Structure Balance and Gradient Matching-Based Signed Graph Condensation

Rong Li[1], Long Xu[1], Songbai Liu[1], Junkai Ji[1], Lingjie Li[2], Qiuzhen Lin[1], Lijia Ma[1]*

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[2]College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China
ljma1990@szu.edu.cn

Abstract

Training graph neural networks (GNNs) for graph representation has received increasing concerns due to its outstanding performance in the link prediction and node classification tasks, but it incurs much time and storage for tackling large-scale graphs. To alleviate this issue, graph condensation has been emerged to condense the large graph into a small but highly-informative graph, while achieving comparable performance of GNNs trained on the small graph and large graph. However, existing works mainly focus on the gradient or distribution matching under GNN training trajectories to condense simple link structures, while overlooking the structure matching for condensing signed graph that exists conflict links and structural balance among nodes. To bridge this gap, we propose a novel **S**tructure Balance and **G**radient Matching-Based **S**igned **G**raph **C**ondensation (SGSGC) method for condensing signed graph with node attributes, conflict links and structural balance into informative smaller ones. Specifically, we first propose a structure-balanced matching to match the structural balance between the original and condensed signed graph, and then combine it with the gradient matching to condense signed graph for the link sign prediction task, while preserving both conflicting link structures and node attributes. Moreover, we use the feature smoothing and the graph sparsification technique to improve the robustness for the GNN training, respectively. Finally, a bi-level optimization technique is proposed to simultaneously find the optimal node attributes and conflict structure of the condensed graph. Experiments on six datasets demonstrate that SGSGC achieves excellent performance. On Epinions, 94% test accuracy of training on the original signed graph, while reducing their graph size by 99.95% - 99.99%, and there exist 2.24% – 6.26% accuracy improvements for link sign prediction compared to the state-of-the-arts.

Code — https://github.com/BaoFit/SGSGC

Introduction

Graph Neural Networks (GNNs) (Velickovic et al. 2018) have been widely used to extract essential information about characteristics and structure from graphs using graph representation learning, which can be well used to tackle downstream graph tasks such as link prediction (Zhang and Chen 2018) and node classification (Zhao, Zhang, and Wang 2021). However, training GNNs on large graphs is highly resource-intensive in time, storage, and computation due to the large scale, and redundancy of real-world datasets and the incremental re-training of hyperparameter settings, extensive weights, and neural architectures.

To address this challenge, a simple but effective idea is to simplify the large-scale graph while preserving the essential structure and properties, so as to decrease the cost of training GNN on the graph for tasks. Such representative methods include graph sparsification (Batson et al. 2013) and graph coarsening (Purohit et al. 2014). The graph sparsification tries to reduce the size of links by deleting redundant links, while the graph coarsening aims to reduce the size of nodes by merging similar nodes. However, these methods mainly focus on preserving certain graph properties such as approximate eigenvalues, pairwise distances, and spectra, which may lead to the loss of comprehensive properties of graph and the sub-optimal for the downstream performance of GNNs.

To significantly simplify graph and preserve sufficient properties, graph condensation (GC) has been emerged to condense the large graph into a small but highly-informative one through learning both the synthetic graph structure and the weight parameters of GNNs, while ensuring comparable performance of GNNs trained on both the original and condensed graphs (Jin et al. 2022; Yang et al. 2023). There are two main strategies to minimize the performance gap between GNNs trained on the original and condensed graphs: gradient matching (Jin et al. 2022) and distribution matching (Liu et al. 2022). Gradient matching minimizes the gradient difference between GNNs trained on the original and condensed graphs, and ensures similar learning trajectories. Distribution matching minimizes the difference of the node embedding distribution between the original and condensed graphs, ensuring the preservation of structural features and links. Moreover, a combination of gradient matching and laplacian distribution matching was proposed in SGDD (Yang et al. 2023), resulting in a significant performance improvement.

While the aforementioned methods have achieved promising results for GC, they mainly follow the gradient matching under GNN training trajectories to condense simple link structures and node attributes, while overlooking the

*Corresponding author.

structure matching for condensing signed graphs that exists conflict links and structural balance among nodes. In the real-world cases, there exist low-level conflicting relationships (e.g., enemy and friend, cooperation and competition, like and dislike, etc.) among entities, and high-level unbalanced structures among communities. The unbalanced structures were first proposed by the structural balance theory (Cartwright and Harary 1956), namely the friend of my friend is my enemy is unbalanced, and the friendly relationships across two communities and the hostile relationships within one communities are unbalanced. The existence of conflicting structures in signed graphs occurs a great challenging for the existing GC methods, as they cannot use either gradient matching or distribution matching to preserve both low-level conflicting relationships and high-level structural balance. Moreover, inaccuracies in assigning link signs can lead to erroneous interactions, which may accumulate over time and ultimately result in suboptimal and unstable performance.

To bridge this gap, we propose a novel Structure Balance and Gradient Matching-Based Signed Graph Condensation (SGSGC) method for condensing signed graphs, taking the structural balance matching and the preserving conflicting structures into consideration. The main idea behind the structure-balanced matching is to establish positive and negative association among nodes with positive and negative links in the original and condensed graphs, respectively. Then, the structural balance matching is combined with the gradient matching to condense signed graphs for the link sign prediction task, while preserving both conflicting link structures and node attributes. Moreover, to improve graph stability, we integrate feature smoothing techniques (Jin et al. 2020) to enhance the discrimination between positive and negative relationships among nodes and train a GNN on the condensed sparsified graph by removing low-weight edges. Finally, a bi-level optimization technique is proposed to simultaneously find the optimal node attributes and conflict structure of the condensed graph. In summary, the main contribution of this work is as follows:

- We propose the SGSGC method for condensing signed graphs that exists conflicting structures and structural balance among nodes.
- In SGSGC, we propose a structure-balanced matching to align the structural balance between the original and condensed signed graphs, and then combine it with gradient matching to preserve conflicting link structures and node attributes for the link sign prediction task. Additionally, we employ feature smoothing and graph sparsification techniques to enhance the robustness and efficiency of GNN training.
- Extensive experiments validate the superiority of the proposed SGSGC over the state-of-the-arts for condensing signed graphs in terms of accuracy for link sign prediction. The results also show the applicability of SGSGC to various GNN models.

