are connected are likely to be similar in unsigned networks, negative link between two users means they are likely to be dissimilar and positive link means they are likely to be similar in signed networks. Therefore, *homophily* and *social influence* does not depend only on the presence/absence of

Authors' addresses: Roshni Chakraborty, Indian Institute of Technology Patna, Patna, India, roshni.pcs15@iitp.ac.in; Ritwika Das, National Institute of Technology Durgapur, Durgapur, India, ritwikadas1@gmail.com; Joydeep Chandra, Indian Institute of Technology Patna, India, joydeep@iitp.ac.in.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.
1556-4681/2022/5-ART $15.00
https://doi.org/10.1145/3532610

Just Accepted

an edge but also on the sign of the edge for signed networks. Similarly, understanding of the sign is pivotal to understand the nature of transitive relationship between any three users [62], such as, *enemy of my friend is my enemy*, *friend of my friend is my friend* and *enemy of my enemy is my friend* are common in signed networks whereas in unsigned networks, two nodes connected to a common node has a higher likelihood to be connected among themselves than two nodes who do not share a common node. Therefore, the existing approaches of link prediction in unsigned networks can not be extended directly to signed link prediction.

To mitigate these issues, existing research works have proposed traditional feature based learning [10, 12, 36, 60] and recently, deep learning approaches [31, 34, 46]. Existing deep learning approaches rely on different forms of node embedding approaches which does not require the feature engineering efforts. Most of these approaches either follow *generative* or *discriminative* learning based models to derive the node embedding. The *generative* model-based representation learning frameworks rely on the conditional distribution of a node, v_b, being a neighbor of node v_a, given the connectivity preferences of v_a. These frameworks learn the likelihood of edges based on this conditional connectivity distribution, and thereby generate the vertex embedding that maximizes the likelihood of the given network connectivity. Several existing works, like SNE [77], SIDE [38], SigNET [33], sign2Vec [3] follow *generative* based approach to generate the node embedding. Subsequently, several existing research works, like SiNE [70], SHINE [69], DNE-SBP [56], have proposed *discriminative* models to predict the presence of an edge based on the characteristics of the node pair, v_a and v_b. However, there are several issues like missing and incomplete data that are quite prevalent in large scale network data that affect the performance of the generative or discriminative based models [24]. GAN based models are highly robust to these issues [24] and highly successful in several applications, such as, information retrieval [49, 80], image generation [18] and collaborative filtering [5, 6]. Furthermore, Wang et al. [68] have proposed a GAN based model to learn the graph structure of traditional unsigned networks and generate node embedding which is highly effective for link prediction in unsigned networks. Therefore, we intuitively believe that a GAN based model might be effective for signed link prediction.

However, existing GAN based models, especially GraphGAN [68] is not directly applicable to signed networks. For example, it needs to handle the challenges of signed networks and consider the specific characteristics of signed networks, like *structural balance theory* [4, 28, 32]. *Structural balance theory* discusses relationships among a group of three people on the basis of the sign of their edges, i.e., three people have higher probability to exist and sustain in nature if either all of them are friends or a couple of them are friends. Therefore, a signed link prediction approach must consider *structural balance theory*, i.e., in a structurally balanced triplet, the sign of the link between a pair of nodes (v_a, v_b) connected to a common neighbor v_c would be positive if the sign of the links (v_a, v_c) and (v_b, v_c) are either both positive or negative. Similarly, the sign of link (v_a, v_b) would be negative if the sign of one of the links to v_c is positive, and the other is negative. In this paper, we propose SigGAN, a GAN based model to generate node representations for signed networks that integrates information from both the signs, considers the structural proximity of the nodes, integrates *structural balance theory* and handles the high imbalance in number of negative and positive edges. SigGAN effectively unifies two models, such as, a binary-class *generative model* that generates likely positive (or, negative), connected neighbors of a node, v_a and a *discriminative model* that identifies whether the generated node pair with the corresponding sign is one from the generated samples or the actual dataset and learns individual parameters for each model to minimize a combined loss function. Validation on real world datasets, such as, *Slashdot*, *Epinions*, *Reddit*, *Bitcoin* and *Wiki-RFA* indicate that the SigGAN can ensure $2-18\%$ higher mean micro F1-score than the existing works in *link prediction*. We also study the effectiveness of SigGAN in *handling sparsity* and *structural balance theory* maintenance. The organization of the paper is as we discuss the existing research works in Section 2 and SigGAN in Section 3. We provide details of the experimental settings and results in Section 4 and Section 5 respectively. We finally draw our conclusions in Section 6.

Just Accepted

2 RELATED WORKS

Existing research works which predict the sign of a link can be categorized into feature based approaches and representation learning based approaches. We discuss the works that belong to either of these categories next.

2.1 Feature Engineering based Approaches

Existing feature engineering based approaches use either *local*, *global* or a combination of both to predict the sign of the link. For example, the *local* attributes include information about common neighbours, structural balance [43, 44], node features [60], etc. Existing approaches have explored the role of user interactions [1], similarity in user characteristics [14], neighborhood information [29], etc., to predict the sign of an edge. While the utilization of the *local* attributes incur low computational cost, it fails in sparse networks. While *global* attributes, like, social hierarchy [48], individual trust [26, 72], structural balance theory [4] and community based ranking [54] can handle the sparsity, they are computationally expensive [37]. To mitigate this computation cost, several research works explore *meso* attributes, like, community structures [15, 16, 37], edge dual properties [78], Katz similarity-based walk [57] and clustering-based approaches [35]. However, utilization of only one type of attribute affects the prediction accuracy of the sign of the link.

To handle these issues, research works explore a combination of *local* and *global* attributes [11, 29, 36]. Though a combination of both attributes can ensure higher accuracy, the extraction of *global* features for all nodes will lead to high computation cost. Further, these existing works ignore the high variance in the network characteristics of the edges that may provide vital cues in the prediction of the sign of an edge [10]. For example, [10] propose an an adaptive system that applies *local*, *global*, or a combination of both based on the characteristics of the edge. However, all of these approaches are highly dependent on the selection of the features and therefore, do not posess the unlimited scope of improved, existing deep learning based approaches inherently integrate both *local* and *global* node attributes are more successful.

2.2 Node Embedding Based Approaches

Several eminent approaches [25, 52, 63, 64, 66] map each node into a low-dimensional vector space and then, use the generated node embedding for link prediction in unsigned networks. However, these approaches can not be directly applied to signed networks due to the inherent differences between signed and unsigned networks. We observe that neighbourhood based approaches for signed networks could be segregated on the basis of the initial representation of a node which is either through the adjacency list [56] of a node or random walk based representations [38, 77] and optimize through cross-entropy or auto-encoder. For example, such as, Yuan et al. [77] consider the co-occurrence of nodes in randomly selected paths and Javari et al. [34] further included the information from the structural role of the nodes into the random walk based information. Further, to handle the high computational cost to incorporate all the possible edges, Kim et al. [38] propose a variant of *negative sampling* for signed networks. However, none of these works consider the higher-order proximity or the signed relationships. Bhowmick et al. [3] address this issue by a trust-based random walk that considers both higher-order neighborhood and trustworthy relationships. However, none of these works explicitly consider structural balance into the generated embedding. To mitigate this, Islam et al. [33] introduce the integration of *structural balance* into random walks. While these approaches follow a *generative* model based learning, there are several research works that follow *discriminative* models, such as, Shen et al. [56] explicitly considers structural balance information while generating the embedding vector of a node by an autoencoder. Lu et al. [47] consider both *status* and *structural balance* and Chen et al. [13] consider *bridge edges*, *status* and *structural balance* to generate the node embedding and Song et al. [58] proposed non-Euclidean network embedding approach to inherently capture the hierarchical structure of networks. Additionally, the huge success of graph neural networks

Just Accepted

in unsigned networks [27, 39, 65] has led to the development of graph neural networks for signed networks. For example, SGCN [19] model utilizes balance theory to integrate the neighbourhood information in GCN. However, SGCN follow an average pooling approach to aggregate information and do not consider the difference in impact due to different neighbours. To alleviate this, Huang et al. [30] proposed attention mechanisms that leverage the graph motif relationships whereas Li et al. [45] and Huang et al. [31] proposed different variants to consider both balance and direction of the neighbors connections for information aggregation respectively. Additionally, Yuan et al. [74] incorporate the multiple facets of a node for embedding and Lin et al. [46] proposed a status aware heterogeneous node embedding for signed networks. Furthermore, although *generative* models and *discriminative* models seem inherently different, Goodfellow et al. [23] have shown that an adversarial learning based model which combines generator and discriminator based models in a minimax game is highly effective for different applications. Therefore, intelligent and iterative integration of these two disjoint classes of representation learning methods through GAN framework has been shown to be efficient and successful for different applications, such as image generation [18, 49], node embedding for graphs [67, 68], collaborative filtering [5, 6] and information retrieval [80]. However, none of these models could be directly applied to signed networks. In this paper, we propose a GAN model for generating node representations specifically for signed networks. We next briefly define the problem followed by the details of the SigGAN.

