Contrastive Learning for Signed Bipartite Graphs

Zeyu Zhang
University of Electronic Science and Technology of China
Chengdu, China
The University of Auckland
Auckland, New Zealand
zzha669@aucklanduni.ac.nz

Jiamou Liu*
The University of Auckland
Auckland, New Zealand
jiamou.liu@auckland.ac.nz

Kaiqi Zhao
The University of Auckland
Auckland, New Zealand
kaiqi.zhao@auckland.ac.nz

Song Yang
The University of Auckland
Auckland, New Zealand
syan382@aucklanduni.ac.nz

Xianda Zheng
The University of Auckland
Auckland, New Zealand
xzhe162@aucklanduni.ac.nz

Yifei Wang
The University of Auckland
Auckland, New Zealand
wany107@aucklanduni.ac.nz

ABSTRACT

This paper is the first to use contrastive learning to improve the robustness of graph representation learning for signed bipartite graphs, which are commonly found in social networks, recommender systems, and paper review platforms. Existing contrastive learning methods for signed graphs cannot capture implicit relations between nodes of the same type in signed bipartite graphs, which have two types of nodes and edges only connect nodes of different types. We propose a Signed Bipartite Graph Contrastive Learning (SBGCL) method to learn robust node representation while retaining the implicit relations between nodes of the same type. SBGCL augments a signed bipartite graph with a novel two-level graph augmentation method. At the top level, we maintain two perspectives of the signed bipartite graph, one presents the original interactions between nodes of different types, and the other presents the implicit relations between nodes of the same type. At the bottom level, we employ stochastic perturbation strategies to create two perturbed graphs in each perspective. Then, we construct positive and negative samples from the perturbed graphs and design a multi-perspective contrastive loss to unify the node presentations learned from the two perspectives. Results show proposed model is effective over state-of-the-art methods on real-world datasets. **Implementation available in PyTorch**[1].

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**; **Social networks**.

*Corresponding Author
[1]https://github.com/Alex-Zeyu/SBGCL

SIGIR '23, July 23–27, 2023, Taipei, Taiwan.

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591655

KEYWORDS

Contrastive learning, graph augmentation, signed bipartite graph, graph neural networks

ACM Reference Format:
Zeyu Zhang, Jiamou Liu, Kaiqi Zhao, Song Yang, Xianda Zheng, and Yifei Wang. 2023. Contrastive Learning for Signed Bipartite Graphs. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591655

1 INTRODUCTION

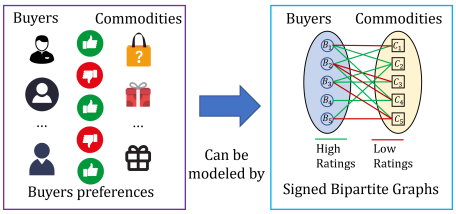

Figure 1: A scenario for the signed bipartite graph.

A *signed bipartite graph* is comprised of two disjoint sets of nodes and a number of positive or negative edges connecting nodes from different node sets. Signed bipartite graphs widely exist in social networks, recommender systems, paper review systems, etc. In social networks, users can place positive or negative opinions on a post. In recommender systems, users can rate positively (e.g., high ratings) or negatively (e.g., low ratings) on an item as illustrated in Figure 1. Analysis of the interactions between the two sets of nodes in signed bipartite graphs is essential for applications such as recommendations and link prediction.

Representation learning is one of the key techniques for graph data analysis. Although some representation learning methods for signed graphs have been proposed in recent years [4, 6, 19, 37], they work on unipartite graphs, and they are sensitive to noisy interactions. In real-world applications, noisy interactions commonly exist. For instance, a user may "unlike" an item by mistake. The noisy interactions can mislead the signed graph representation learning methods through the message-passing process, and thus degrade their performance in downstream applications.

Contrastive learning has achieved great success in learning robust graph representations that are invariant to small perturbations. In general, contrastive graph representation learning methods apply graph augmentation techniques (e.g., randomly adding or removing edges) to create several augmented graphs, and obtain node representations by contrasting the representations of the same node (i.e., positive samples) and those of different nodes (i.e., negative samples) in different augmented graphs. In this way, the learned representation of each node is consistent across the augmented graphs. GCA [39] proposes an adaptive augmentation method that incorporates various priors for topological and semantic aspects of the graph. A recent study proposes SGCL [24], a contrastive learning method for robust signed graph representation learning. SGCL employs an additional contrastive loss to make two nodes connected with a positive link to be close, and two nodes connected with a negative link to be distant, in the representation space. Although contrastive learning techniques have been introduced to graphs and signed graphs, how to design contrastive learning methods for signed bipartite graphs remains an open problem.

The main challenge in designing effective contrastive learning methods for signed bipartite graphs is to retain the *implicit relations* between two nodes in the same set. For example, in Fig. 1, two buyers B_3 and B_5 are considered to have similar interests because they both rate positively on C_2 & negatively on C_5. Such implicit relations have proved to be effective in link sign prediction [10]. Since existing contrastive learning methods only use *explicit relations*, i.e., links between nodes from different sets in signed bipartite graphs, they are incapable of modeling the implicit relations between nodes in the same set. In Fig. 1, existing contrastive learning methods for signed graphs will make the representation of C_5 distant from those of B_3 and B_5 because they are connected with negative edges. However, B_3 and B_5 are not necessarily close in the representation space, even though they have similar rating behaviors.

To overcome the challenge above, we propose a novel Signed Bipartite Graph Contrastive Learning method (**SBGCL**) that obtains robust node representations from two perspectives of a signed bipartite graph, i.e., *inter-set perspective* and *intra-set perspective*. The two perspectives present the explicit and implicit relations between nodes, respectively. Specifically, we propose a two-level graph augmentation method that creates the inter-set and intra-set perspectives at the top level and two randomly perturbed graphs for each perspective at the bottom level. To create the intra-set perspective, we connect two nodes in the same set according to their agreement/disagreement with their neighbors (in the other set), motivated by balance theory [9]. Then, we design a multi-perspective contrastive loss, which consists of two losse: (1) *Perspective-specific contrastive loss* to capture the explicit and implicit relations in the two graph perspectives; and (2) *Cross-perspective contrastive loss* to obtain node representations that are consistent with those obtained from the two perspectives. In this way, the learned node representations are not only invariant to small perturbations but also capture the implicit relations between nodes in the same set. Further, we provide a theoretical analysis of the impact of different graph perturbations (e.g., adding/removing edges, flipping the sign) on the learned representations.

The main contributions of this paper are summarized as follows:

- This work is the first to introduce contrastive learning for robust signed bipartite graph representation learning.
- We propose a two-level data augmentation method to capture explicit and implicit relations between nodes in signed bipartite graphs and conduct a theoretical analysis of different graph perturbations on the representation learning of graphs.
- We design a multi-perspective contrastive loss to obtain perturbation-invariant node representations that encode the implicit relations between nodes in the same set.
- We conduct experiments on four real-world signed bipartite networks, including product review, and peer review. Experimental results demonstrate the effectiveness of our proposed model in link sign prediction.