Preliminaries

Notation

A signed graph $\mathcal{G}$ can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ and $\mathcal{E}$, denote sets of nodes and edges, respectively, and while $\mathbf{X}$ represents the properties of nodes. In $\mathcal{G}$, $\mathcal{E} = (\mathcal{E}^+, \mathcal{E}^-)$ is further divided into the positive link set $\mathcal{E}^+$ and negative link set $\mathcal{E}^-$, and it can be denoted by a adjacent matrix $\mathbf{A} \in [a_{ij}]^{N \times N}$, where each element $a_{ij} \in [-1, 1]$ represents the weight and sign of edge (v_i, v_j). Specifically, an edge with $a_{ij} > 0$, $a_{ij} < 0$, and $a_{ij} = 0$ represent positive, negative, and non-existent edge, respectively. Moreover, $\mathbf{X} \in \mathbb{R}^{N \times d}$ is the d-dimensional node feature matrix, where N is the number of nodes. In this case, the signed network can be formally represented as $\mathcal{G} = (\mathbf{A}, \mathbf{X})$. For each node $v_i \in \mathcal{V}$, its positive and negative neighbors are denoted as $\mathcal{N}_i^+ = \{v_l | (v_i, v_l) \in \mathcal{E}^+\}$ and $\mathcal{N}_i^- = \{v_k | (v_i, v_k) \in \mathcal{E}^-\}$, respectively.

Graph Condensation

Given a signed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, graph condensation aims to create a synthetic signed graph $\mathcal{S} = (\mathbf{A}', \mathbf{X}')$ with a much smaller set of nodes $\mathcal{V}' = \{v'_1, v'_2, ..., v'_n\}$, where $n \ll N$, such that a GNN trained on $\mathcal{S}$ maintains comparable performance to one trained on the original graph $\mathcal{G}$. This objective can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{\mathcal{S}} \mathrm{E}_{\theta_0 \sim P_{\theta_0}} \left[\mathcal{L} \left(\hat{\mathbf{Y}}, \mathbf{Y} \right) \right] \\ \text{s.t.} \quad &\theta_{\mathcal{S}} = \arg\min_{\theta} \mathcal{L} \left(\hat{\mathbf{Y}}', \mathbf{Y}' \right) \end{aligned} \tag{1}$$

where GNN_θ is a graph neural network with parameters θ, and $\mathcal{L}$ is a loss function to measure the discrepancy between model prediction and ground truth. $\mathbf{Y}'$ and $\mathbf{Y}$ denote the labels of the condensed and original graph, while $\hat{\mathbf{Y}}'$ and $\hat{\mathbf{Y}}$ represent the respective prediction labels through GNN_θ on them. In this case, $\hat{\mathbf{Y}}'$ and $\hat{\mathbf{Y}}$ are computed as $\mathrm{GNN}_{\theta_{\mathcal{S}}}(\mathbf{A}', \mathbf{X}')$ and $\mathrm{GNN}_\theta(\mathbf{A}, \mathbf{X})$, respectively. $\theta_0 \sim P_{\theta_0}$ indicates that θ_0 is sampled from a distribution of random initialization to reduce overfitting.

Our Approach

In this section, we introduce the proposed SGSGC, which mainly consists of three main modules (see Figure 1): parameterization for condensed graph, GNN training, and graph optimization.

Parameterization for Condensed Graph

The signed graph generation in SGSGC aims to facilitate flexible adjustment and optimization of the synthetic graph. This is achieved by parameterizing the graph's attribute features $\mathbf{X}'$ and structure $\mathbf{A}'$. To achieve this objective, a generative model denoted as $\mathbf{A}' = \mathrm{GEN}_\psi(\mathbf{X}')$ is adopted to create $\mathbf{A}'$ based on $\mathbf{X}'$. In this model, each edge $a'_{ij} \in \mathbf{A}'$ is generated as follows:

$$a'_{ij} = \sigma \left(\frac{\mathrm{MLP}_\psi([x'_i; x'_j]) + \mathrm{MLP}_\psi([x'_j; x'_i])}{2} \right) \tag{2}$$