Just Accepted

3 PROPOSED APPROACH

Let, $G = (V, E, S)$ denote a signed network, where V denotes the set of vertices (nodes), E represents the edges and S comprises of either positive (s) or negative (s') sign. We represent the signs as s and s' for + or − respectively. We assume that we have information about certain edges along with their signs for a network. Thus, for a given node v_j, the objective of the problem is to predict the sign (s or s') of the missing links from v_j. Given v_j, we determine the probability $p(\phi_i|v_j)$, for all nodes $v_i (i \neq j)$, where ϕ_i is a random variable that could be either s or s'. For example, $p(\phi_i = s|v_j)$ denotes the probability that v_i is positively connected to v_j. Although SigGAN can be extended to directed signed networks such that $p(\phi_i|v_j)$ is not equal to $p(\phi_j|v_i)$, it requires design of a discriminator model that consider the directions of the edges. As we do not explicitly consider the direction of the edge, SigGAN can not handle situations where $p(\phi_i|v_j)$ is not equal to $p(\phi_j|v_i)$.

Like GraphGAN [67, 68] for unsigned networks, SigGAN comprises of a generator, J and a discriminator, D who acts as opponents in a minimax game. Additionally, we explicitly consider the characteristics of signed networks, such as, *structural balance theory* property [33], *signed homophily* and the usual requirements that have been considered in GraphGAN [68], such as, *structural awareness* and *low computational complexity*. Therefore, we propose SigGAN as a graph adversarial network based model to generate node embedding for signed networks

SigGAN learns the underlying signed adjacency distribution of each node, v_j across all nodes in V, i.e., $p_{true}(\phi_i|v_j)$, where $\phi_i \in S$ represents the sign of the link from v_j to v_i. The task of $J(\phi_i|v_j; \theta_J)$ is to generate fake positively (or, negatively) connected neighbors of v_j which are similar to its true negatively (or, positively) connected neighbors and the task of $D(v_i, v_j, \phi_i; \theta_D)$ is to distinguish between this actual and fake positively (or, negatively) connected neighbors of v_j. However, as previously discussed, the number of negative edges is very less compared to the number of positive edges and the cost of forming a negative edge is higher than a positive edge [56]. Therefore, to ensure SigGAN can predict both negative and positive edge with high precision, we select negative edges with equal probability to positive edges to train D. D determines if the generated neighbors

of v_j are fake or true through $\mathcal{V}(J, D)$ as :

$$\min_{\hat{J}} \max_{\hat{D}} \mathcal{V}(J, D) = \sum_{j=1}^{V} (\mathbb{E}_{v \sim p_{true}(\cdot|v_j)}[\log D(v, v_j, \phi; \theta_D)] + \mathbb{E}_{v \sim J(\cdot|v_c;\theta_J)}[\log(1 - D(v, v_j, \phi; \theta_D)]) \quad (1)$$

θ_D represents the vector representations of all nodes v_i which can be the potential neighbors of v_j. The optimal parameters of J and D are learned by alternate maximization and minimization of the value function $\mathcal{V}(J, D)$. At each iteration, D is trained using a batch of true samples from $p_{true}(\cdot|v_j)$ and fake samples generated using $J(\cdot|v_j)$ and J is updated using a policy gradient. The continuous competition between D and J ensures generation of a suitable representation of the node, v_j that satisfies the required properties. Furthermore, J ensures :

(1) It generates a probability of the sign of the edges from V_i to other nodes.
(2) It is structurally aware, i.e., the probability of the existence of an edge to a far away node tends to zero.
(3) It maintains *structural balance theory*.
(4) It is computationally efficient.

We discuss the functions of D and J next.

3.1 Discriminator

Although there is a plethora of models that can be used as D, we do not investigate the suitability of the models and leave it as a future goal for this paper. In SigGAN, we use a dense network to determine node representations that can be compared to derive the likely chances of them being connected with a given sign ϕ. The model uses a single hidden layer, in which the number of units equals the dimension of the node representations, say k. The input to the model is the one-hot vector of a node in the network and each unit in the output layer corresponds to a node in the network which is activated by a sigmoid function. The output of the sigmoid function of the i^{th} output unit measures the possibility of node v_i being connected to v_j with sign ϕ_i. Suppose, $h_j \in \mathbb{R}^n$ is the one hot encoding vector of the input node v_j, $W \in \mathbb{R}^{n \times k}$ is the weight matrix of whose i^{th} column elements represent the corresponding weights of the links from each of the inputs to the i^{th} unit of the hidden layer and $d_j = W^T h_j$ represents the hidden unit vector of length k. Similarly, $W' \in \mathbb{R}^{k \times n}$ represents a matrix of the weights of hidden units to the output layer, where the $[ij]^{th}$ element of W' represents the weight of the link connecting the i^{th} hidden layer unit to the j^{th} output unit. Thus, the output o_i of each of the output units can be represented as $o_i = \sigma(W'_{.i} \cdot d_j)$ which is a k dimension embedding of the node v_i, represented as d_i. Thus, if $d_i, d_j \in \mathbb{R}^k$ are the k dimensional embedding vector of nodes v_i and v_j and ϕ represents the sign of the edge between v_i and v_j, then the discriminator function would be represented as

$$D(v_i, v_j, \phi) = \sigma(\phi d_i^T d_j) = \frac{1}{1 + \exp(-\phi d_i^T d_j)} \quad (2)$$

Therefore, as seen in Equation 2, when ϕ is positive, D returns values near 1 if d_i and d_j are similar which shows that the nodes with similar representations are more likely to be connected by a positive edge. Similarly, when ϕ is negative, the discriminator returns a higher score if d_i and d_j are highly dissimilar. The discriminator is trained based on the loss function stated in Equation 1 and we use stochastic gradient ascent to update θ_D, the vector representations of v_i for d_j.

3.2 Generator

Contrary to D, the objective of J is to generate links of v_j with the sign that mimics the true adjacency and sign distribution $p_{true}(\phi_i|v_j)$. So, J is a function $J(\phi_i|v_j)$ that generates node representations to minimize the loss

Just Accepted

function

$$\mathbb{E}_{v\sim J(\cdot|v_j;\theta_J)}[\log(1 - D(v_i, v_j, \phi; \theta_D)], \tag{3}$$

which is the log probability of D that it correctly identifies fake link samples generated by J. Since J samples through the discrete space v_i to increase the probability score, to minimize the loss function we derive its gradient with respect to θ_J using policy gradient. Thus, if $N(v_j)$ are the neighbors of v_j, then

$$\begin{aligned}
& \nabla_{\theta_J} \sum_{j=1}^{|V|} \left[\mathbb{E}_{v\sim J(\cdot|v_j;\theta_J)}[\log(1 - D(v, v_j, \phi; \theta_D)]\right] \\
= \; & \nabla_{\theta_J} \sum_{j=1}^{|V|} \sum_{i=1}^{N(v_j)} J(\phi_i|v_j)[\log(1 - D(v_i, v_j, \phi_i; \theta_D)] \\
= \; & \sum_{j=1}^{|V|} \sum_{i=1}^{N(v_j)} \nabla_{\theta_J} J(\phi_i|v_j)[\log(1 - D(v_i, v_j, \phi_i; \theta_D)] \\
= \; & \sum_{j=1}^{|V|} \sum_{i=1}^{N(v_j)} J(\phi_i|v_j) \times \\
& \left[\nabla_{\theta_J} \log(J(\phi_i|v_j))\right] \left[\log(1 - D(v_i, v_j, \phi_i; \theta_D))\right]
\end{aligned} \tag{4}$$