2 RELATED WORK

Signed graph representation. Signed graphs have been widely-studied due to the rapid development of social networks and recommender systems. Various signed graph analysis tasks have been explored, e.g., node classification [27], node ranking [14], link sign prediction, community detection [1], and visualization [30]. Graph representation learning is the key technique for most graph analysis tasks. Early graph representation learning methods for signed graphs are based on random walk [15, 16, 37] and matrix factorization [3, 18]. Recent years have seen deep learning been employed in signed representation learning. First, SiNE [30] extracts structural information from triangle motifs and designs an objective function based on balance theory [9]. Then, SGCN [6] becomes the first signed graph neural network model extending GCN [17] and employs balance theory to determine positive and negative relationships between nodes in multi-hop neighborhoods. Similarly, other models such as SiGAT [11], SNEA [19], and SDGNN [12] learn signed graph representations using graph attention nets [28]. Yet, these methods rely on the message-passing mechanism, which is sensitive to noisy interactions between nodes. A recent study [24] proposes a contrastive learning method for robust signed graph representation learning. Still, most signed graph representation models are designed for unipartite graphs. Although there have been many studies on unsigned bipartite graph embedding [2, 13, 25, 34], this paper only focuses on signed graph embedding methods. It is non-trivial to extend the existing signed graph representation learning methods to signed bipartite graphs. This is because existing methods may introduce unbalanced circles to signed bipartite graphs, and we theoretically prove that the unbalanced circles prohibit the message-passing mechanism to learn proper node presentations.

Graph contrastive learning. Contrastive learning has recently received significant interest due to its success in self-supervised representation learning in the computer vision domain. Unlike supervised methods where a human annotation is needed for every input sample, contrastive learning is a self-supervised learning method by contrasting positive and negative samples. For image data, negative samples can be generated using a multi-stage augmentation pipeline, consisting of color jitter, random flip, cropping, resizing, rotation, color distortion, etc. Inspired by the success of contrastive learning in images, researchers manage to extend contrastive learning to graph representation learning in recent years.

Graph contrastive representation learning aims to learn node representations invariant to small perturbations [23, 32, 39]. For example, DGI [29] incorporates graph neural networks and contrastive learning and generates node representations by maximizing mutual information between global graph representations and local node representations. GraphCL [36] develops a graph contrastive learning framework that leverages various types of graph augmentations to capture the invariant node representations. GCA [39] generates graph views by two graph augmentation methods and maximizes the agreement of node representations in different views. Most graph contrastive learning models are designed for unsigned graphs. Shu et al. [24] propose a graph contrastive learning method called SGCL for signed unipartite graphs. However, SGCL cannot learn proper node representations when applied to signed bipartite graphs because it may create unbalanced circles, as discussed above.

3 PROBLEM FORMULATION

A *signed bipartite graph* is $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$, where the two sets of nodes $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ & $\mathcal{V} = \{v_1, v_2, \ldots, v_{|V|}\}$ are disjoint, $\mathcal{E}^+ \subseteq \mathcal{U} \times \mathcal{V}$ and $\mathcal{E}^- \subseteq \mathcal{U} \times \mathcal{V}$ are positive & negative edge sets, resp.. Note that $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$ and the two endpoints of any edge in $\mathcal{E}^+ \cup \mathcal{E}^-$ must come from different sets $\mathcal{U}$ and $\mathcal{V}$. For example in Figure 1, the set $\mathcal{U}$ could be the set of users in an e-commerce platform, and the set $\mathcal{V}$ could be the set of products. A positive edge $(u, v) \in \mathcal{E}^+$ represents that the user u rates favourably the product v, and a negative edge $(u, v) \in \mathcal{E}^-$ represents that u rates v unfavourably. In this paper, we ignore the directions of the edges and treat the graph $\mathcal{G}$ as an undirected graph.

Given $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$, the goal is to learn a function f to map nodes $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$ to low-dimensional embeddings $z_{u_i} \in \mathbb{R}^d$ and $z_{v_j} \in \mathbb{R}^d$, where d is the dimension of node embeddings, so that the embeddings are useful in downstream tasks such as *link sign prediction* [6].

4 PROPOSED METHOD

In this section, we present a new Signed Bipartite Graph Contrastive Learning (**SBGCL**) model, aiming to obtain node representations that are invariant to small perturbations, and retain the implicit relations between nodes in the same set. Figure 2 shows the overall architecture. SBGCL consists of two novel components, i.e., (1) **two-level graph augmentation**; and (2) **multi-perspective contrastive learning**. To be specific, we first propose a two-level graph augmentation scheme to create two different perspectives of the original signed bipartite graph, namely *inter-set perspective* and *intra-set perspective*, to present the explicit and implicit relations of the graph, respectively. In each graph perspective, we employ stochastic graph augmentation to generate two augmented graphs, which are then passed to the multi-perspective contrastive learning component. In the multi-perspective contrastive learning component, we first split each augmented graph into a *positive graph* and a *negative graph* and apply graph neural networks (GNNs) to learn node representations. Then, we design a *multi-perspective contrastive loss* that enforces the representation of the same node to be consistent in different augmented graphs, while retaining both explicit and implicit relations. In addition, we analyze some natural graph augmentation methods from a theoretical point of view and conclude that those stochastic augmentation methods that can reduce unbalanced cycles are more effective.

4.1 Two-level Graph Augmentation

Graph augmentation plays an important role in graph contrastive learning, which aims to reduce the harm of interaction noise to models. In previous works [24, 38, 39], stochastic graph augmentation schemes are used to generate variants of the original graph by perturbing the graph structure and attributes. In essence, contrastive learning methods seek to learn representations that are invariant to perturbation introduced by the augmentation schemes [31]. When it comes to signed bipartite graphs, the situation is more complex, and there are two problems to be solved. **First,** the relations between nodes from the same set are not present in a signed bipartite graph, and GNNs cannot utilize such implicit relations. **Second,** there is a lack of theoretical analysis of stochastic augmentation methods (e.g., randomly adding and removing edges, flipping edge signs) adopted by existing graph augmentation schemes. It is unclear how different perturbation influences the learned representations. **To solve the first problem**, we propose a two-level graph augmentation scheme. The new graph augmentation scheme maintains two perspectives of the graph, namely *inter-set perspective* and *intra-set perspective*, respectively, at its top level. The inter-set perspective contains the original edges of the graph to capture the relations between nodes from different sets. The intra-set perspective concentrates on the implicit relations between nodes in the same set. Specifically, we create the intra-set perspective by placing positive and negative edges between nodes in the same set (Section 4.1.1). At the bottom level, we apply stochastic perturbation methods to generate two augmented graphs for each perspective (Section 4.1.2). **To solve the second problem**, we provide a theoretical analysis of three perturbation methods, including adding edges, removing edges, and flipping the edge sign (Section 4.1.3). Our theoretical analysis is based on the proof that current GNN encoders cannot learn proper representations for nodes from unbalanced circles. Therefore, the stochastic graph augmentation schemes that can reduce the number of unbalanced circles are more effective.

4.1.1 Intra-set Perspective. Relations between nodes are essential for downstream tasks such as link sign prediction. Since nodes in the same set $\mathcal{U}$ or $\mathcal{V}$ are not explicitly connected, the message-passing mechanism of existing GNNs cannot capture their implicit relations. As such, we propose an intra-set perspective of the original graph that showcase solely the implicit relations between nodes in the same set. The key question here is, *how to determine the sign of a link established between two nodes in the same set?* Intuitively, if two nodes agree with each other, i.e., share similar positive and negative neighbors, they should be connected positively. Otherwise, they should be connected negatively. For example in Fig. 3, u_1 & u_2 should be positively connected when they connect to node v_1 with the same sign (i.e., $u_1 \rightarrow^+ v_1, u_2 \rightarrow^+ v_1$ or $u_1 \rightarrow^- v_1, u_2 \rightarrow^- v_1$). When u_1 and u_2 connect v_1 with different signs (e.g., $u_1 \rightarrow^+ v_1, u_2 \rightarrow^- v_1$), they should be connected negatively. Our intuition aligns with *balance theory* [9]: Call the three triangles in Fig. 3 *balanced*. Balance theory says that human societies tend to avoid conflictual relations and form balanced triangles.