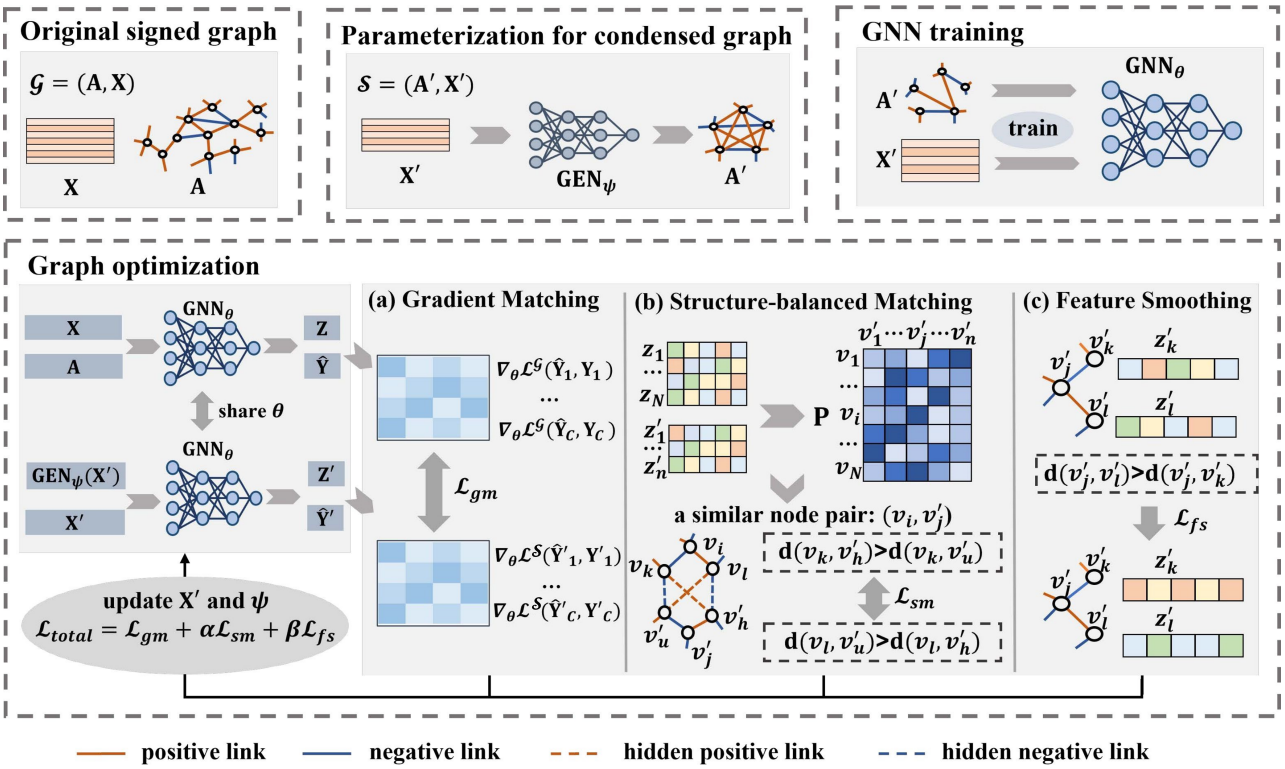


Figure 1: The overall framework of the proposed SGSGC.

where MLP_ψ is a multi-layer perceptron with parameters ψ, σ represents the activation function $\tanh$ which controls the output to $a' \in [-1, 1]$, and $[:;:]$ denotes a vector concatenation.

GNN Training

In SGSGC, the GNN training aims to obtain comparable sign link prediction performance of GNNs trained on the condensed graph and the original graph. Given n test samples, this objective can be formulated as follows:

$$\mathcal{L}_{GNN} = \frac{1}{n}\mathcal{L}\left(\hat{\mathbf{Y}}', \mathbf{Y}'\right) \tag{3}$$

where $\mathcal{L}$ is a loss function that computes the average cross-entropy loss between the predicted sign link values and the ground truths.

In the synthetic adjacency matrix $\mathbf{A}'$, many edges have very small weights, making it difficult to assign a clear sign to them. This ambiguity could cause errors in sign link prediction tasks. To address this problem, a sparsification technique is adopted, which tries to generated a sparse synthetic matrix for edges by removing these edges with small weights. The sparsification can effectively prevent incorrect message passing during GNN training. In this sparsification, the edge removing strategy is formulated as follows:

$$a'_{ij} = \begin{cases} 0, & -\delta < a'_{ij} < \delta \\ a'_{ij}, & \text{others} \end{cases} \tag{4}$$

where δ represents the given threshold of edge weight for removing edges.

Graph Optimization

The graph optimization tries to optimize the $\mathbf{X}'$ of condensed graph and the ψ of $\mathrm{GEN}\psi$, so as to obtain the optimal solution for the condensed graph. This optimization step is achieved by three key components: gradient matching, structure-balanced matching, and feature smoothing.

Gradient Matching. In GNNs, node features are propagated and updated through edges during training, and gradients in GNNs reflect the importance and signs of edges in this process. However, gradient computation among nodes is time-consuming. In view of this, we introduce gradient matching (Jin et al. 2022) for alignment at the label level to retain task-related information, which is defined as follows:

$$\mathcal{L}_{gm} = \frac{1}{C}\sum_{i=1}^{C} \mathrm{D}\left(\nabla_\theta \mathcal{L}\left(\hat{\mathbf{Y}}'_i, \mathbf{Y}'_i\right), \nabla_\theta \mathcal{L}\left(\hat{\mathbf{Y}}_i, \mathbf{Y}_i\right)\right) \tag{5}$$

where C is the number of label categories, $\mathrm{D}(\cdot)$ represents a cosine distance function, and $\nabla_\theta \mathcal{L}(\cdot)$ denotes the gradient of a certain class's label data on the GNN.

Structure-balanced Matching. Different from ordinary graphs, signed graphs have conflicting links and structural balance. In this case, to preserve both conflicting relationships and structural balance in condensed graphs, we propose a structure-balanced matching approach under the mathematical graph homomorphism framework.

Definition 1 (Homomorphism of Signed Graph) (Lovász 2012; Naserasr, Rollová, and Sopena 2015; Naserasr,

Sopena, and Zaslavsky 2021). *A homomorphism of two signed graphs $\mathcal{G}$ and $\mathcal{S}$ is denoted as $f : \mathcal{G} \to \mathcal{S}$, such that for any edge $(v_i, v_k) \in \mathcal{E}$, we have $(f(v_i), f(v_k)) \in \mathcal{E}'$ and $\pi(v_i, v_k) = \pi'(f(v_i), f(v_k))$, where π and π' are functions representing the sign of an edge.*

Graph homomorphism enables to preserve sign properties and adjacency relationships during graph condensation. This preservation can be captured by minimizing the following objective function:

$$\min_{f} \sum_{(v_i,v_k)\in\mathcal{E}} \mathcal{L}(\pi(v_i, v_k), \pi'(f(v_i), f(v_k)) \tag{6}$$

where $\mathcal{L}(\cdot)$ is a loss function that measures the discrepancy between the edge signs in the two graphs.

Based on the graph homomorphism, we propose the structure-balanced matching, taking the structural balance theory (Heider 1946; Cartwright and Harary 1956) into consideration.