Therefore, Equation 4 shows that a higher value of $\log(1 - D(v_i, v_j, \phi_i; \theta_D))$ for a given v_i will lower the probability of generating v_i with respect to v_j. We use a Softmax function for the generator $J(\phi_i|v_j)$ which is given as

$$J(\phi_i = s|v_j)) \quad = \quad \frac{e^{s(g_i^T g_j)}}{\sum_{k\neq j} e^{s(g_k^T g_j)}} \tag{5}$$

As evident from Equation 5, for v_j, J generates a neighbor v_i with sign s based on the representations, g_i and g_j of the nodes v_i and v_j, respectively. A similar representation between the two nodes increases the probability of them being positively connected, whereas dissimilar representations increase the chances of a negative link. These representations are suitably derived based on the loss function stated in Equation 4 using gradient descent. However, a major problem with the softmax function is the high computation involved in updating the gradients. As evident from Equation 5, the gradient must be calculated for all the vertices and $J(\phi_i|v_j)$ updates the representations of $|V|$ nodes. Further, the softmax function ignores the rich structural information of the graphs [68]. Existing approximation techniques, such as, negative sampling and subsampling [51] do not consider the structural information. Therefore, we propose modified softmax specifically designed to handles the issues of signed networks.

3.3 Modified Softmax for Signed Graphs

In this Section, we discuss the proposed variant of the softmax function that can be used for the signed graphs. Although GraphGAN [68] proposed a modified softmax function that maintain normalization, graph structure awareness and computational efficiency for unsigned networks, it is not applicable for signed networks. Further, an additional requirement for the softmax function of the signed networks to consider is the *structural balance theory*. We consider the following four properties:

(1) *Normalization* : The function must be normalized such that it produces a valid probability distribution, i.e., $\sum_{i\neq j} \sum_{t\in S} J(\phi_i = t|v_j) = 1$.

Just Accepted

(2) *Graph Structure aware*: The function must consider the graph structure to calculate the connectivity distribution.
(3) *Computationally Efficient* : Updating the parameters for gradient descent of θ_J must be computationally efficient.
(4) *Structurally Balanced* : The function must maintain *structural balance theory*.

We next discuss the proposed softmax approach. Given a node, we introduce *sign-specific relevance* probability of its neighbor. For a node v_j, the positive relevance probability of its neighbor v_i is given as

$$p(\phi_i = s|v_j) = \frac{e^{s(g_j^T g_i)}}{\sum_{k \in N(v_j)} \sum_{t \in S} e^{t(g_j^T g_k)}}, \tag{6}$$

whereas the negative relevance probability is obtained by replacing s by s'. The sign-specific relevance probability indicates how likely a neighbor v_i of node v_j would be connected by either s or s'. Similar to the GraphGAN model, for each node v_j, we initially make a Breadth First Search (BFS) traversal with v_j as the root node. Let $v_{r_0} \rightarrow v_{r_1} \rightarrow \cdots \rightarrow v_{r_m}$ be a path in the BFS tree with root at v_{r_0}. Let $\pi_{n,s}^{n-1} = p(\phi_{r_n} = s|v_{r_{n-1}})$ be the positive relevance probability of node v_{r_n} with respect to its uplink parent $v_{r_{n-1}}$. Therefore, $\pi_{m,s}^0$ is the relevance probability of the node at hop m (written as sub-script of π) from the node at hop 0 (written at super-script of π) connected by an edge with positive sign, s. From *structural balance theory*, we can derive the sign of the edge connecting node at level 0 and m through the level, $m-1$. For example, on the basis of *structural balance theory* for triads, the sign of the edge between node at level 0 and m would be positive, s, if the sign of the edge node at level 0 and $m-1$ is s and the sign of the edge node at level $m-1$ and m is s. Additionally, the sign of the edge between node at level 0 and m would also be positive if the sign of the edge node at level 0 and $m-1$ is s' and the sign of the edge node at level $m-1$ and m is s'. Similarly, we can find the the sign of the edge between node at level 0 and m would be negative if the sign of the edge between node at level 0 and $m-1$ and between $m-1$ and m is different. Therefore, we can extend the positive and negative relevance probability, respectively, for nodes that are more than one hop away by recursive formulations given as

$$\begin{aligned} \pi_{m,s}^0 &= (\pi_{m-1,s}^0 \pi_{m,s}^{m-1} + \pi_{m-1,s'}^0 \pi_{m,s'}^{m-1}) \\ \pi_{m,s'}^0 &= (\pi_{m-1,s}^0 \pi_{m,s'}^{m-1} + \pi_{m-1,s'}^0 \pi_{m,s}^{m-1}) \end{aligned} \tag{7}$$

We are interested in determining the modified softmax function $J(\phi_{r_n}|v_{r_0})$ for any arbitrary node v_{r_n} in the tree path with respect to the root node v_{r_0}. Assuming that the graph is strongly connected, a BFS tree with respect to a node as root will traverse all the nodes through a single unique path. This assumption holds true for any undirected graph; several real-world directed networks are also strongly connected with a large core and hence, can be applied to these. Thus, we define $J(\phi_{r_n}|v_{r_0})$ as

$$\begin{aligned} J(\phi_{r_n} = s|v_{r_0}) &= \pi_{n,s}^0 \pi_{n-1,s}^n + \pi_{n,s'}^0 \pi_{n-1,s'}^n \\ J(\phi_{r_n} = s'|v_{r_0}) &= \pi_{n,s}^0 \pi_{n-1,s'}^n + \pi_{n,s'}^0 \pi_{n-1,s}^n \end{aligned} \tag{8}$$

We prove using appropriate theorems that the generator function $J(\phi_{r_n} = s|v_{r_0})$ satisfies the first three requirements that are mentioned above, whereas, we intuitively show that the modified softmax can also learn the structural balance property if the same exists in the network.

Theorem 1. *For a given node v_j, using the modified softmax we get $\sum_{i \neq j} \sum_{t \in S} J(\phi_i = t|v_j) = 1$*

Proof. We start by showing that for any sub-tree ST_m (rooted at v_{r_m}) of the BFS tree with v_{r_0} as the root, the total sum of the softmax scores of all the nodes in the sub-tree, with respect to v_{r_0}, is the softmax score of v_{r_m} calculated with respect to v_{r_0}. We use this concept for the sub-trees rooted at the child nodes of the BFS tree of

Just Accepted

v_{r_0}. From the expression of sign-specific relevance probability stated in Equation 6, we can directly conclude that for a given node v_j, if $N(v_j)$ be its neighbors then $\sum_{v_i \in N(v_j)} \sum_{t \in S} p(\phi_i = t|v_j) = 1$.