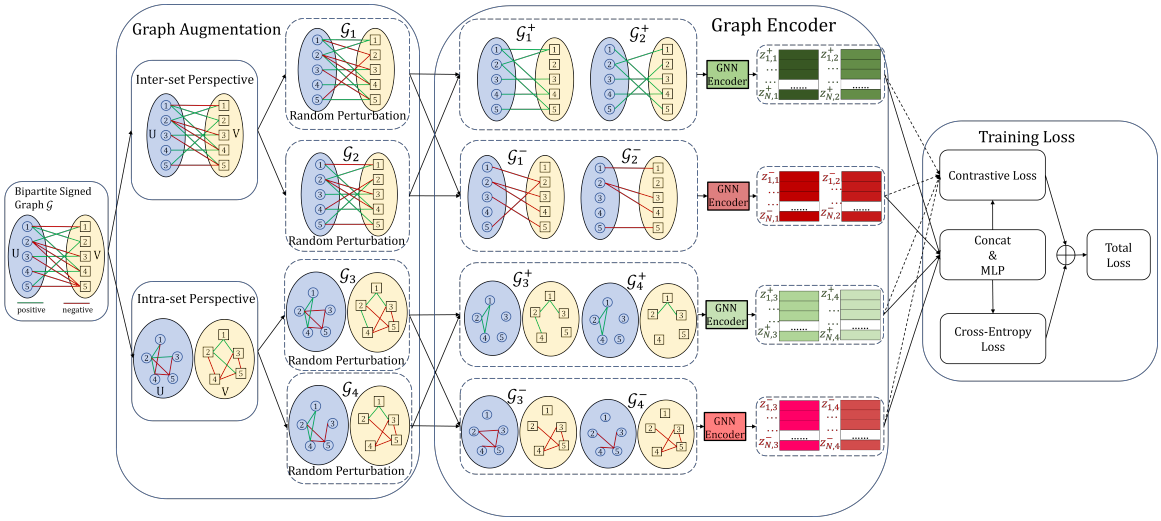

Figure 2: The overall architecture of SBGCL

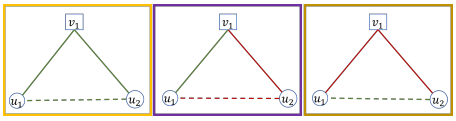

Figure 3: The model creates edges between nodes in the same set, where green lines refer to the positive edges and red lines refer to negative edges. Dotted lines refer to the created edges

Based on the intuition above, we connect two nodes in the same set $\mathcal{U}$ or $\mathcal{V}$ with the sign that favors producing balanced triangles. Since there could be multiple common neighbors of these two nodes, inconsistency may occur. For example, although nodes u_1 and u_2 connect to v_1 with the same sign, they may connect to other nodes, e.g., v_2, with different signs ($u_1 \rightarrow^+ v_1, u_2 \rightarrow^+ v_1$ or $u_1 \rightarrow^+ v_2$, $u_2 \rightarrow^- v_2$). In this paper, we determine the sign by *majority vote*: Say two nodes u_i, u_j in the same set are to be connected, each of their common neighbors produces a candidate sign (either positive or negative) that forms a balance triangle. For $u_i \in \mathcal{U}$ ($v_i \in \mathcal{V}$), let $\mathcal{N}^+_{\mathcal{V}}(u_i)$ & $\mathcal{N}^-_{\mathcal{V}}(u_i)$ ($\mathcal{N}^+_{\mathcal{U}}(v_i)$ & $\mathcal{N}^-_{\mathcal{U}}(v_i)$) denote the set of neighbors in $\mathcal{V}$ (in $\mathcal{U}$) with positive and negative edges to $u_i \in \mathcal{U}$ ($v_i \in \mathcal{V}$), respectively. Compute the difference between the number of positive candidates and the number of negative candidates as:

$$\mathcal{E}_{ij} = sgn\Big(|\mathcal{N}^+_{\mathcal{V}}(u_i) \cap \mathcal{N}^+_{\mathcal{V}}(u_j)| + |\mathcal{N}^-_{\mathcal{V}}(u_i) \cap \mathcal{N}^-_{\mathcal{V}}(u_j)| - |\mathcal{N}^+_{\mathcal{V}}(u_i) \cap \mathcal{N}^-_{\mathcal{V}}(u_j)| - |\mathcal{N}^-_{\mathcal{V}}(u_i) \cap \mathcal{N}^+_{\mathcal{V}}(u_j)|\Big) \tag{1}$$

where $sgn(\cdot)$ is the sign function. The notation $|\cdot|$ refers to the cardinality of a set. Nodes u_i and u_j are connected by a positive (negative) edge when $\mathcal{E}_{ij} > 0$ ($\mathcal{E}_{ij} < 0$). When $\mathcal{E}_{ij} = 0$, no edge is added between nodes u_i and u_j. Likewise, we can construct the sign between nodes of the other node set $\mathcal{V}$.

4.1.2 Stochastic Perturbation. Stochastic perturbation is used to enhance the robustness and generalization capability of the learned node representations. Concretely, through perturbing the structure, we make the learned representations invariant to perturbations by contrasting the representations obtained from different perturbed graphs [35]. In this work, we adopt and analyze three graph perturbation methods:

Randomly add or remove edges. Given a signed (bipartite) graph, we perturb the connectivity of the graph by randomly adding or removing edges. Each non-existing edge is added with probability p. Similar, each edge is discarded with probability p. For each edge to be added, we sample its sign uniformly.

Randomly flip edge signs. Different from the connectivity perturbation, an exclusive augmentation for signed graphs is randomly flipping edge signs. Particularly, a positive edge is transformed into a negative one with a probability p while a negative edge is transformed into a positive one with the same probability.

With any of the above stochastic perturbation methods, the two-level graph augmentation can produce several augmented graphs for each perspective. In this paper, we create two augmented graphs for each perspective, resulting in four augmented graphs, denoted as $\mathcal{G}_m = (\mathcal{U}, \mathcal{V}, \mathcal{E}^+_m, \mathcal{E}^-_m)$, where $m \in \{1, 2, 3, 4\}$.

Although data augmentation methods based on stochastic perturbation have been widely used in many graph contrastive learning methods [24, 38], it is unclear which perturbation method is more effective for signed bipartite graph representation learning. Next, we proceed to give a theoretical analysis of the effectiveness of the above three stochastic graph perturbation methods in the context of our two-level graph augmentation scheme.

4.1.3 Theoretical Analysis. Once the augmented graphs $\mathcal{G}_m$ are obtained, we apply GNNs to learn the node representations on each augmented graph. We prove that current GNN encoders cannot learn *proper representations* for nodes from unbalanced circles in the augmented signed graphs, i.e., $\mathcal{G}_m$. Thus, graph perturbation methods that can reduce the number of unbalanced circles are more effective than those that cannot. We do not strictly distinguish the node type in the theoretical analysis, because the message-passing mechanism of GNNs does not distinguish nodes of different types.

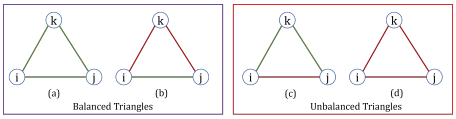

Figure 4: Four isomorphism types of undirected triangles. Green and red lines represent + and - edges, resp.