Definition 2 Structure-balanced Matching. *Given a node v_i in the original graph $\mathcal{G}$ and its corresponding node v'_j in the condensed graph $\mathcal{G}'$. For every positive neighbor $v_k \in \mathcal{N}_i^+$ of v_i, there exists a corresponding positive neighbor $v'_u \in \mathcal{N}_j^+$ of v'_j such that the similarity between v_k and v'_u is maximized, ensuring the preservation of positive associations. Additionally, for each $v_k \in \mathcal{N}_i^+$, the similarity between v_k and any negative neighbor $v'_h \in \mathcal{N}_j^-$ is minimized, thus maintaining the structural conflict between positive and negative associations in the condensed graph.*

With the structure-balanced matching, the objective function in Eq. (6) for preserving conflicting structures and structural balance in condensed graphs is redefined as follows:

$$\min_{\mathcal{S}} \sum_{\substack{v_i\in\mathcal{V}\\ v'_j\in\mathcal{V}'}} p_{ij}\mathcal{L}(\mathcal{N}_i, \mathcal{N}_j) \tag{7}$$

where p_{ij} represents the similarity probability between nodes v_i and v_j, and it is computed as follows:

$$p_{ij} = \frac{\exp\left(z_i{}^{\mathrm{T}} z'_j\right)}{\sum_{k=1}^{n} \exp(z_i{}^{\mathrm{T}} z'_k)} \tag{8}$$

where z_i and z'_j are node embeddings obtained from the GNN with GNN_θ.

$\mathcal{L}(\mathcal{N}_i, \mathcal{N}_j)$ measures the differences between the neighbor sets of nodes in the original graph and the condensed one. Specifically, it is decomposed into two components that handle the matching of positive and negative neighbors separately. For $v_k \in \mathcal{N}_i^+$, the score for the structure-balanced matching among the triplet (v_k, v'_u, v'_h), $v'_u \in \mathcal{N}_j^+$, and $v'_h \in \mathcal{N}_j^-$, is computed as follows:

$$\mathrm{C}^+\left(v_k, v'_u, v'_h\right) = \max\left[\left(\mathrm{d}\left(v_k, v'_u\right) - \mathrm{d}\left(v_k, v'_h\right)\right), 0\right] \tag{9}$$

where $\mathrm{C}^+(\cdot)$ maximizes positive neighbor matching while minimizing matching between positive and negative neighbors to preserve structural balance. $\max(\cdot)$ penalizes triplets that violate the structure-balanced match definition. $\mathrm{d}(\cdot)$ is a function describing the similarity between two nodes, with a smaller value indicating a higher similarity:

$$\mathrm{d}\left(v_i, v_j\right) = -\log \mathrm{sigmoid}(z_i{}^{\mathrm{T}} z_j) \tag{10}$$

Similarly, for $v_l \in \mathcal{N}_i^-$, the structural balance match among the triplet (v_l, v'_u, v'_h) is computed as follows:

$$\mathrm{C}^-\left(v_l, v'_u, v'_h\right) = \max\left[\left(\mathrm{d}\left(v_l, v'_h\right) - \mathrm{d}\left(v_l, v'_u\right)\right), 0\right] \tag{11}$$

Note that, it is time-consuming to compute the structural balance score for all positive and negative neighbors of nodes in $\mathcal{V}'$. To reduce the computation, we propose a negative sampling strategy, which only samples a small number m of nodes from the positive and negative neighbors of v'_k, forming set $\mathcal{T}_k = \left\{(v'_u, v'_h) \,|v'_u \in \mathcal{N}_k^+, v'_h \in \mathcal{N}_k^-\right\}$. In this case, the loss function $\mathcal{L}_{sm}$ for the structure-balanced matching can be computed as follows:

$$\begin{aligned}\mathcal{L}_{sm} = &\frac{1}{|\mathcal{E}^+|} \sum_{(v_i,v_j)\in\mathcal{E}^+} \sum_{\substack{k\in\mathcal{V}'\\ (v'_u,v'_h)\in\mathcal{T}_k}} p_{ik}\mathrm{C}^+\left(v_j, v'_u, v'_h\right)\\ &+ \frac{1}{|\mathcal{E}^-|} \sum_{(v_i,v_j)\in\mathcal{E}^-} \sum_{\substack{k\in\mathcal{V}'\\ (v'_u,v'_h)\in\mathcal{T}_k}} p_{ik}\mathrm{C}^-\left(v_j, v'_u, v'_h\right)\end{aligned} \tag{12}$$

where $|\mathcal{E}^+|$ and $|\mathcal{E}^-|$ denote the numbers of positive edges and negative edges in the signed graph, respectively.

Feature Smoothing. The feature smoothing tries to maximally generate difference between the positive edges and negative edges of nodes in $\mathcal{T}_j = \left\{(v'_k, v'_l) \,|v'_k \in \mathcal{N}_j^+, v'_l \in \mathcal{N}_j^-\right\}$, helping to train the feature embedding of nodes in GNNs. The loss function $\mathcal{L}_{fs}$ of feature smoothing can be written as follows:

$$\mathcal{L}_{fs} = \frac{1}{n} \sum_{j\in\mathcal{V}'} \sum_{(v'_k,v'_l)\in\mathcal{T}_j} \max\left[\left(\mathrm{d}\left(v'_j, v'_l\right) - \mathrm{d}\left(v'_j, v'_k\right)\right), 0\right] \tag{13}$$

The proposed SGSGC integrates the gradient matching, structure balance matching, and feature smoothing. In this case, the loss function of SGSGC for signed graph optimization is a combination of $\mathcal{L}_{gm}$, $\mathcal{L}_{sm}$, and $\mathcal{L}_{fs}$, and it is computed as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{gm} + \alpha\mathcal{L}_{sm} + \beta\mathcal{L}_{fs} \tag{14}$$

where the parameters α and β determine the weights of $\mathcal{L}_{sm}$ and $\mathcal{L}_{fs}$ on the $\mathcal{L}_{total}$, respectively.

Learning Process of SGSGC

The node features $\mathbf{X}'$ and parameters ψ of the generation function GEN_ψ are optimized through a bi-level loop.

Initialization. We randomly sample n nodes from the original graph and initialize their attributes as $\mathbf{X}'$. The graph structure $\mathbf{A}'$ is generated by $\mathrm{GEN}\psi$, forming the initial condensed graph $\mathcal{S} = (\mathbf{A}', \mathbf{X}')$. Subsequently, $\mathrm{GEN}\psi$ adopts Kaiming initialization (He et al. 2015) to initialize ψ. Moreover, we employ a sampling strategy from a uniform distribution P_{θ_0} to initialize the parameters θ_0 of GNN_θ with random numbers. This approach is utilized to mitigate overfitting in a certain GNN model.