Initially, we consider the case where the subtree is rooted at the node v_{r_m}, which has only leaf nodes as its children. Let the children be denoted as $v_{cm_1}, v_{cm_2}, \ldots, v_{cm_l}$. So,

$$\sum_{i \in ST_{r_m}} \sum_{t \in S} J(\phi_i = t|v_{r_0})$$

can be written as

$$\sum_{t \in S} J(\phi_{r_m} = t|v_{r_0}) + \sum_{i \in C(v_{r_m})} \sum_{t \in S} J(\phi_{cm_i} = t|v_{r_0}),$$

where $C(v_{r_m})$ denotes the child nodes of v_{r_m}. From Equation 7, we have

$$\begin{aligned}
&\sum_{t \in S} J(\phi_{r_m} = t|v_{r_0}) \\
= \;& (\pi^0_{m-1,s}\pi^{m-1}_{m,s} + \pi^0_{m-1,s'}\pi^{m-1}_{m,s'})\pi^m_{m-1,s} \\
+ \;& (\pi^0_{m-1,s}\pi^{m-1}_{m,s'} + \pi^0_{m-1,s'}\pi^{m-1}_{m,s})\pi^m_{m-1,s'} \\
+ \;& (\pi^0_{m-1,s}\pi^{m-1}_{m,s} + \pi^0_{m-1,s'}\pi^{m-1}_{m,s'})\pi^m_{m-1,s'} \\
+ \;& (\pi^0_{m-1,s}\pi^{m-1}_{m,s'} + \pi^0_{m-1,s'}\pi^{m-1}_{m,s})\pi^m_{m-1,s} \\
= \;& (\pi^0_{m-1,s}\pi^{m-1}_{m,s} + \pi^0_{m-1,s'}\pi^{m-1}_{m,s'})(\pi^m_{m-1,s} + \pi^m_{m-1,s'}) \\
+ \;& (\pi^0_{m-1,s}\pi^{m-1}_{m,s'} + \pi^0_{m-1,s'}\pi^{m-1}_{m,s})(\pi^m_{m-1,s} + \pi^m_{m-1,s'}) \\
= \;& \pi^0_{m,s}(\pi^m_{m-1,s} + \pi^m_{m-1,s'}) + \pi^0_{m,s'}(\pi^m_{m-1,s} + \pi^m_{m-1,s'})
\end{aligned} \tag{9}$$

A similar set of derivations for $\sum_{i \in C(v_{r_m})} \sum_{t \in S} J(\phi_{cm_i} = t|v_{r_0})$ yields

$$\begin{aligned}
&\sum_{i \in C(v_{r_m})} \sum_{t \in S} J(\phi_{cm_i} = t|v_{r_0}) \\
= \;& \sum_{i \in C(v_{r_m})} [(\pi^0_{m,s}\pi^m_{cm_i,s} + \pi^0_{m,s'}\pi^m_{cm_i,s'})(\pi^{cm_i}_{m,s} + \pi^{cm_i}_{m,s'}) \\
+ \;& (\pi^0_{m,s}\pi^m_{cm_i,s'} + \pi^0_{m,s'}\pi^m_{cm_i,s})(\pi^{cm_i}_{m,s} + \pi^{cm_i}_{m,s'})]
\end{aligned} \tag{10}$$

Since for a leaf node $\pi^{cm_i}_{m,s} + \pi^{cm_i}_{m,s'} = 1$, Equation 10 can be expressed as

$$\sum_{i \in C(v_{r_m})} \left[\pi^0_{m,s}(\pi^m_{cm_i,s} + \pi^m_{cm_i,s'}) + \pi^0_{m,s'}(\pi^m_{cm_i,s} + \pi^m_{cm_i,s'})\right] \tag{11}$$

One may note that for the node v_{r_m}, from Equation 6 we get $\pi^m_{m-1,s} + \pi^m_{m-1,s'} + \sum_{i \in C(v_{r_m})} \left[\pi^m_{cm_i,s} + \pi^m_{cm_i,s'}\right] = 1$ and thus adding Equations 9 and 11, we get

$$\sum_{i \in ST_{r_m}} \sum_{t \in S} J(\phi_i = t|v_{r_0}) \quad = \quad \pi^0_{m,s} + \pi^0_{m,s'} \tag{12}$$

We collapse the sub-tree rooted at v_{r_m} so as to form a single leaf node and recursively work towards the sub-tree rooted at the child nodes of the root node v_{r_0}. Then, for the child node v_{r_1}, we have

$$\sum_{i \in ST_{r_1}} \sum_{t \in S} J(\phi_i = t|v_{r_0}) \quad = \quad \pi^0_{1,s} + \pi^0_{1,s'} \tag{13}$$

Just Accepted

Taking the sum over all subtrees rooted at the child nodes of v_{r_0}, we have

$$\sum_{i \neq r_0} \sum_{t \in S} J(\phi_i = t | v_{r_0}) = 1, \tag{14}$$

which completes our proof that the normalization property holds for the modified softmax function. □

In order to show that the modified softmax considers the connectivity pattern of a node, we prove next how the modified softmax score of the root node changes with respect to the shortest path length to any node as followed by Wang et al. [68] in unsigned networks.

Theorem 2. *In the modified softmax $J(\phi_i|v_j, \theta_J)$ decreases with increasing shortest path distance between v_i and v_j.*

Proof. Since BFS tree preserves the shortest path from the root node to the other nodes, so for a root node v_{r_0}, the probability $J(\phi_{r_n}|v_j, \theta_J)$ for an arbitrary node v_{r_n} that is n hops away is proportional to the product of n exponential terms (the sign-specific relevance probabilities) and thus, decreases exponentially. □

Therefore, on the basis of proof of Theorem 2, we conclude that the modified softmax can capture the connectivity pattern of a node as it always follows the shortest path distance. We follow Wang et al. [68] who discuss that capturing the connectivity pattern inherently captures the graph structure and therefore, conclude that SigGAN can capture the graph structure of signed networks. We next show that the modified softmax improves upon the computational efficiency in deriving $J(\phi_i|v_j)$.

Theorem 3. *For large networks with power-law degree distributions, to derive the probability $J(\phi_i|v_j)$, the average number of node representation updates is $O(d \ln|V|)$, where d is the average degree of the nodes.*

Proof. For an arbitrary node v_i, calculating $J(\phi_i|v_j)$ requires updating the representations of the nodes that lie in the path from v_j to v_i and also those that are directly connected to these. As derived in [22], for networks with power-law degree distribution, given as, $P(k) \sim k^{-\alpha}$, in the limit $|V| \rightarrow \infty$, the average path length between the nodes depends on the scaling exponent α. When $\alpha < 3$, the average path length saturates to $\frac{2}{3-\alpha} + \frac{1}{2}$, whereas for $\alpha = 3$, the average path length approximates to $\frac{\ln|V|}{\ln\ln|V|}$ and when $\alpha > 3$, then the same tends to $\ln|V|$. Thus, as the average path length for a large class of networks is maximally $O(\ln|V|)$, hence, the total number of nodes whose representations would be updated is $O(d \ln|V|)$ on an average. □

We discuss whether the modified softmax can maintain the structural balance property if it exists in the network next. For any node triplet, v_{r_i}, v_{r_j} and v_{r_k}, if edges exist between (v_{r_i}, v_{r_j}) and (v_{r_j}, v_{r_k}), then from Equation 8, we find that $J(\phi_{r_k} = s|v_{r_i})$ contains the expression $(\pi^i_{j,s}\pi^j_{k,s} + \pi^i_{j,s'}\pi^j_{k,s'})$ indicating that the corresponding probability of an edge between v_{r_i} and v_{r_k} being positive is high when either the probability of both edges (v_{r_i}, v_{r_j}) and (v_{r_j}, v_{r_k}) being positive is high or both of them being negative is high. This relation would be learnt from D which would return true with a very high probability when the sign of edge (v_{r_i}, v_{r_k}) is determined as positive by J. Similarly,for the negative edges between $(v_{r_i}$ and $v_{r_k})$. If either $\pi^i_{j,s}$ and $\pi^j_{k,s'}$ or $\pi^i_{j,s'}$ and $\pi^j_{k,s}$ are both high, then J would automatically increase the probability of $J(\phi_{r_k} = s'|v_{r_i})$. We investigate the performance of SigGAN next .

4 EXPERIMENTS

In this Section, we provide details of the datasets, the existing research works that we use for comparison and the details of SigGAN.

Just Accepted

Dataset	*Slashdot*	*Epinions*	*Reddit*	*Bitcoin*	*Wiki-RFA*
Nodes	7000	7000	6999	5877	7118
Edges	431098	734408	277050	21436	201386
Positive Edges	321030	642397	249580	18282	157206
Negative Edges	110068	92011	27470	3154	44180

Table 1. Dataset Details

4.1 Dataset Details

A brief overview of the five datasets, *Slashdot*, *Epinions*, *Bitcoin*, *Reddit* and *Wiki-RFA* is as follows :

- *Slashdot*: It represents the friendship or animosity relationship among users on *Slashdot* news website[1] [44].
- *Epinions*: It represents the trust or mistrust relationship among users on a consumer review site, *Epinions* [2] [44].
- *Reddit*: It represents the positive or negative sentiment between subreddits[3] [40].
- *Bitcoin*: It represents the trust or distrust relationship among users who trade on Bitcoin-OTC [4] [42] platform.
- *Wiki-RFA*: It represents the voting relationship among users on Wikipedia[5] [71].