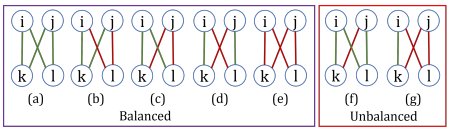

Figure 5: Seven isomorphism types of undirected Signed Butterfly. Green and red lines represent + and - edges, resp.

We start with the definition of proper representations of nodes in signed graphs and balanced/unbalanced circles.

Definition 4.1 (Proper node representations). Given a (augmented) signed graph $\mathcal{G}_m = (\mathcal{U}, \mathcal{V}, \mathcal{E}_m^+, \mathcal{E}_m^-)$, a GNN model $f_\theta : \mathcal{U} \cup \mathcal{V} \to H$ and any non-negative distance metric $dist : H \times H \to \mathbb{N}$, we call $h_{v_i} = f_\theta(v_i)$ a *proper representation* of any node $v_i \in \mathcal{U} \cup \mathcal{V}$ if the following conditions hold:

(a) There exists $\epsilon > 0$ such that for any $v_j \in \mathcal{N}_m^-(v_i)$ and $h_{v_j} = f_\theta(v_j)$, $dist(h_{v_i}, h_{v_j}) > \epsilon$;
(b) For any $v_j \in \mathcal{N}_m^+(v_i)$, $v_k \in \mathcal{N}_m^-(v_i)$ and $h_{v_j} = f_\theta(v_j)$, $h_{v_k} = f_\theta(v_k)$, $dist(h_{v_i}, h_{v_j}) < dist(h_{v_i}, h_{v_k})$,

where $\mathcal{N}_m^+(v_i)$ and $\mathcal{N}_m^-(v_i)$ denote the set of positive and negative neighbors of node v_i in the graph $\mathcal{G}_m$, respectively.

An intuitive explanation for Definition 4.1 is that two nodes connected with a negative edge should be distant, i.e., greater than some positive value ϵ (Condition a), and nodes connected with positive edges should be closer in the embedding space than those connected with negative edges (Condition b). Most GNNs follow a message-passing scheme where the node representations are recursively updated by aggregating and transforming the information from the neighboring nodes. A recent study [33] shows that a maximally powerful GNN maps two nodes to the same location in the embedding space only if they have identical subtree structures. On the contrary, for two nodes with different rooted subtree structures, they will be mapped to different embeddings. Based on this conclusion, we will analyze the representation limitation of applying GNN models on the augmented signed graphs.

For graphs in the inter-set perspective, the common circle structures are quadruple circles (i.e., **butterflies**), while **triangles** are the basic circle structure in the intra-set perspective. Figure 4 illustrates four possible configurations for signed triangles. The different types of signed butterflies are summarized in Figure 5.

Definition 4.2. Balanced (unbalanced) triangles are cycles with 3 nodes containing even (odd) negative edges.

Definition 4.3. Balanced (unbalanced) butterflies are cycles with 4 nodes containing even (odd) negative edges.

Theorem 4.4. *A GNN cannot learn proper representations for nodes in a signed graph from unbalanced triangles and butterflies.*

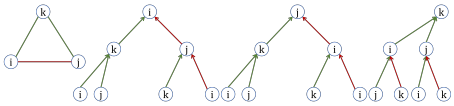

Figure 6: Rooted subtree of situation (c) in Figure 4

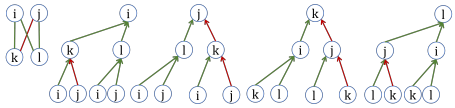

Figure 7: Rooted subtree of situation (f) in Figure 5

Proof. For simplicity, we only discuss the situation (c) in triangles and situation (f) in butterflies. Other unbalanced situations follow a similar proving process.

For situation (c) in Figure 4, we construct the 2-hop rooted subtree of node v_i, v_j, v_k and v_l in Figure 6, denoted as τ_i, τ_j, τ_k, respectively. We can observe that τ_i and τ_j are isomorphic. Based on the previous analysis [33], nodes i and j will be mapped to the same embeddings. On the contrary, as τ_i and τ_k are not isomorphic, they will be mapped to different embeddings. Therefore, we can get $dist(h_{v_i}, h_{v_j}) \leq dist(h_{v_i}, h_{v_k})$, where $dist$ is a distance metric. As a result, the representations of nodes connected with negative edges are closer than nodes connected with positive edges, which contradicts Condition b of Definition 4.1. Thus, the $h_{v_i}, h_{v_j}, h_{v_k}$ learned by GNN are not proper representations for node v_i, v_j, and v_k.

For situation (f) in Figure 5, we construct the 2-hop rooted subtree of node v_i, v_j, v_k, v_l in Figure 7, denoted as $\tau_i, \tau_j, \tau_k, \tau_l$, respectively. We can observe that τ_j and τ_k are isomorphic, and thus they will be mapped to the same embeddings. Meanwhile, τ_i and τ_k are not isomorphic, and thus they will be mapped to different embeddings. Therefore, we can get $dist(h_{v_i}, h_{v_k}) \leq dist(h_{v_j}, h_{v_k})$, where $dist$ is a distance metric, which means the representations of nodes connected with negative edges are closer than nodes connected with positive edges. Based on Definition 4.1, the learned $h_{v_i}, h_{v_j}, h_{v_k}$ are not proper representations for node v_i, v_j and v_k. □

Intuitively, Theorem 4.4 implies that unbalanced cycles may harm the performance of GNN models. Therefore, it is reasonable to claim that the signed graph augmentation methods that can reduce the number of unbalanced circles are more effective than those that cannot. Recall the three stochastic perturbation methods discussed in Section 4.1.2, i.e., randomly adding edges, randomly removing edges, and randomly flipping edges. Random edge removal can reduce the number of unbalanced circles by chance, while randomly adding edges could result in more unbalanced circles. For randomly flipping edge signs, some balanced circles may become unbalanced, while some unbalanced circles may become balanced. Overall, whether randomly flipping edge signs reduces unbalanced circles depends on the balanced degree of the datasets. Considering that most circle structures are balanced in real datasets [10], this perturbation method will eventually produce more unbalanced structures. Based on this analysis, randomly removing edges is in principle a more preferred method than the other two perturbation methods in signed graph representation learning.

4.2 Graph Encoder

After data augmentation, we obtain four augmented graphs as shown in Figure 2, where $\mathcal{G}_1, \mathcal{G}_2$ belong to the inter-set perspective and $\mathcal{G}_3, \mathcal{G}_4$ belong to the intra-set perspective. In signed graphs, positive and negative edges have distinct semantic properties – positive edges reflect closeness to friends and negative edges imply hatred to enemies. This motivates us to design two separate GNNs to aggregate information from positive and negative neighbors, respectively, which also serves for the contrastive objective to be introduced in the next subsection. Therefore, we further split each augmented graph $\mathcal{G}_m = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_m^+, \mathcal{E}_m^-)$ into two graphs containing only positive edges and only negative edges, referred to as positive graph $\mathcal{G}_m^+ = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_m^+)$ and negative graph $\mathcal{G}_m^- = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_m^-)$, respectively. Following the similar design from [24], positive GNN is leveraged to learn node representations from positive graphs while a negative GNN is used to learn representations from the negative graphs. Parameters are shared in the same perspective for positive (negative) GNNs. SBGCL is model-agnostic, and thus any GNN model can be used as its graph encoder. However, in this paper, we adopt GAT [28] as the graph encoder. The comparison between different graph encoders is out of the scope of this paper and we leave it as future work. Formally, the node representation in each augmented graph is computed as follows:

$$h_{i,m}^{(l+1),\sigma} = \text{GNN}_m^\sigma(h_{i,m}^{(l),\sigma}, \mathcal{G}_m^\sigma) \tag{2}$$

$$z_{i,m}^\sigma = \left[h_{i,m}^{(0),\sigma} \| h_{i,m}^{(1),\sigma} \| \cdots \| h_{i,m}^{(L),\sigma}\right] W_m^\sigma \tag{3}$$

where $\sigma \in \{+, -\}$, m refers to the m-th augmented graph, L denotes the number of GNN layers. $h_{i,m}^{(0),\sigma}$ denotes the input feature vector of the i-th node. W_m^σ is a learnable transformation matrix.