Algorithm 1: SGSGC for Signed Graph Condensation

Input: Training signed graph data $\mathcal{G} = (\mathbf{A}, \mathbf{X})$
Output: Condensed signed graph data $\mathcal{S} = (\mathbf{A}', \mathbf{X}')$

Initialize $\theta_0 \sim P_{\theta_0}$.
Let $k = 0$.
while $k \leq K - 1$ **do**
  Compute $\mathbf{A}' = \mathrm{GEN}_{\psi}(\mathbf{X}')$
  Compute $\mathcal{L}_{gm} \leftarrow$ Eq. (5)
  Compute $\mathcal{L}_{sm} \leftarrow$ Eq. (12)
  Compute $\mathcal{L}_{fs} \leftarrow$ Eq. (13)
  Compute $\mathcal{L}_{total} \leftarrow$ Eq. (14)
  if $t\%(\tau_1 + \tau_2) < \tau_1$ **then**
    Update $\mathbf{X}'_{k+1} \leftarrow$ Eq. (15)
  else
    Update $\psi_{k+1} \leftarrow$ Eq. (16)
  end if
  Sparsify $\mathbf{A}'$ according to Eq. (4)
  $l = 0$.
  while $l \leq \tau_\theta$ **do**
    Update $\theta_{l+1} \leftarrow$ Eq. (17).
  end while
  Sparsify $\mathbf{A}'$ according to Eq. (4)
end while

Optimization. We adopt a bi-level optimization framework using Stochastic Gradient Descent (SGD) (Robbins and Monro 1951) to compute gradients.

Outer-level Optimization: In this process, the parameters $\mathbf{X}'$ and ψ are alternately updated with τ_1 and τ_2 steps, respectively, where τ_1 and τ_2 are predefined parameters.

More specifically, when ψ is frozen, $\mathbf{X}'$ is updated as follows:

$$\mathbf{X}'_{k+1} = \mathbf{X}'_k - \mu_1 \nabla_{\mathbf{X}'_k} \mathcal{L}_{total} \tag{15}$$

where k is the current update step for graph optimization, μ_1 is the learning rate, and $\nabla_{\mathbf{X}'_k} \mathcal{L}_{total}$ is the gradient of the total loss function $\mathcal{L}_{\text{total}}$ with respect to $\mathbf{X}'_k$.

Similarly, when $\mathbf{X}'$ is frozen, ψ is updated as follows:

$$\psi_{k+1} = \psi_k - \mu_2 \nabla_{\psi} \mathcal{L}_{total} \tag{16}$$

where μ_2 is the learning rate and $\nabla_{\psi} \mathcal{L}_{total}$ is the gradient of $\mathcal{L}_{total}$ with respect to ψ.

Inner-level Optimization: In this process, to ensure the quality of node embeddings, the model GNN_θ is trained using the loss function $\mathcal{L}_{GNN}$. The parameters θ are updated as follows:

$$\theta_{l+1} = \theta_l - \mu_3 \nabla_{\theta_l} \mathcal{L}_{GNN} \tag{17}$$

where μ_3 is the learning rate, and l is the current update step within the GNN training. The number of epochs for training the GNN is τ_θ.

The algorithm framework of SGSGC is shown in Algorithm 1.

Experiments

In this part, to validate the effectiveness, the proposed SGSGC is tested on five large-scale networks, and compared it with eight baseline methods for link sign prediction tasks.

Experimental Settings

Datasets. We conducted experiments on five real-world network datasets from diverse domains, including collaboration, business, and social interactions, to validate the effectiveness of the proposed SGSGC, namely Bitcoin-alpha (Kumar et al. 2018), Wiki (Leskovec, Huttenlocher, and Kleinberg 2010a), Wiki-rfa (West et al. 2014), Slashdot (Leskovec, Huttenlocher, and Kleinberg 2010b), and Epinions (Leskovec, Huttenlocher, and Kleinberg 2010b). To create training and testing sets, we randomly split the positive and negative edges of each dataset in an 8:2 ratio. Moreover, for the signed networks without node properties, we use the feature extraction in SSE (Kunegis et al. 2010) to obtain the top-d eigenvectors of $\mathbf{A}$ as node feature vectors.

Baselines. We compare our SGSGC with eight state-of-the-art algorithms, including three sparsification-based methods: Random, Herding (Welling 2009), and K-Center (Sener and Savarese 2018), one coarsening-based method: Coarsening (Huang et al. 2021), and four graph condensation-based methods: DCG (Zhao, Mopuri, and Bilen 2021), GCond (Jin et al. 2022), GCDM (Liu et al. 2022), and SGDD (Yang et al. 2023) . Details of these algorithms could be found in the related references.

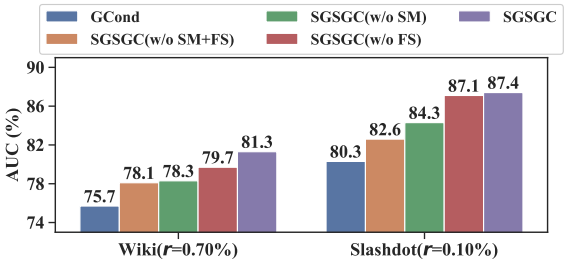


Figure 2: Ablation study results of SGSGC.

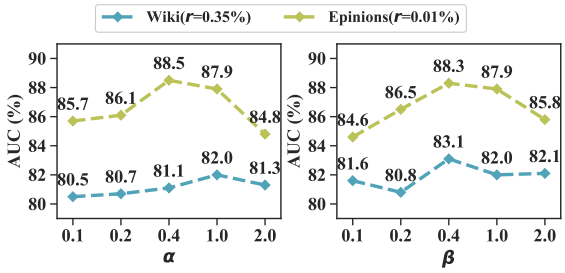


Figure 3: Parameter sensitivity of α and β.