The details of the datasets are in the Table 1 which indicates the high imbalance in number of positive and negative edges and the sparsity.

4.2 Baselines

We compare SigGAN with 7 state-of-the-art existing research works which generate node embedding for signed networks and then, utilize the generated node embedding for signed link prediction. We discuss these works briefly next.

(1) *DNE-SBP*: Shen et al. [56] proposed an semi-supervised stacked auto-encoder which considers structural balance theory to generate node embedding for signed networks.
(2) *SIDE*: Kim et al. [38] proposed a random walk based representation learning approach for signed directed networks that incorporates sign, direction, and proximity relationships of the nodes to generate low dimensional vector representation of each node.
(3) *SNE*: Yuan et al. [77] proposed a technique for signed network embedding that uses a log-bilinear model for generating node embedding.
(4) *SiGAT*: Huang et al. [30] proposed a graph attention based model that utilizes the motif based structure to generate node embedding for signed networks.
(5) *SNEA*: Li et al. [45] proposed a graph attention based model for signed networks that utilizes the neighbourhood of a node to generate the embedding.
(6) *HSNE*: Song et al. [58] proposed a hyperbolic representation learning to generate node embedding for signed networks.
(7) *SNRM*: Xu et al. [73] proposed a random walk based model that can capture the high order network structure and edge signs to generate node embedding for signed networks.

[1]https://snap.stanford.edu/data/soc-sign-Slashdot090221.html
[2]https://snap.stanford.edu/data/soc-sign-epinions.html
[3]https://snap.stanford.edu/data/soc-RedditHyperlinks.html
[4]https://snap.stanford.edu/data/soc-sign-bitcoin-otc.html
[5]https://snap.stanford.edu/data/wiki-RfA.html

Just Accepted

Dataset	*Slashdot*	*Epinions*	*Reddit*	*Bitcoin*	$Wiki - RFA$
L1	0.742	0.883	0.932	0.861	0.765
L2	0.760	0.878	0.942	**0.872**	0.766
Had	**0.793**	**0.893**	**0.963**	0.863	0.754
Avg	0.783	0.883	0.93	0.868	**0.775**
Concat	0.741	0.875	0.924	0.873	0.746

Table 2. Mean micro F1-score of the SigGAN for different edge vector representations

Dataset	*Slashdot*	*Epinions*	*Reddit*	*Bitcoin*	$Wiki - RFA$
SigGAN	**0.789**	**0.894**	**0.960**	**0.861**	0.753
$DNE - SBP$	0.779	0.840	0.940	0.820	0.711
SIDE	0.698	0.842	0.912	0.775	0.701
SNE	0.648	0.728	0.812	0.792	0.691
SiGAT	0.763	0.873	0.921	0.828	0.738
SNEA	0.761	0.856	0.932	0.836	0.747
HSNE	0.778	0.872	0.918	0.843	**0.762**
SNRM	0.770	0.871	0.92	0.801	0.735

Table 3. Mean micro F1- score of the SigGAN with the existing research works

For our experiments, we perform stochastic gradient descent to update parameters in SigGAN. We consider the learning rate as 0.001, batch size as 32, the number of epochs as 10 and the dimension of node embedding as 50. For each iteration, we set the number of positive samples as 20 and repeat this for 10 epochs for both generator and discriminator. We set these hyper-parameter values by cross-validation which we discuss in Sub-subsection 5.1.1. We outline the experiments and observations next.

5 RESULTS AND DISCUSSIONS

In this Section, we compare the performance of SigGAN with the baselines. As the number of negatively signed edges is very less compared to the number of positively signed edges (shown in Table 1), we consider *micro-F1* score which captures the effectiveness of an approach to predict both positive and negative sign. *Micro-F1* score is the harmonic mean of average *precision* and average *recall* such that average *precision* and average *recall* are calculated as the average of the precision and recall to predict positively and negatively signed edges correctly. Therefore, a higher *micro-F1* score implies that the approach can predict the sign of both types of edges with high effectiveness. We also evaluate the performance of SigGAN in handling sparse datasets and *structural balance theory* satisfaction.

5.1 Link Prediction

We compare the performance of SigGAN with the baselines to identify the sign of an edge. Existing research works on signed link prediction primarily generate the edge representation from the node vector representations obtained from the corresponding embedding proposed model and then, use these edge representations to train a logistic regression classifier [31, 38, 56, 58] by 5-fold cross-validation for signed link prediction. We found 5 possible approaches to generate the edge representation as followed in existing research works, such as, L1-norm, L2-norm, Hadamard product, Average and Concatenation. For each method for edge representation, we train

Just Accepted

Appr	Slashdot				Epinions				Wiki-RFA			
	80%	60%	40%	20%	80%	60%	40%	20%	80%	60%	40%	20%
SigGAN	0.726	**0.763**	**0.778**	**0.789**	**0.861**	**0.868**	**0.876**	**0.894**	.732	0.741	0.748	0.753
DNE − SBP	0.713	0.761	0.764	0.779	0.829	0.841	0.848	0.840	0.698	0.701	0.703	0.711
SIDE	0.601	0.643	0.662	0.698	0.817	0.824	0.836	0.842	0.674	0.687	0.690	0.701
SNE	0.586	0.602	0.610	0.648	0.709	0.716	0.720	0.728	0.659	0.673	0.676	0.691
SiGAT	0.722	0.736	0.749	0.763	0.843	0.849	0.861	0.873	0.74	0.75	0.73	0.738
SNEA	0.739	0.740	0.746	0.757	0.813	0.840	0.847	0.856	0.72	0.73	0.73	0.747
HSNE	**0.749**	0.757	0.760	0.767	0.848	0.854	0.868	0.872	**0.736**	**0.742**	**0.750**	**0.762**
SNRM	0.737	0.748	0.753	0.770	0.853	0.858	0.869	0.871	0.704	0.721	0.726	0.735

Table 4. Comparison results of mean micro F1 score of SigGAN with the baselines for different different levels of sparsity

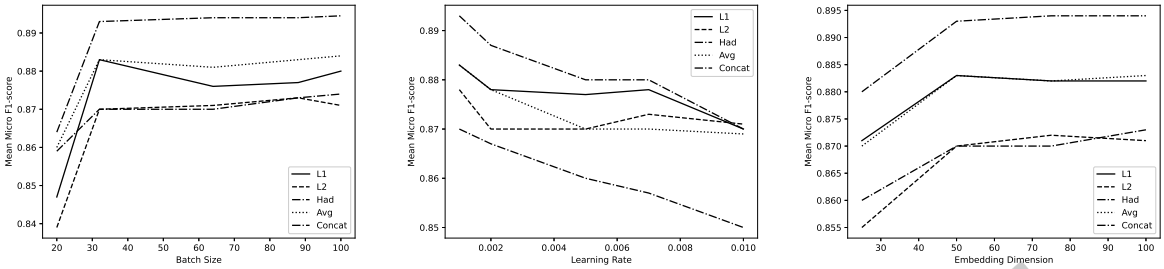
Appr	Bitcoin				Reddit			
	80%	60%	40%	20%	80%	60%	40%	20%
SigGAN	**0.830**	**0.837**	**0.848**	**0.861**	**0.938**	**0.947**	**0.953**	**0.960**
DNE − SBP	0.801	0.814	0.817	0.820	0.909	0.914	0.919	0.940
SIDE	0.736	0.741	0.753	0.775	0.881	0.892	0.906	0.912
SNE	0.760	0.775	0.790	0.792	0.785	0.791	0.802	0.812
SiGAT	0.801	0.812	0.820	0.828	0.897	0.903	0.914	0.921
SNEA	0.810	0.821	0.825	0.836	0.893	0.901	0.911	0.932
HSNE	0.821	0.829	0.830	0.843	0.879	0.886	0.903	0.918
SNRM	0.763	0.770	0.783	0.801	0.857	0.861	0.882	0.920

Table 5. Comparison results of mean micro F1 score of the SigGAN with the baselines for different levels of sparsity