4.3 Multi-Perspective Contrastive Objective

In this subsection, we design a multi-perspective contrastive objective for robust signed bipartite graph representation learning, which consists of two losses – (1) *Perspective-specific contrastive loss* and (2) Cross-perspective contrastive loss.

4.3.1 Perspective-specific contrastive loss. As shown in Figure 2, after graph augmentation, SBGCL creates four augmented graphs, i.e., $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$ and $\mathcal{G}_4$, each of which is further split into positive and negative graphs. Since positive and negative graphs have distinct semantic properties, we define the perspective-specific contrastive losses for positive and negative augmented graphs separately. We focus our discussion on the positive graphs and define the loss for negative graphs in a similar way.

For any positive graph in each perspective, in order to obtain node representations invariant to small perturbations, SBGCL maximizes the agreements between representations of the same node learned from the other positive augmented graph in the same perspective while minimizing the similarities between the representations of different nodes. Figure 8 illustrates the process. For example, the representation of the i-th node in graph $\mathcal{G}_1^+$ of the inter-set perspective, i.e., $z_{i,1}^+$, should be consistent with representations generated from the same node in the other positive augmented graph $\mathcal{G}_2^+$ in the same perspective, i.e., $z_{i,2}^+$. Thus, we use the representations of the same node generated from different augmented graphs in the same perspective as positive samples. Meanwhile, the representation of a node (e.g., $z_{i,1}^+$) should be distinct from other nodes' representations (e.g., $z_{j,2}^+$). We take the representations generated from different nodes in the other positive augmented graph in the same perspective as negative examples. Inspired by the InfoNCE loss [21, 26], the perspective-specific contrastive loss for positive augmented graphs is defined as follows:

$$\mathcal{L}_{\text{per}}^+ = -\frac{1}{I}\sum_{i=1}^{I} \log \frac{\exp\left(\text{sim}\left(z_{i,m}^+, z_{i,m'}^+\right)/\tau\right)}{\sum_{j=1,j\neq i}^{I} \exp\left(\text{sim}\left(z_{i,m}^+, z_{j,m'}^+\right)/\tau\right)} \tag{4}$$

where I is the number of nodes in a mini-batch, $z_{i,m}^+$ represents the representation of node i in the m-th augmented positive graph, $sim(\cdot,\cdot)$ represents the similarity function between the two representations (e.g., cosine similarity) and τ denotes the temperature parameter. Likewise, the perspective-specific contrastive loss for negative graphs is:

$$\mathcal{L}_{\text{per}}^- = -\frac{1}{I}\sum_{i=1}^{I} \log \frac{\exp\left(\text{sim}\left(z_{i,m}^-, z_{i,m'}^-\right)/\tau\right)}{\sum_{j=1,j\neq i}^{I} \exp\left(\text{sim}\left(z_{i,m}^-, z_{j,m'}^-\right)/\tau\right)} \tag{5}$$

Combining the above two losses, we obtain the perspective-specific contrastive loss:

$$\mathcal{L}_{\text{per}} = \mathcal{L}_{\text{per}}^+ + \mathcal{L}_{\text{per}}^- \tag{6}$$

4.3.2 Cross-perspective contrastive loss. The perspective-specific contrastive loss only makes the node representation learned from positive and negative graphs in each perspective invariant to small perturbations. Still, node representations of the positive and negative graphs in different perspectives are learned independently. We follow two intuitions to design a cross-perspective contrastive loss to obtain the final node representations by contrasting the information from the positive and negative graphs in both perspectives. Our first intuition is that the final node representation should capture both explicit and implicit information, and thus should be close to representations obtained from the positive graphs in both perspectives. Our second intuition is that the final node representation should be distant from the representations obtained from negative graphs in both perspectives because the information aggregated from negative neighbors tends to diverge from the target node. Fig. 9 illustrates the idea of cross-perspective contrastive learning. SBGCL makes the final representation of each node close to its representations obtained from positive graphs across both perspectives, and distant from its representations obtained from negative graphs across both perspectives. Specifically, we compute the final representation of the ith node by concatenating $z_{i,1}^+$, $z_{i,2}^+$, $z_{i,3}^+$, $z_{i,4}^+$, $z_{i,1}^-$, $z_{i,2}^-$, $z_{i,3}^-$, $z_{i,4}^-$, since all these representations contain useful information of diverse aspects, which is formulated as:

$$z_i = g\left(z_{i,1}^+ \| z_{i,2}^+ \| \| z_{i,3}^+ \| z_{i,4}^+ \| z_{i,1}^- \| z_{i,2}^- \| z_{i,3}^- \| z_{i,4}^-\right) \tag{7}$$

where $g(\cdot)$ is an MLP layer and $z_i \in \mathbb{R}^d$ is the final representation of node i. We regard the representations of the i-th node from positive graphs as positive samples and representations obtained from negative graphs as negative samples. In this way, we can make nodes more similar to neighbors connected with positive

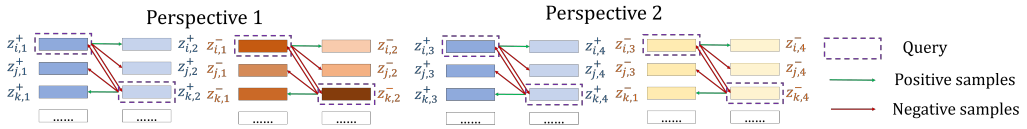

Figure 8: Inter-view Contrastive Learning

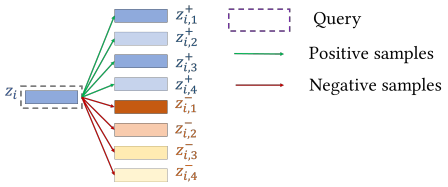

Figure 9: Intra-view Contrastive Learning

edges and dissimilar to those with negative edges. Formally, the cross-perspective contrastive objective is defined as follows:

$$\mathcal{L}_{\text{cross}} = -\frac{1}{I}\sum_{i=1}^{I}\log\frac{\sum_{m=1}^{M}\exp\left(\text{sim}\left(z_i, z_{i,m}^{+}\right)/\tau\right)}{\sum_{m=1}^{M}\exp\left(\text{sim}\left(z_i, z_{i,m}^{-}\right)/\tau\right)} \tag{8}$$

where M denotes the number of augmented graphs, which equals to 4 in our paper.

4.3.3 Combined contrastive loss. We perform both perspective-specific and cross-perspective contrastive learning and combine the two losses as follows:

$$\mathcal{L}_{CL} = (1-\alpha)\cdot\mathcal{L}_{\text{per}} + \alpha\cdot\mathcal{L}_{\text{cross}} \tag{9}$$

where α is the weight coefficient that controls the significance between two losses.