Hyper-parameter setting. For all datasets, the dimensionality of node attribute features in the original graph is fixed at 64. During the condensation process, for all graph condensation methods, the outer loop T is set to 600, and K is set to 20. The update steps τ_1 and τ_2 are set to 20 and 30, respectively. Moreover, in the GNN training steps, τ_θ is set to 15 on the GNN. For GDC, we use a two-layer MLP

Dataset	Ratio(r)	Heuristic methods				Graph condensation					△(%)	Whole
		Random	Herding	K-Center	Coarsening	GDC	GCond	GCDM	SGDD	SGSGC		
Bitcoin-alpha	1.00%	90.2±3.7	91.3±3.9	85.8±1.8	93.5±2.2	94.7±0.6	93.1±4.4	89.8±5.7	93.2±4.3	**95.4±1.1**	↑ 0.73	98.3±0.1
	1.50%	89.9±6.6	90.7±1.6	89.8±3.6	92.7±2.3	95.2±1.0	79.2±6.6	91.4±2.5	94.6±1.9	**96.0±0.3**	↑ 0.84	
	3.00%	90.7±1.5	90.8±1.7	87.1±4.3	93.9±0.7	95.3±1.5	84.1±11.6	94.6±1.3	95.1±1.9	**95.4±0.9**	↑ 0.10	
Wiki	0.35%	73.3±3.6	75.0±0.8	73.8±3.4	67.1±3.3	74.4±2.1	76.5±5.8	73.5±3.1	77.9±3.8	**82.0±2.7**	↑ 5.26	91.3±0.1
	0.70%	71.5±3.0	70.8±2.5	73.4±2.7	73.2±3.1	74.3±1.1	75.7±7.9	74.1±2.0	78.2±4.2	**81.3±2.7**	↑ 3.96	
	1.50%	74.4±1.9	74.8±1.2	74.5±2.5	68.4±3.3	74.4±2.4	75.4±2.3	68.1±9.9	76.7±4.1	**79.1±1.6**	↑ 3.13	
Wiki-rfa	0.50%	78.1±1.3	77.0±3.0	77.8±1.7	84.6±1.6	75.8±4.1	87.2±5.1	74.1±5.0	87.1±3.3	**89.4±2.2**	↑ 2.52	97.1±0.2
	1.00%	81.6±2.1	80.5±1.8	77.1±2.0	88.1±1.3	77.9±2.1	83.7±6.8	75.3±3.7	86.0±6.9	**89.3±1.7**	↑ 1.36	
	2.00%	86.5±1.7	86.2±0.7	84.3±3.3	87.1±0.6	82.0±4.7	87.0±3.1	76.5±2.9	89.9±2.8	**91.5±1.2**	↑ 1.78	
Slashdot	0.02%	81.6±1.4	82.0±0.6	82.9±0.4	47.7±15.4	82.5±0.7	84.9±3.5	79.5±2.9	84.2±3.0	**87.9±1.1**	↑ 3.53	94.4±0.2
	0.05%	82.9±2.2	82.6±0.7	82.4±1.4	74.0±1.9	82.7±1.0	87.0±1.0	79.2±9.1	87.9±1.2	**88.6±0.7**	↑ 0.80	
	0.10%	83.0±1.6	82.6±1.6	82.1±3.3	78.3±1.6	82.6±0.7	80.3±5.1	79.9±3.1	83.4±5.9	**87.4±1.5**	↑ 4.80	
Epinions	0.01%	84.0±4.3	87.7±0.9	86.0±3.1	57.1±19.3	84.4±5.4	86.6±5.1	83.9±4.0	87.8±4.2	**91.2±1.8**	↑ 3.87	97.2±0.2
	0.02%	83.8±2.5	85.1±4.3	82.4±4.2	61.0±12.5	85.5±2.1	83.8±5.6	82.1±4.8	89.3±2.5	**91.3±1.4**	↑ 2.24	
	0.05%	86.3±1.4	85.3±1.8	85.6±1.2	71.8±2.1	86.0±0.9	79.2±6.8	85.8±1.9	85.2±5.2	**91.7±0.7**	↑ 6.26	

Table 1: Comparisons to state-of-the-art methods. SGSGC achieves the best results in all datasets on link sign prediction. Performance is reported as Auc ± standard deviation (%). (Bold: best result, underline: sub-optimal result.) $\triangle(\%)$ denotes the improvements of our method upon the sub-optimal result.

Dataset	Methods	MSGCNN	SSSNET	SGCN	SNEA	Avg. (AUC)	Avg. (std)
Wiki 0.35%	GCond	64.0±9.2	73.2±5.5	76.5±5.8	90.9±2.0	76.2	5.6
	GCDM	51.5±3.8	55.3±8.4	73.5±3.1	88.8±5.0	67.3	5.1
	SGDD	62.9±1.8	64.2±2.5	77.9±3.8	91.2±1.5	74.1	**2.4**
	SGSGC	68.4±2.1	71.4±5.4	82.0±2.7	90.1±2.3	**78.0**	3.1
Wiki-rfa 0.50%	GCond	77.4±2.5	79.4±1.7	87.2±5.1	96.4±0.6	85.1	2.5
	GCDM	54.0±6.3	57.9±5.6	74.1±5.0	95.8±1.2	70.5	4.5
	SGDD	75.1±2.3	79.2±3.4	87.1±3.3	95.6±0.3	84.3	2.3
	SGSGC	81.2±1.6	80.2±2.4	89.4±2.2	96.7±0.4	**86.9**	**1.7**
Epinions 0.02%	GCond	66.7±11.4	74.7±5.9	84.9±3.5	95.0±0.2	80.3	5.3
	GCDM	53.5±7.1	54.5±7.0	79.5±2.9	94.7±0.5	70.6	4.4
	SGDD	64.4±9.4	70.4±5.4	84.2±3.0	94.7±0.2	78.4	4.5
	SGSGC	74.7±2.1	76.4±2.2	87.9±1.1	95.0±0.2	**83.5**	**1.4**

Table 2: Results of different architectures setting. Avg.(AUC) represents the average performance and Avg.(std) stands for the average standard deviation.

with hidden units fixed at 32 to obtain node embeddings. For other graph condensation methods, a two-layer MLP is used to construct GEN_ψ. Specifically, for the small dataset Bitcoin-alpha, the hidden units are set to 128, while for the other datasets, the hidden units are set to 256. We consistently use a two-layer SGCN (Derr, Ma, and Tang 2018) to obtain node embeddings, with hidden units fixed at 32. The task of predicting the sign is treated as a binary classification problem, employing a single-layer MLP to output the probability of a positive sign. The learning rates μ_1, μ_2, and μ_3 are set to 0.0001, 0.0001, and 0.01, respectively.