Just Accepted

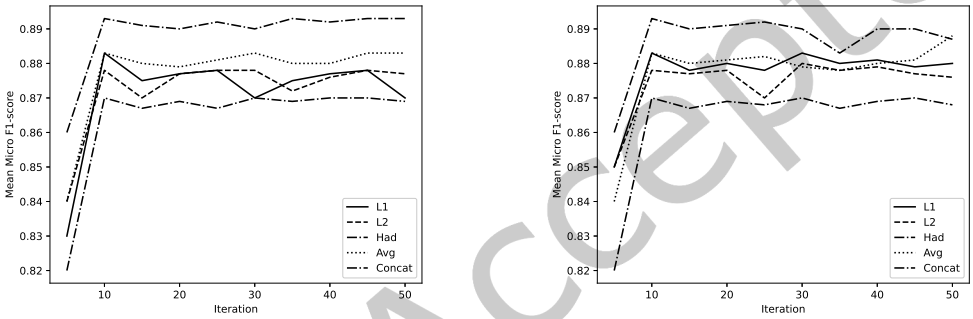
a *Logistic Regression* model on the training set and report the mean *micro F1-score* which is calculated as the average of *micro F1-score* over 5 independent runs. As there is a high imbalance in the number of negatively and positively connected edges, we calculate *micro F1-score* which can capture the performance of a model in both positive and negative sign prediction.

We show the mean micro F1-score for the 5 possible ways of edge representations for SigGAN in Table 2. We observe that the edge representation by Hadamard product performs the best for *Slashdot*, *Epinions* and *Reddit*, edge representation by Average for *Wiki-RFA* and edge representation by L2-norm for *Bitcoin* respectively. We also observe that the edge representation by Hadamard product for *Wiki-RFA* and *Bitcoin* is very near to the best performance with a difference of 0.02 and 0.01 respectively. Therefore, to maintain uniformity irrespective of the method for edge representations, we follow edge representation by Hadamard product for SigGAN. For the baselines, we follow the method described in the corresponding research work, such as, Hadamard product for *DNE-SBP*[56] and *SNE*[77], concatenation for *SIDE* [38], *SiGAT* [30], *SNEA* [45] and *SNRM* [73] and L2-Norm for *HSNE* [58]. Our observations from Table 3 indicates the SigGAN provides better performance than all the existing research works in all the datasets except *Wiki-RFA*. The mean micro F1-score of the SigGAN is around $2-18\%$ more than the existing research works and SigGAN performs the best for *Reddit* dataset with around 0.960 mean micro F1-score followed by 0.894 in *Epinions*. Among the baselines, the performance of *HSNE* [58] is the best and is also better than SigGAN by 1.18% for *Wiki-RFA*. We discuss the values of each hyper-parameter and its effect on the performance of SigGAN next.



(a) (b) (c)

Fig. 1. Comparing the effect of change in batch size, learning rate and embedding dimension on Mean Micro F1-score for *Epinions* in Figure 1a, Figure 1b and Figure 1c respectively.



(a) (b)

Fig. 2. Comparing the effect of change of number of epochs for Generator and Discriminator and number of epochs of SigGAN on Mean Micro F1-score for *Epinions* dataset in Figure 2a and Figure 2b respectively.

5.1.1 Hyperparameter Analysis. : We discuss next how we set the values of the different parameter or hyperparameter for SigGAN. For each parameter or hyperparameter, we vary its values and calculate the mean micro F1-score for 5 independent runs for each value. Therefore, we select that value for a parameter or hyperparameter which gives the best mean micro F1-score during training. We consider edge representations by all the 5 methods, such as, L1-norm, L2-norm, Hadamard product, Average and Concatenation. We show our observations for *Epinions* in Figure 1 and Figure 2 respectively. Based on our observations from Figure 1a, we find the value of batch-size of 32 provides the maximum mean micro F1-score for L1-norm, Average and Concatenation and 100 for L2-norm and Hadamard product. We also observe that the increase in results for L2-norm and Hadamard product when batch size is increased to 100 from 32 is very small and further, it might lead to overfitting if we increase the batch size to 100. Therefore, we consider 32 as the batch size for SigGAN. Similarly, we observe the learning rate as 0.001 provides the maximum mean micro F1-score for SigGAN as shown in Figure 1b. Additionally, to decide the embedding dimension, we vary the dimension from $25 - 100$ and calculate the result of SigGAN. Our observations as shown in Figure 1c indicate that increasing the dimension from 50 to 100 provides almost similar mean micro F1-score and dimension of size 25 has the least mean micro F1-score. As we want to have as low

Just Accepted

dimension as possible, we choose 50 as the node embedding dimension. We specify the number of epochs by which we train SigGAN as s and the number of epochs for which discriminator and generator runs for each value of s as t. Our observations from Figure 2a and Figure 2b indicate that SigGAN has the maximum mean micro F1-score irrespective of the edge representation function when s and t are both 10 and any further increase in the number of epochs has very less impact on the mean micro F1-score. We intuitively believe that both the design of specific Softmax for signed networks and simultaneous training by equal number of positive and negative edges leads to quicker convergence.

5.2 Handling Sparsity

We investigate the performance of SigGAN in handling sparse graphs. For our experiments, we induce sparsity in the original graph by randomly removing 20%, 40%, 60% and 80% of the edges from the original graph. For each of these sparse graphs, we generate the embedding of each node using SigGAN. Further, to create a graph with specific sparsity, say 20%, we repeat the procedure 5 times so that we have 5 different graphs with 20% sparsity. For each of these graphs, we then evaluate the performance of SigGAN in sign prediction by following the procedure discussed in Section 5.1. We repeat the same experiment for *Slashdot*, *Epinions*, *Wiki-RFA*, *Reddit* and *Bitcoin*. We calculate the mean *micro F1-score* as the average of *micro F1-score* over 5 different graphs with same sparsity and *micro F1-score* for a graph is calculated by 5-fold cross validation. We show our observations in Table 4 and Table 5 respectively. We observed that the SigGAN ensures better performance than the baselines in all the datasets except *Wiki-RFA*. Subsequently, we observe that SigGAN ensures almost consistent performance irrespective of the sparsity in the signed network, i.e., $0.726 - 0.789$ for *Slashdot*, $0.861 - 0.894$ for *Epinions*, $0.732 - 0.753$ for *Wiki-RFA*, $0.830 - 0.861$ for *Bitcoin* and $0.938 - 0.960$ for *Reddit*.

5.3 Case Study : Analysis of Embedding by SigGAN

We now study whether the node representation by SigGAN follows the extended structural balance property, i.e., positively connected nodes are closer in the representation space than the negatively connected nodes [33]. For our experiments, we follow the procedure adopted in existing research works [33, 38]. In order to measure this, we calculate the average positive edge distance, *APED* as the average distance between the vector representation of the positively connected nodes and *ANED* for negatively connected nodes, such as,

$$APED = \frac{(\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^{+} d_{ij})}{(\sum_{i=1}^{n} A_{ij}^{+})} \tag{15}$$

$$ANED = \frac{(\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^{-} d_{ij})}{(\sum_{i=1}^{n} A_{ij}^{-})} \tag{16}$$

where, A_{ij}^{+} is 1 if node i and node j are positively connected else 0, A_{ij}^{-} is is 1 if node i and node j are negatively connected else 0 and d_{ij} is the Euclidean distance between the vector representations of i and j. Therefore, for a network to follow *extended structural balance theory*, *ANED* should be greater than *APED*. We randomly select equal number of positively and negatively connected edges, i.e., 40% of the total negatively connected edges. We repeat the same experiment for all the datasets, i.e., *Slashdot*, *Epinions*, *Reddit*, *Bitcoin* and *Wiki-RFA*. The results are shown in Table 6 which indicates that *APED*, is around $1.5 - 4.0$ less than *ANED*, irrespective of the dataset. We observe that SigGAN has the best performance for *Epinions* and *Bitcoin* where the average distance between positively and negatively connected nodes are the highest. Our observations confirm that SigGAN can ensure that the positively connected nodes have a smaller distance than the negatively connected nodes. Thus, these results indicate that for signed networks, the SigGAN ensures positively connected node pairs are placed closer than the negatively connected node pairs and therefore, ensure *extended structural balance theory* property.