4.4 Loss Function

In this paper, we focus on the most common analysis task for signed graphs, i.e., link sign prediction [6, 19, 20, 30] that predicts the edge sign (either positive or negative) between two nodes. For signed bipartite graphs, the link sign prediction task only happens between two nodes from different sets. Specifically, after obtaining the final representations in Equation 7, we employ a 2-layer MLP to estimate the sign scores between two nodes from different sets:

$$y_{\text{pred}} = \text{sigmoid}\left(\text{MLP}\left(z_{u_i} \| z_{v_j}\right)\right) \tag{10}$$

where y_{pred} is the predicted score of edge sign between nodes $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$. The larger the y_{pred} is, the higher probability that the edge sign is positive. Contrarily, the smaller y_{pred} is, the higher probability that the edge sign is negative. Following prior work [10], we use cross-entropy as the loss function of the link sign prediction task :

$$\mathcal{L}_{\text{label}} = -y\cdot\log y_{\text{pred}} + (1-y)\cdot\log\left(1-y_{\text{pred}}\right) \tag{11}$$

where y is the ground truth mapped from $\{-1, 1\}$ to $\{0, 1\}$.

At last, SBGCL is trained by the joint loss of link sign prediction and the combined contrastive loss:

$$\mathcal{L} = \mathcal{L}_{\text{label}} + \beta\cdot\mathcal{L}_{CL} \tag{12}$$

Table 1: Statistics on Signed Bipartite Networks.

	Review	Bonanza	ML-1M	Amazon-Book
$\mathcal{U}$	182	7,919	6,040	35,736
$\mathcal{V}$	304	1,973	3,952	38,121
$\lvert\mathcal{E}\rvert = \lvert\mathcal{E}^+\rvert + \lvert\mathcal{E}^-\rvert$	1,170	36,543	1,000,209	1,960,674
%Positive Edges	0.403	0.980	0.575	0.806
%Negative Edges	0.597	0.020	0.425	0.194

where β is the weight that controls the contribution of the contrastive loss.

5 EXPERIMENTS

In this section, we evaluate the performance of link sign prediction of our proposed method SBGCL on four real-world datasets through answering the following questions:

- **Q1**: Does our proposed model SBGCL outperform existing baselines on link sign prediction task?
- **Q2**: Can the experimental results confirm our theoretical analysis results for graph perturbation methods?
- **Q3**: Is the proposed model sensitive to hyperparamters ? How do key hyperparameters impact the model performance?

5.1 Datasets

We conduct experiments on four real-world datasets, i.e., Review, Bonanza, ML-1M and Amazon-Book. The main statistics of each dataset are summarized in Table 1. In the following, we explain important characteristics of the datasets briefly.

Review. Review [12] is the peer review data from a top computer science conference[2]. Reviewers $\mathcal{U}$ can give "SA"(Strong Accept), "A"(Accept), "WA"(Weak Accept), "WR"(Weak Reject), "R"(Reject), and "SR"(Strong Reject) to papers $\mathcal{V}$ after reviewing, where "SA", "A" and "WA" are regarded as positive edges and "SR", "R", and "WR" are regarded as negative edges.

Bonanza. Bonanza [3] is a e-commerce website where users can purchase products from a seller and rate the seller with "Positive", "Neutral" or "Negative" scores.

ML-1M. MovieLens-1M (ML-1M) [4] is a platform where users can rate movies from 1 to 5.

Amazon-Book. This is a Amazon-Review dataset [5] which contains book ratings from users (also from 1 to 5)

Following the experimental settings in [5], we randomly select 10% of the edges as test set, 5% for validation set, and the remaining as training set for each dataset. We run with different train-val-test splits for 5 times to get the average scores and standard deviation.

[2]Due to anonymity, we removed the name of the conference.
[3]https://www.bonanza.com/
[4]https://grouplens.org/datasets/movielens/1m/
[5]https://jmcauley.ucsd.edu/data/amazon/index.html

5.2 Baselines and Experiment Setting

We compare our method SBGCL with several baselines including Random Embeddings, Unsigned Network Embeddings, Unsigned GNN, Signed GNN.

Random Embeddings: It generate d dimensional random values from a uniform distribution over $[0, 1)$. Given embeddings z_{u_i} and z_{v_j}, we concatenate them and train a Logistic Regressor(LR) on the training data to predict the sign between u_i and v_j in the test data. Since random embeddings do not encode any information of the graph, it is supposed to perform the worst [10].

Unsigned Network Embeddings: Node2vec [8] is a classical unsigned network embedding method where we only consider positive edges to learn node embeddings for node u_i and v_j. Like Random Embeddings, we concatenate node embeddings z_{u_i} and z_{v_j}, and use LR to predict the sign of links.

Unsigned GNN: We employ two classical GNN models (i.e., GCN [17] and GAT [29]) and one contrastive learning based model GCA [39]. These methods are designed for unsigned graphs, thus, as mentioned before, we only consider the positive edges to learn node embeddings in the experiments.

Signed Graph Neural Networks: SGCN [6] and SNEA [19] respectively generalize GCN [17] and GAT [29] to signed graphs based on message mechanism and balance theory. SGCL [24] is the first research to employ graph contrastive learning on unipartite signed graphs. SBGNN [10] designs a new message-passing mechanism for signed bipartite graph which is our most competitive competitor.

For a fair comparison, we set all the node embedding dimension to 32 which is as same as that in SBGNN [10] for all embedding based methods. For other parameters in baselines, we follow the recommended settings in their original papers. For our SBGCL, we use PyTorch [22] and its associated graph libraries, PyTorch Geometric [7] to implement it. We use Adam optimizer with an initial learning rate of 0.005 and a weight decay of 1e-5. We run 2000 epochs for SBGCL and choose the model that performs the best AUC metrics on the validation set.

The evaluation task is link sign prediction which is a binary classification problem, we use AUC, Binary-F1, Macro-F1 and Micro-F1 to evaluate the results. These metrics are widely used in existing work [10, 20]. Note that, for all these four evaluation metrics, a higher value indicates a better performance.

5.3 Performance on Link Sign Prediction (Q1)

The performance of all methods is summarized in Table 2. We have bolded the highest value of each row and underlined the second highest value. From Table 2, we summarize some major observations as follows:

- Even with random embedding, LR can still achieve a certain accuracy (partial AUC>0.5) on link sign prediction task, which demonstrates the classification ability of LR.
- Unsigned GNNs improve the prediction results compared to random embedding. For example, GRACE outperforms random embedding by 19.43%, 5.80%, 33.05% and 75.75% on AUC in Review, Bonanza, ML-1M and Amazon-book, respectively. The results demonstrate that positive structure information is helpful in link sign prediction task.
- Signed GNNs can employ both positive and negative edges. The experimental results demonstrate the negative edges contribute to the link sign prediction tasks. Even the worst-performing signed GNN model, i.e., SGCN, outperforms the best unsigned GNN model GRACE in several metrics, especially in ML-1M(2.34% on AUC) and Amazon-book (6.49% on AUC) datasets.
- Our proposed model SBGCL achieves the best results on most metrics, except for Binary-F1 and Macro-F1 on Bonanza dataset. SBGCL shows a significant improvement over the state-of-the-art signed bipartite GNN, i.e., SBGNN, by 10.82%, 2.27%, 5.01% and 5.55% in terms of AUC on the four datasets, respectively.