During testing, we use a unified SWGCN framework with a training epoch of 600 and a learning rate of 0.01 to ensure fairness in evaluation.

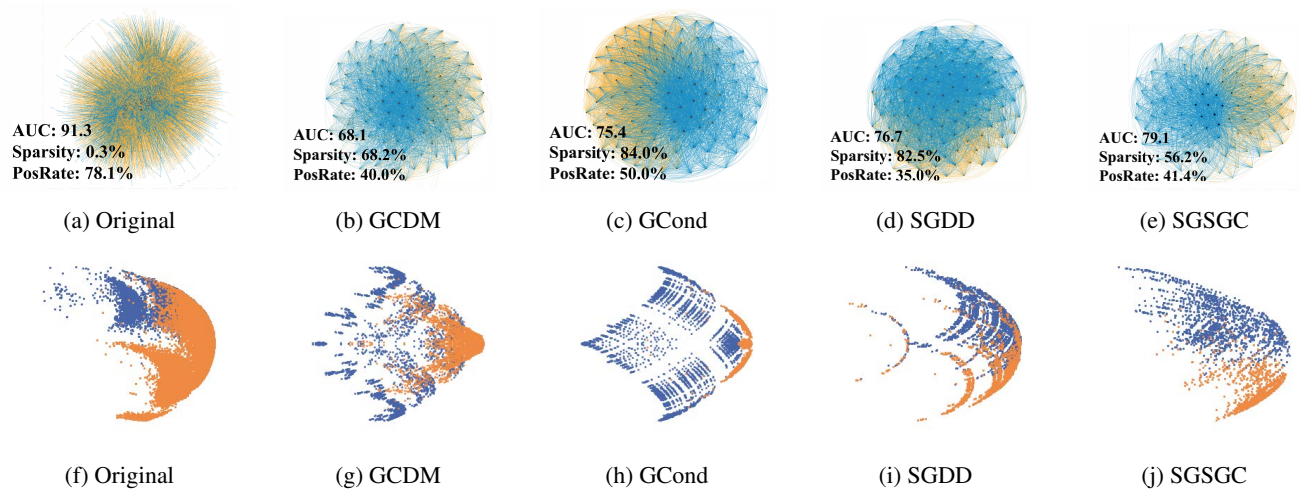
Experimental Evaluation

All baselines are trained and tested with a GNN model for link sign prediction tasks. In the training phase, the synthesized sign graphs are generated using the baseline methods. In the testing phase, the generated synthesized signed graphs are utilized to train the GNN model for link sign prediction. Here, the AUC is chosen as the evaluation metric to assess the model performance.

Evaluation Results and Analysis

Comparison with Baseline Methods. In our experiments, we condense the original graph using five different seeds, conduct each test ten times for all baselines, and record their average AUC values in Table 1. The results indicate that heuristic methods, such as Random, Herding, and K-Center, generally perform well but have lower performance than graph condensation methods. Specifically, on the Bitcoin-alpha dataset with a 3.00% ratio, heuristic methods like Random and Herding achieve AUC scores of 90.7% and 90.8%, respectively. In contrast, graph condensation methods such as GDC and SGSGC achieve significantly higher scores of 95.3% and 95.4%. This demonstrates that graph condensation methods are more effective in capturing the intrinsic graph structure, leading to better link sign prediction performance. On the Epinions dataset with a 0.05% ratio, heuristic methods show AUC scores ranging from 71.8% to 86.3%, while graph condensation methods like SGSGC achieve an AUC of 91.7%, showing a substantial improvement of approximately 6.26%. Among the graph condensation methods, SGSGC consistently outperforms the others in both effectiveness and robustness.

Moreover, GCDM performs worse than other methods, indicating that gradient matching can more accurately retain label-related information compared to distribution matching. On the Wiki and Epinions datasets, SGDD and SGSGC achieve the best results, demonstrating that preserving structural information can enhance condensation effectiveness. For instance, on the Wiki dataset with a 0.35% ratio, SGSGC achieves an AUC of 82.0%, significantly higher than SGDD's 77.9%, highlighting its superior effectiveness.



(a) Original (b) GCDM (c) GCond (d) SGDD (e) SGSGC

(f) Original (g) GCDM (h) GCond (i) SGDD (j) SGSGC

Figure 4: Visualizations of the dataset Wiki with condensing ratio $r = 1.5\%$. (a)-(e) are visualizations of original graph and synthesize one by GCDM, GCond, SGDD, and SGSGC, where PosRate represents the proportion of positive edges. (f)-(j) depict the visualizations of edge embeddings obtained through T-SNE. Edge signs indicated by different colors.

Additionally, SGSGC shows an improvement of 5.26% over SGDD. These results underscore the superiority of SGSGC in link sign prediction tasks across various datasets compared to other baselines.

Different Architectures. The experimental results in Table 2 demonstrate the generative capabilities of SGSGC across various GNN architectures, such as MSGNN (He et al. 2022a), SSSNET (He et al. 2022b), SGCN (Derr, Ma, and Tang 2018), and SNEA (Li et al. 2020). SGSGC consistently performs well across these different architectures, showcasing its robust generalization ability. For instance, on the Wiki-rfa dataset with a 0.50% ratio, SGSGC achieves an average AUC of 86.9% with a low standard deviation of 1.7, highlighting its high effectiveness and stability. These results confirm that SGSGC effectively preserves conflict structures for signed graph condensation, maintaining high performance and stability across different GNN models.