Just Accepted

Dataset	*Slashdot*	*Epinions*	*Reddit*	*Bitcoin*	$Wiki - RFA$
APED	73	54.5	10.90	45	10
ANED	74.50	58.56	12	48	11

Table 6. We show *APED* and *ANED* for *Slashdot*, *Epinions*, *Reddit*, *Bitcoin* and *Wiki-RFA*

5.4 Summary of Insights

We discuss SigGAN in comparison to GraphGAN [68] and its applicability to signed networks followed by an analysis of the performance of the SigGAN and its the shortcomings. We develop SigGAN based on GraphGAN [68] which utilizes the inherent characteristics of the graph structure in a GAN based framework to generate node embedding. Although GraphGAN [68] is very effective for link prediction in unsigned networks, it can not be directly applied to signed networks. Therefore, in this paper, we propose SigGAN which is not a direct extension of GraphGAN but explicitly considers the specific characteristics of signed networks, like *structural balance theory* and *high imbalance in number of positively and negatively connected edges*. We propose a modified softmax for signed networks which integrates *structural balance theory* with the existing properties from unsigned networks, like *normalization* and *graph structure awareness*. Therefore, by integration of these properties in the proposed modified softmax, we ensure that the generator is highly effective in generating negative samples for the discriminator. We show in Subsection 3.3 how we formulate modified softmax such that it considers the properties. We prove appropriate theorems in Subsection 3.3 to show that *normalization*, *graph structure awareness* and *computational efficiency*, are satisfied in the generator. We also show through intuition in Subsection 3.3 and through a case study in Subsection 5.3 that SigGAN satisfies *structural balance theory*. In order to handle *high imbalance in number of positive and negative edges* and high information content in the negative edges, we train the discriminator with an equal number of positive and negative edges. This further ensures that SigGAN can ensure prediction of negative edge effectively. Our observations in Subsection 5.1 and Subsection 5.2 shows that SigGAN has a higher mean micro F1-score than the baselines. As the micro F1-score considers the efficiency in predicting both the positive and the negative edges, Table 3 - 5 indicates the high performance of SigGAN in capturing both the positive and negative edges for original and sparse graphs respectively. Therefore, we believe the design of the modified softmax specific for signed networks for generator and simultaneous training by both positive and negative edges for discriminator are the main reasons behind success of SigGAN. While the generator is effective in generating negative samples irrespective of the sign of the edge considering the signed network characteristics, the discriminator learns to handle the imbalance in data.

5.4.1 Limitations of SigGAN. : Although SigGAN ensures high effectiveness in comparison with the existing baselines irrespective of the dataset and presence of sparsity, we observe few limitations in SigGAN which we discuss next. A critical analysis of SigGAN would be that we did not explore the applicability of different discriminator models which can improve the performance. We evaluate the execution time for SigGAN and the other baselines on a machine with an Intel Core $i7$ $4.2GHz$ CPU and 32 GB RAM. We observe during sparsity analysis that SigGAN is linearly scalable when the fraction of edges is increased gradually from 20% of the total edges to all the edges irrespective of the dataset. The time taken are 32, 40, 51, 60 and 72 minutes respectively for 20%, 40%, 60%, 80% and 100% of the edges for *Epinions*. Furthermore, on comparing the execution time of SigGAN with the baselines, we observe that the execution time of DNE-SBP is the best followed by SigGAN, for example, the execution time on *Epinions* was 50 minutes for DNE-SBP. We show our observations on *Epinions* as it is the dataset with the most number of edges (around 0.8 million edges). Therefore, although SigGAN has better performance than DNE-SBP, it requires more time to converge. We intuitively believe by exploring different

Just Accepted

discriminator models and by including structural role based information of the nodes in modified softmax, we can lower the execution time of SigGAN. We consider these as the future directions of the paper.

6 CONCLUSIONS

We propose a Generative Adversarial Network (GAN) based approach, SigGAN, for generating node representation for signed networks. SigGAN can handle the inherent challenges of signed networks, such as, the high imbalance in negative and positive edges and incorporates structural balance theory while ensuring high effectiveness in signed link prediction. Validation on 5 datasets, like *Slashdot*, *Epinions*, *Reddit*, *Bitcoin* and *Wiki-RFA* indicate the SigGAN can predict links with a high mean micro F1-score of $0.75 - 0.96$ which is higher than the existing state-of-the-art research works. As a future goal, we plan to extend SigGAN by exploring different discriminator functions and integration of role based information for node embedding. As most of the signed relationships formed in real life undergo change in polarity with respect to time, we intend to make SigGAN applicable for temporal signed networks .

REFERENCES

[1] Priyanka Agrawal, Vikas K Garg, and Ramasuri Narayanam. 2013. Link Label Prediction in Signed Social Networks.. In *IJCAI*. 2591–2597.

[2] Ghazaleh Beigi, Jiliang Tang, and Huan Liu. 2016. Signed Link Analysis in Social Media Networks.. In *ICWSM*. 539–542.

[3] Ayan Kumar Bhowmick, Koushik Meneni, and Bivas Mitra. 2019. On the Network Embedding in Sparse Signed Networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 94–106.

[4] Dorwin Cartwright and Frank Harary. 1956. Structural balance: a generalization of Heider's theory. *Psychological review* 63, 5 (1956), 277.

[5] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jaeho Choi. 2019. Rating augmentation with generative adversarial networks towards accurate collaborative filtering. In *The World Wide Web Conference*. 2616–2622.

[6] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. Cfgan: A generic collaborative filtering framework based on generative adversarial networks. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 137–146.

[7] Roshni Chakraborty, Srishti Bhandari, Nilotpal Chakraborty, and Ritwika Das. 2020. Eve2Sign: Creating Signed Networks of News Events.. In *Text2Story@ ECIR*. 79–87.

[8] Roshni Chakraborty, Maitry Bhavsar, Sourav Kumar Dandapat, and Joydeep Chandra. 2019. Tweet summarization of news articles: An objective ordering-based perspective. *IEEE Transactions on Computational Social Systems* 6, 4 (2019), 761–777.

[9] Roshni Chakraborty, Maitry Bhavsar, Sourav Kumar Dandapat, and Joydeep Chandra. 2022. Detecting Stance in Tweets: A Signed Network based Approach. *arXiv preprint arXiv:2201.07472* (2022).

[10] Roshni Chakraborty and Nilotpal Chakraborty. 2020. Are we friends or enemies? Let's ask thy neighbour!. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*. 61–62.

[11] Xiao Chen, Jing-Feng Guo, Xiao Pan, and Chunying Zhang. 2017. Link prediction in signed networks based on connection degree. *Journal of Ambient Intelligence and Humanized Computing* (2017), 1–11.

[12] Xiao Chen, Jing-Feng Guo, Xiao Pan, and Chunying Zhang. 2019. Link prediction in signed networks based on connection degree. *Journal of Ambient Intelligence and Humanized Computing* 10, 5 (2019), 1747–1757.

[13] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. 2018. Bridge: Enhanced Signed Directed Network Embedding. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 773–782.

[14] Kewei Cheng, Jundong Li, and Huan Liu. 2017. Unsupervised feature selection in signed social networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 777–786.

[15] Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Inderjit S Dhillon, and Ambuj Tewari. 2014. Prediction and clustering in signed networks: a local to global perspective. *The Journal of Machine Learning Research* 15, 1 (2014), 1177–1213.

[16] Kai-Yang Chiang, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S Dhillon. 2011. Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 1157–1162.

[17] Debasis Das. 2018. Positive and Negative Link Prediction Algorithm Based on Sentiment Analysis in Large Social Networks. *Wireless Personal Communications* 102, 3 (2018), 2183–2198.

[18] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751* (2015).

[19] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 929–934.

Just Accepted

[20] Tyler Derr, Chenxing Wang, Suhang Wang, and Jiliang Tang. 2018. Relevance Measurements in Online Signed Social Networks. In *Proceedings of the 14th International Workshop on Mining and Learning with Graphs (MLG)*.