5.4 Performance on Different Graph Pertubations (Q2)

In Section 4.1.3, we give a theoretical analysis of graph augmentation methods. We prove that current GNN encoders cannot learn proper representations for nodes from unbalanced circles. Thus, the signed graph augmentation methods that can reduce the number of unbalanced circles are more effective than those that cannot. Next, we conduct experiments to investigate the effectiveness of different graph augmentation methods. We compare the three graph augmentation methods described in Section 4.1.2, random edge deletion, random sign flip, random edge addition. The AUC scores of SBGCL with the different stochastic augmentations are summarized in Table 3. With a small perturbation probability (e.g., 5%), random edge deletion performs better than the other two graph stochastic augmentation methods. This is consistent with the conclusion of our theoretical analysis.

5.5 Ablation Study and Hyper-parameter Analysis (Q3)

5.5.1 Ablation Study. We conduct ablation study to investigate the effectiveness of different components in our proposed model, where we choose edge deletion as the graph augmentation in the next experiments. Concretely, we compare our method SBGCL with four variants: $SBGCL_{w/o\ aug}$, $SBGCL_{w/o\ GCL}$, $SBGCL_{w/o\ inter}$ and $SBGCL_{w/o\ intra}$, which are defined as follows:

- SBGCL$_{w/o\ aug}$: This variant removes the stochastic perturbation. For the edge deletion augmentation, we set the deletion ratio to zero, i.e., $p = 0$.
- SBGCL$_{w/o\ \mathcal{L}_{CL}}$: This variant removes the graph contrastive loss ($\mathcal{L}_{CL}$) and only considers link sign prediction loss, i.e., $\beta = 0$.
- SBGCL$_{w/o\ \mathcal{L}_{\text{per}}}$: This variant ignores the perspective-specific contrastive loss, i.e., $\alpha = 1$.
- SBGCL$_{w/o\ \mathcal{L}_{\text{cross}}}$: This variant ignores the cross-perspective contrastive loss, i.e., $\alpha = 0$.

The AUC comparisons of SBGCL with the four variants are summarized in Table 4. From this table, we can conclude that:

- The performance of SBGCL$_{w/o\ aug}$ and SBGCL$_{w/o\ \mathcal{L}_{CL}}$ decrease greatly in the first two datasets (Review and Bonanza), which demonstrates the effectiveness of both stochastic perturbation and contrastive loss in contrastive learning.
- The performance of SBGCL$_{w/o\ \mathcal{L}_{\text{per}}}$ and SBGCL$_{w/o\ \mathcal{L}_{\text{cross}}}$ demonstrate both losses can boost the performance of SBGCL.

Table 2: The results of Link Sign Prediction on four datasets

Dataset	Metric	Random Embedding	Unsigned Network Embeddings	Unsigned GNN			Signed GNN			Our Method	
		Random	Node2vec	GCN	GAT	GRACE	SGCN	SGCL	SBGNN	SBGCL	Impro
Review	AUC	0.514± 0.010	0.508± 0.031	0.522± 0.053	0.643± 0.052	0.613 ± 0.013	0.610 ± 0.034	0.729 ± 0.024	0.675 ± 0.035	**0.748** ± 0.017	2.61%
	Binary-F1	0.435 ± 0.014	0.129± 0.087	0.396± 0.084	0.538± 0.076	0.592 ± 0.020	0.593 ± 0.031	0.656 ± 0.015	0.637± 0.030	**0.706** ± 0.022	7.62%
	Macro-F1	0.430 ± 0.025	0.422± 0.055	0.514± 0.059	0.639 ± 0.064	0.621 ± 0.021	0.601 ± 0.011	0.631 ± 0.023	0.663 ± 0.016	**0.747** ± 0.016	12.67%
	Micro-F1	0.539 ± 0.018	0.571 ± 0.049	0.547 ± 0.059	0.668± 0.065	0.653 ± 0.017	0.637 ± 0.023	0.633 ± 0.017	0.667 ± 0.016	**0.754** ± 0.015	13.04%
Bonanza	AUC	0.503 ± 0.018	0.562± 0.018	0.506± 0.011	0.498± 0.006	0.532 ± 0.018	0.587± 0.007	0.584 ± 0.024	0.577 ± 0.033	**0.590** ± 0.031	1.02%
	Binary-F1	0.608 ± 0.015	0.719± 0.125	0.757± 0.372	0.582± 0.476	0.953 ± 0.013	0.896 ± 0.014	**0.987** ± 0.002	0.962 ± 0.001	0.973 ± 0.002	–
	Macro-F1	0.327 ± 0.021	0.385 ± 0.066	0.396 ± 0.184	0.304± 0.232	0.505 ± 0.013	0.487 ± 0.014	0.514 ± 0.020	0.540 ± 0.015	**0.558** ± 0.013	3.33%
	Micro-F1	0.445 ± 0.025	0.580 ± 0.141	0.722 ± 0.354	0.574± 0.454	0.923± 0.016	0.814± 0.033	**0.974** ± 0.005	0.927 ± 0.006	0.947 ± 0.005	–
ML-1M	AUC	0.473 ± 0.019	0.635 ± 0.002	0.502 ± 0.003	0.508 ± 0.003	0.629 ± 0.007	0.632 ± 0.015	0.632 ± 0.014	0.652 ± 0.007	**0.685** ± 0.004	5.06%
	Binary-F1	0.322 ± 0.023	0.582 ± 0.002	0.344 ± 0.233	0.401± 0.068	0.637 ± 0.017	0.652 ± 0.024	0.673 ± 0.015	0.699 ± 0.009	**0.702** ± 0.011	0.42%
	Macro-F1	0.352 ± 0.021	0.609 ± 0.002	0.412± 0.062	0.467± 0.024	0.623 ± 0.011	0.615 ± 0.003	0.662 ± 0.002	0.653 ± 0.006	**0.678** ± 0.006	2.42%
	Micro-F1	0.346 ± 0.023	0.611 ± 0.002	0.476± 0.046	0.478± 0.015	0.614 ± 0.006	0.627 ± 0.013	0.652 ± 0.002	0.674 ± 0.004	**0.680** ± 0.006	0.89%
Amazon-Book	AUC	0.343 ± 0.004	0.547 ± 0.002	0.501 ± 0.001	0.515± 0.008	0.602 ± 0.013	0.593 ± 0.003	0.613 ± 0.007	0.603 ± 0.006	**0.637** ± 0.005	3.91%
	Binary-F1	0.402 ± 0.024	0.656 ± 0.004	0.618± 0.331	0.634± 0.008	0.651 ± 0.007	0.693 ± 0.003	0.710 ± 0.016	0.720 ± 0.014	**0.734** ± 0.014	1.94%
	Macro-F1	0.362 ± 0.024	0.488 ± 0.002	0.395± 0.114	0.489± 0.023	0.512 ± 0.017	0.504 ± 0.020	0.502 ± 0.015	0.552 ± 0.008	**0.587** ± 0.011	6.24%
	Micro-F1	0.414 ± 0.023	0.543 ± 0.004	0.579± 0.241	0.615± 0.011	0.601 ± 0.013	0.582 ± 0.023	0.604 ± 0.014	0.612 ± 0.006	**0.640** ± 0.015	4.07%