Ablation. To demonstrate the effectiveness of our two operations on the preservation of conflict structures and the sparsification design during the training process, we conduct a series of ablation experiments. Specifically, we perform five sets of experiments on the Wiki and Slashdot datasets for comparisons: SGSGC, SGSGC (w/o SM) without the structure-balanced matching component, SGSGC (w/o FS) without the feature smoothing, SGSGC (w/o SM+FS) without these two operations, and GCond. Figure 2 shows that the designed two operations contribute to the condensation of signed graphs. More specifically, on the Wiki dataset, the sparsification operation results in 3.17% improvement in AUC, and therefore SGSGC achieves 7.40% improvement compared to GCond.

Exploring the sensitivity of α and β. We conduct sensitivity testing experiments on the Wiki and Epinions datasets to assess the impact of parameters α and β on graph condensation performance. As illustrated in Figure 3, we systematically vary the values of α and β, and observe their effects on performance. The results demonstrate that the proposed SGSGC maintains stable performance across a wide range of α and β values, indicating good robustness to these parameters. Specifically, the optimal performance on the Epinions dataset is achieved when both α and β are set to 0.4.

Visualization. Figure 4 shows the visualization of original signed graph and synthesized graphs from GCDM, GCond, SGDD, and SGSGC on the Wiki dataset, with different colors for positive and negative edges. Despite the original graph having more positive edges, the synthesized graphs achieve a balanced distribution. Using T-SNE to visualize the embeddings of positive and negative edges, we explore the relationship between the synthesized and original graphs. The results indicate that GCond, SGDD, and SGSGC effectively differentiate between positive and negative edges, with SGSGC particularly excelling in capturing the original graph's positive and negative link distribution.

Conclusions

In this paper, we introduced a novel Structure Balance and Gradient Matching-Based Signed Graph Condensation (SGSGC) to generate a compact but informative graph, enabling GNNs to achieve performance comparable to training on the original one. SGSGC ensures consistency with conflicting link structures and structural balance through structure-balanced matching, while gradient matching preserves node attributes and conflict links by emulating the original graph's learning trajectory. Additionally, feature smoothing and sparsification enhance robustness by reducing noise and ensuring clear node embeddings. Extensive experiments on five large-scale networks showed the superiority of the proposed SGSGC over eight baseline methods for sign link prediction. In the future work, we will consider the signed graph condensation with privacy preservation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62173236, and in part by the Shenzhen Natural Science Foundation under Grant JCYJ20240813141416022.

References

Batson, J. D.; Spielman, D. A.; Srivastava, N.; and Teng, S. 2013. Spectral sparsification of graphs: theory and algorithms. *Commun. ACM*, 56(8): 87–94.

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

Derr, T.; Ma, Y.; and Tang, J. 2018. Signed Graph Convolutional Networks. In *IEEE International Conference on Data Mining, ICDM 2018*, 929–934.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, 1026–1034.

He, Y.; Perlmutter, M.; Reinert, G.; and Cucuringu, M. 2022a. MSGNN: A Spectral Graph Neural Network Based on a Novel Magnetic Signed Laplacian. In *Learning on Graphs Conference, LoG 2022*, volume 198 of *Proceedings of Machine Learning Research*, 40.

He, Y.; Reinert, G.; Wang, S.; and Cucuringu, M. 2022b. SSSNET: Semi-Supervised Signed Network Clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining, SDM 2022*, 244–252.

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

Huang, Z.; Zhang, S.; Xi, C.; Liu, T.; and Zhou, M. 2021. Scaling Up Graph Neural Networks Via Graph Coarsening. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event*, 675–684.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph Structure Learning for Robust Graph Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 66–74.

Jin, W.; Zhao, L.; Zhang, S.; Liu, Y.; Tang, J.; and Shah, N. 2022. Graph Condensation for Graph Neural Networks. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Kumar, S.; Hooi, B.; Makhija, D.; Kumar, M.; Faloutsos, C.; and Subrahmanian, V. S. 2018. REV2: Fraudulent User Prediction in Rating Platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018*, 333–341.

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

Leskovec, J.; Huttenlocher, D. P.; and Kleinberg, J. M. 2010a. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*, 641–650.

Leskovec, J.; Huttenlocher, D. P.; and Kleinberg, J. M. 2010b. Signed networks in social media. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010*, 1361–1370.

Li, Y.; Tian, Y.; Zhang, J.; and Chang, Y. 2020. Learning Signed Network Embedding via Graph Attention. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 4772–4779.

Liu, M.; Li, S.; Chen, X.; and Song, L. 2022. Graph condensation via receptive field distribution matching. *arXiv preprint arXiv:2206.13697*.

Lovász, L. 2012. *Large networks and graph limits*, volume 60. American Mathematical Soc.

Naserasr, R.; Rollová, E.; and Sopena, É. 2015. Homomorphisms of signed graphs. *Journal of Graph Theory*, 79(3): 178–212.

Naserasr, R.; Sopena, É.; and Zaslavsky, T. 2021. Homomorphisms of signed graphs: An update. *European Journal of Combinatorics*, 91: 103222.

Purohit, M.; Prakash, B. A.; Kang, C.; Zhang, Y.; and Subrahmanian, V. S. 2014. Fast influence-based coarsening for large networks. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, 1296–1305. ACM.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *6th International Conference on Learning Representations, ICLR 2018*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018*.

Welling, M. 2009. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, volume 382 of *ACM International Conference Proceeding Series*, 1121–1128.

West, R.; Paskov, H. S.; Leskovec, J.; and Potts, C. 2014. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2: 297–310.

Yang, B.; Wang, K.; Sun, Q.; Ji, C.; Fu, X.; Tang, H.; You, Y.; and Li, J. 2023. Does Graph Distillation See Like Vision Dataset Counterpart? In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

Zhang, M.; and Chen, Y. 2018. Link Prediction Based on Graph Neural Networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, 5171–5181.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2021. Dataset Condensation with Gradient Matching. In *9th International Conference on Learning Representations, ICLR 2021*.

Zhao, T.; Zhang, X.; and Wang, S. 2021. GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining*, 833–841.