[21] Patrick Doreian. 1989. Network autocorrelation models: Problems and prospects. *Spatial statistics: Past, present, future* (1989), 369–89.

[22] Agata Fronczak, Piotr Fronczak, and Janusz A. Hołyst. 2004. Average path length in random networks. *Phys. Rev. E* 70 (Nov 2004), 056110. Issue 5.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[24] Ian J. Goodfellow. 2017. NIPS 2016 Tutorial: Generative Adversarial Networks. *CoRR* abs/1701.00160 (2017). arXiv:1701.00160 http://arxiv.org/abs/1701.00160

[25] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[26] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*. ACM, 403–412.

[27] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[28] Fritz Heider. 1958. The psychology of interpersonal relations Wiley. *New York* (1958).

[29] Baofang Hu, Hong Wang, Xiaomei Yu, Weihua Yuan, and Tianwen He. 2017. Sparse network embedding for community detection and sign prediction in signed social networks. *Journal of Ambient Intelligence and Humanized Computing* (2017), 1–12.

[30] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. 2019. Signed Graph Attention Networks. *arXiv preprint arXiv:1906.10958* (2019).

[31] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. 2021. SDGNN: Learning Node Representation for Signed Directed Networks. *arXiv preprint arXiv:2101.02390* (2021).

[32] Norman P Hummon and Patrick Doreian. 2003. Some dynamics of social balance processes: bringing Heider back into balance theory. *Social Networks* 25, 1 (2003), 17–49.

[33] Mohammad Raihanul Islam, B Aditya Prakash, and Naren Ramakrishnan. 2018. SIGNet: scalable embeddings for signed networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 157–169.

[34] Amin Javari, Tyler Derr, Pouya Esmailian, Jiliang Tang, and Kevin Chen-Chuan Chang. 2020. Rose: Role-based signed network embedding. In *Proceedings of The Web Conference 2020*. 2782–2788.

[35] Amin Javari and Mahdi Jalili. 2014. Cluster-based collaborative filtering for sign prediction in social networks with positive and negative links. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 2 (2014), 24.

[36] Amin Javari, HongXiang Qiu, Elham Barzegaran, Mahdi Jalili, and Kevin Chen-Chuan Chang. 2017. Statistical Link Label Modeling for Sign Prediction: Smoothing Sparsity by Joining Local and Global Information. In *Data Mining (ICDM), 2017 IEEE International Conference on*. IEEE, 1039–1044.

[37] Abtin Khodadadi and Mahdi Jalili. 2017. Sign prediction in social networks based on tendency rate of equivalent micro-structures. *Neurocomputing* 257 (2017), 175–184.

[38] Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang. 2018. Side: representation learning in signed directed networks. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 509–518.

[39] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[40] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2018. Community interaction and conflict on the web. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 933–943.

[41] Srijan Kumar, Francesca Spezzano, and VS Subrahmanian. 2014. Accurately detecting trolls in slashdot zoo via decluttering. In *Proceedings of the 2014 IEEE/ACM international conference on advances in social networks analysis and mining*. IEEE Press, 188–195.

[42] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. 2016. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 221–230.

[43] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*. ACM, 641–650.

[44] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1361–1370.

[45] Yu Li, Yuan Tian, Jiawei Zhang, and Yi Chang. 2020. Learning signed network embedding via graph attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4772–4779.

[46] Wanyu Lin and Baochun Li. 2022. Status-Aware Signed Heterogeneous Network Embedding With Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[47] Chunyu Lu, Pengfei Jiao, Hongtao Liu, Yaping Wang, Hongyan Xu, and Wenjun Wang. 2019. SSNE: Status Signed Network Embedding. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 81–93.

Just Accepted

[48] Can Lu, Jeffrey Xu Yu, Rong-Hua Li, and Hao Wei. 2016. Exploring hierarchies in online social networks. *IEEE Transactions on Knowledge and Data Engineering* 28, 8 (2016), 2086–2100.
[49] Shuqi Lu, Zhicheng Dou, Xu Jun, Jian-Yun Nie, and Ji-Rong Wen. 2019. Psgan: A minimax game for personalized search with limited and noisy click data. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 555–564.
[50] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 17–24.
[51] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[52] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
[53] Amin Salehi and Hasan Davulcu. 2018. Detecting Antagonistic and Allied Communities on Social Media. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 99–106.
[54] Moshen Shahriari and Mahdi Jalili. 2014. Ranking nodes in signed social networks. *Social Network Analysis and Mining* 4, 1 (2014), 172.
[55] Cosma Rohilla Shalizi and Andrew C Thomas. 2011. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research* 40, 2 (2011), 211–239.
[56] Xiao Shen and Fu-Lai Chung. 2018. Deep Network Embedding for Graph Representation Learning in Signed Networks. *IEEE transactions on cybernetics* (2018).
[57] Ranveer Singh and Bibhas Adhikari. 2017. Measuring the balance of signed networks and its application to sign prediction. *Journal of Statistical Mechanics: Theory and Experiment* 2017, 6 (2017), 063302.
[58] Wenzhuo Song, Hongxu Chen, Xueyan Liu, Hongzhe Jiang, and Shengsheng Wang. 2021. Hyperbolic node embedding for signed networks. *Neurocomputing* 421 (2021), 329–339.
[59] Jiliang Tang, Charu Aggarwal, and Huan Liu. 2016. Node classification in signed social networks. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 54–62.
[60] Jiliang Tang, Shiyu Chang, Charu Aggarwal, and Huan Liu. 2015. Negative link prediction in social media. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 87–96.
[61] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. 2016. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)* 49, 3 (2016), 1–37.
[62] Jiliang Tang, Xia Hu, and Huan Liu. 2014. Is Distrust the Negation of Trust?: The Value of Distrust in Social Media. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media* (Santiago, Chile) *(HT '14)*. ACM, New York, NY, USA, 148–157.
[63] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.
[64] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. 2014. Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.
[65] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *stat* 1050 (2017), 20.
[66] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1225–1234.
[67] Hongwei Wang, Jialin Wang, Jia Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Learning graph representation with generative adversarial nets. *IEEE Transactions on Knowledge and Data Engineering* 33, 8 (2019), 3090–3103.
[68] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Graphgan: Graph representation learning with generative adversarial nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[69] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 592–600.
[70] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. 2017. Signed network embedding in social media. In *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 327–335.
[71] Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. 2014. Exploiting social network structure for person-to-person sentiment analysis. *arXiv preprint arXiv:1409.2450* (2014).
[72] Zhaoming Wu, Charu C Aggarwal, and Jimeng Sun. 2016. The troll-trust model for ranking in signed networks. In *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*. 447–456.
[73] Pinghua Xu, Wenbin Hu, Jia Wu, and Weiwei Liu. 2022. Signed network representation with novel node proximity evaluation. *Neural Networks* (2022).
[74] Dengcheng Yan, Youwen Zhang, Wei Li, and Yiwen Zhang. 2021. MUSE: Multi-faceted Attention for Signed Network Embedding. *arXiv preprint arXiv:2104.14449* (2021).

Just Accepted

[75] Bo Yang, William Cheung, and Jiming Liu. 2007. Community mining from signed social networks. *IEEE transactions on knowledge and data engineering* 19, 10 (2007), 1333–1348.
[76] Bo Yang, Xueyan Liu, Yang Li, and Xuehua Zhao. 2017. Stochastic blockmodeling and variational Bayes learning for signed network analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 9 (2017), 2026–2039.
[77] Shuhan Yuan, Xintao Wu, and Yang Xiang. 2017. SNE: signed network embedding. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 183–195.
[78] Weiwei Yuan, Kangya He, Donghai Guan, and Guangjie Han. 2017. Edge-dual graph preserving sign prediction for signed social networks. *IEEE Access* 5 (2017), 19383–19392.
[79] Weiwei Yuan, Chenliang Li, Guangjie Han, Donghai Guan, Li Zhou, and Kangya He. 2017. Negative sign prediction for signed social networks. *Future Generation Computer Systems* (2017).
[80] Weinan Zhang. 2018. Generative Adversarial Nets for Information Retrieval: Fundamentals and Advances. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 1375–1378.

Just Accepted