Table 3: The AUC performances with different graph perturbation methods

Aug.	Review	Bonanza	ML-1M	Amazon-Book
Delete	0.748 ± 0.017	0.590 ± 0.031	0.685 ± 0.004	0.639 ± 0.005
Flip	0.728 ± 0.039	0.588 ± 0.039	0.683 ± 0.002	0.627 ± 0.010
Add	0.704 ± 0.057	0.586 ± 0.024	0.683 ± 0.006	0.630 ± 0.011

Table 4: The AUC performances with SBGCL and its variants

Models	Review	Bonanza	ML-1M	Amazon-Book
SBGCL	0.748 ± 0.017	0.590 ± 0.031	0.685 ± 0.004	0.639 ± 0.005
SBGCL$_{w/o\ aug}$	0.703 ± 0.051	0.578 ± 0.033	0.674 ± 0.005	0.625 ± 0.009
SBGCL$_{w/o\ \mathcal{L}_{CL}}$	0.718 ± 0.048	0.557 ± 0.023	0.680 ± 0.004	0.614 ± 0.010
SBGCL$_{w/o\ \mathcal{L}_{\text{per}}}$	0.691 ± 0.064	0.559 ± 0.043	0.683 ± 0.003	0.630 ± 0.006
SBGCL$_{w/o\ \mathcal{L}_{\text{cross}}}$	0.721 ± 0.017	0.564 ± 0.018	0.680 ± 0.004	0.621 ± 0.004

5.5.2 Hyper-parameters Analysis. The key hyper-parameters in our model: p stochastic perturbation ratio (only considering random edge deletion), α that balances the perspective-specific and cross-perspective contrastive losses, β that balances the contrastive loss and link sign prediction loss and the node embedding dimension d. The default setting for the hyper-parameters are $p = 0.1$, $\alpha = 0.8$, $\beta = 5e-4$, $d = 32$. We fix other hyper-parameters when evaluating each of them. To analyze the effects of node dimension d of SBGCL, we choose $d \in \{4, 8, 16, 32, 64, 128\}$. For α, we choose the value from $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. For β, we choose the value from $\{1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1\}$. For the perturbation ratio p, we choose $p \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The AUC performances are reported in Fig. 10 and observations are summarized as follows:

- Fig.10(a) shows that the AUC of SBGCL increases as d increases from 4 to 32 on most datasets, e.g., Review (from 0.66 to 0.70) and ML-1M (from 0.63 to 0.67). When d is greater than 32, the performance of SBGCL degrades due to overfitting.
- Fig. 10(b) shows that the model achieves the best performance when α is set to a value between 0.5 and 0.75. The results suggest that the cross-perspective loss tends to be more important because it makes the learned representation consistent across different augmented graphs while retaining the explicit and implicit relations. The model performance degrades when $\alpha = 0$ and $\alpha = 1$, which indicates that both losses are indispensable.
- From Figure 10(c), the performance rises when the value of β increases. The result demonstrates that the graph contrastive learning loss is essential to the link sign prediction task. However, when β becomes too large, the information from ground truth labels is weaken. Therefore, we observe that performance of model drops sharply.
- From Figure 10(d), we can see smaller perturbations (e.g., p=0.1) can improve model performance. But as the perturbation probability increases (e.g., p=0.5), the performance of the model degrades sharply due to the loss of too much structure information.

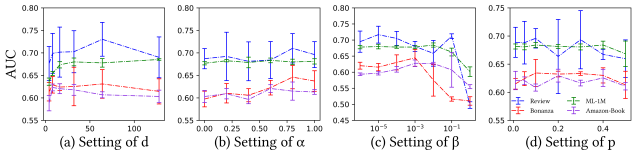

Figure 10: Parameter sensitivity of SBGCL with regard to d, α, β, p on four real-world datasets

6 CONCLUSION

In this paper, we propose a novel graph contrastive representation model, which is the first work to employ graph contrastive learning to signed bipartite graph. We propose a two-level graph augmentation method and theoretically analyze the effectiveness of stochastic graph augmentation methods. Further, we design a new multi-perspective contrastive loss for learning robust node representations while retaining the explicit and implicit relations between nodes in singed bipartite graphs.

Acknowledgement. This research is supported by the Marsden Fund Council from Government funding (MFP-UOA2123), administered by the Royal Society of New Zealand. The first author and sixth author are supported by a PhD scholarship from China Scholarship Council.

REFERENCES

[1] Francesco Bonchi, Edoardo Galimberti, Aristides Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. 2019. Discovering polarized communities in signed networks. In *Proceedings of the 28th acm international conference on information and knowledge management*. 961–970.

[2] Jiangxia Cao, Xixun Lin, Shu Guo, Luchen Liu, Tingwen Liu, and Bin Wang. 2021. Bipartite graph embedding via mutual information maximization. In *Proceedings of the 14th ACM international conference on web search and data mining*. 635–643.

[3] Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S Dhillon. 2012. Scalable clustering of signed networks using balance normalized cut. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 615–624.

[4] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2022. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. *arXiv preprint arXiv:2204.08570* (2022).

[5] Tyler Derr, Cassidy Johnson, Yi Chang, and Jiliang Tang. 2019. Balance in signed bipartite networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1221–1230.

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

[7] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[8] 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.

[9] Fritz Heider. 1946. Attitudes and cognitive organization. *The Journal of psychology* 21, 1 (1946), 107–112.

[10] Junjie Huang, Huawei Shen, Qi Cao, Shuchang Tao, and Xueqi Cheng. 2021. Signed Bipartite Graph Neural Networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 740–749.

[11] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. 2019. Signed graph attention networks. In *International Conference on Artificial Neural Networks*. Springer, 566–577.

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

[13] Baoyu Jing, Yuchen Yan, Kaize Ding, Chanyoung Park, Yada Zhu, Huan Liu, and Hanghang Tong. 2023. STERLING: Synergistic Representation Learning on Bipartite Graphs. *arXiv preprint arXiv:2302.05428* (2023).

[14] Jinhong Jung, Woojeong Jin, Lee Sael, and U Kang. 2016. Personalized ranking in signed networks using signed random walk with restart. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 973–978.

[15] Jinhong Jung, Jaemin Yoo, and U Kang. 2020. Signed Graph Diffusion Network. *arXiv preprint arXiv:2012.14191* (2020).

[16] 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*. 509–518.

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

[18] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. 2010. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM international conference on data mining*. SIAM, 559–570.

[19] 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*. 4772–4779.

[20] Haoxin Liu, Ziwei Zhang, Peng Cui, Yafeng Zhang, Qiang Cui, Jiashuo Liu, and Wenwu Zhu. 2021. Signed Graph Neural Network with Latent Groups. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1066–1075.

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[23] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1150–1160.

[24] Lin Shu, Erxin Du, Yaomin Chang, Chuan Chen, Zibin Zheng, Xingxing Xing, and Shaofeng Shen. 2021. SGCL: Contrastive Representation Learning for Signed Graphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1671–1680.

[25] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1283–1293.

[26] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems* 29 (2016).

[27] 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.

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[29] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. *ICLR (Poster)* 2, 3 (2019), 4.

[30] 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.

[31] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. 2020. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659* (2020).

[32] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[33] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.

[34] Yaming Yang, Ziyu Guan, Zhe Wang, Wei Zhao, Cai Xu, Weigang Lu, and Jianbin Huang. 2022. Self-supervised Heterogeneous Graph Pre-training Based on Structural Clustering. *arXiv preprint arXiv:2210.10462* (2022).

[35] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6210–6219.

[36] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems* 33 (2020), 5812–5823.

[37] 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.

[38] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131* (2020).

[39] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*. 2069–2080.