ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230596

Enhancing Link Prediction with Self-Discriminating Augmentation for Structure-Aware Contrastive Learning

Hao-Wei Yang[a], Ming-Yi Chang[b] and Chih-Ya Shen[a;*]

[a]National Tsing Hua University, Hsinchu, Taiwan
[b]Fu Jen Catholic University, New Taipei City, Taiwan
ORCiD ID: Chih-Ya Shen https://orcid.org/0000-0002-0377-7945

Abstract. Link prediction is a crucial research area for both data mining and machine learning. Despite the success of contrastive learning in node classification tasks, applying it directly to link prediction tasks has revealed two major weaknesses, i.e., *single positive sample contrasting* and *random augmentation*, resulting in inferior performance. To overcome these issues, we propose a new contrastive learning approach for link prediction, called *Structure-aware Contrastive Representation Learning with Self-discriminating Augmentation (SECRET)*. Our approach includes a novel data augmentation scheme based on the prediction model itself and takes into account both the contrastive objective and the reconstruction loss, which jointly improve the performance of link prediction. Our experiments on 11 benchmark datasets demonstrate that *SECRET* significantly outperforms the other state-of-the-art baselines.

1 Introduction

Graph Neural Networks (GNNs) receive much research attention recently due to its outstanding performance in various applications [36, 35]. Among the wide application spectrum of GNNs, *link prediction* is one of the most important tasks and has been receiving significant research attention as the connectivity of entities is the most essential component in a network. Application scenarios of link prediction include recommendation, e-commerce, friend recommendation in social networks, and much more [32, 29, 31, 18, 12, 30].

The link prediction problem aims at predicting potential relations or missing connections of two nodes in a graph. To effectively tackle link prediction problems with GNNs, the core issue is to acquire proper representations for either nodes, links, or graphs. On the other hand, one critical issue for constructing a precise link prediction model is that abundant labeled data are required to ensure the performance; however, in real-world scenarios, labeled data are usually scarce and costly. To alleviate this issue, recently, unsupervised contrastive representation learning on graphs [33, 41, 42, 13, 14, 40] has flourished and achieved significant performance improvement. The basic idea of contrastive learning is to maximize the embedding similarity between augmented examples, aiming to capture the invariant signals across two different views [21].

Although achieving promising performance, however, existing contrastive learning methods are usually designed for *node classification* or *graph classification*. Directly applying these methods to link prediction tasks may not result in promising performance. After a careful analysis, we identify two important weaknesses of directly applying contrastive learning to link prediction.

Weakness 1. Single positive sample contrasting. Existing contrastive learning approaches usually concentrate on maximizing the similarity between the representation of the target node and its corresponding representation in the augmented view, but not considering the connection between other nodes. Therefore, the learned representation focuses more on distinguishing each individual of different classes rather than taking the relations of multiple nodes into consideration. This *single positive sample contrasting* may result in inferior results for link prediction problems, since link prediction copes with *connections between node pairs* instead of a *single node*.

Weakness 2. Random augmentation. The augmentation strategy is also critical to the performance; however, existing works usually adopt **random modification** strategies, such as uniform node dropping, uniform attribute masking, which might lead to sub-optimal performance by accidentally corrupting the inherent topological or attributed information.

To address the above critical weaknesses, we propose a new contrastive learning framework for link prediction on graphs, named *Structure-aware Contrastive Representation Learning with Self-discriminating Augmentation (SECRET)*. The proposed *SECRET* includes new components to deal with the weaknesses mentioned above, i.e., *Self-discriminating Augmentation (SDA)* and *Structure-aware Contrastive Representation Learning (SCL)*. Specifically, to address the weakness of *random augmentation*, the proposed *SDA* sets priorities for edges to perform augmentation, which is motivated by our analysis of various augmentation strategies (detailed later). The proposed SDA utilizes our link prediction model itself as a data augmenter to discriminate edges on the graph before performing the augmentation, which not only resolves the problem of *random augmentation* in previous augmentation scheme but also serves as an automatic and systematical strategy to augment a graph.

Then, to address the weakness, *Single positive sample contrasting*, we propose a novel scheme, named *Structure-aware Contrastive Representation Learning (SCL)*, which minimizes the similarity of node representations with intra- and inter-view contrastive terms. In contrast to previous contrastive learning frameworks for node classification [33, 41, 42, 13, 14, 40], to deal with the link prediction tasks, we take the ego network of each individual into consideration as its positive samples, aiming to capture the topological information around an anchor node by making its surrounding neighborhood have similar latent representation in the embedding space.

We evaluate our proposed *SECRET* on 11 benchmark datasets.

* Corresponding Author. Email: chihya@cs.nthu.edu.tw

The experimental results demonstrate that our proposed model achieves significantly better performance, as compared to the state-of-the-art link prediction approaches. In addition, we also show that *SECRET* achieves its superior performance while requiring less resource in terms of time and memory.

The contributions are summarized as follows.

- We propose a new contrastive learning framework for link prediction, *SECRET*. We conduct experiments to justify our data augmentation strategy.
- We propose the novel ideas of *Self-discriminating Augmentation* and *Structure-aware Contrastive Representation Learning* for effective data augmentation and integrating the neighborhood structure in the contrastive views.
- Experiments on 11 benchmark datasets show that our proposed *SECRET* significantly outperforms the other state-of-the-art baselines in terms of prediction performance and efficiency.

The rest of this paper is organized as follows. Sec. 2 discusses the related works. Sec. 3 presents a preliminary analysis of our proposed augmentation strategy. Sec. 4 details the design of our link prediction algorithm and training pipeline. Sec. 5 presents the experimental results. Sec. 6 concludes this paper.

2 Related Works

Link Prediction. Early works to link prediction often make a strong assumption that nodes with certain relations are more likely to be linked. For example, Common Neighbor (CN) measures the intersection of the targets' neighbor node sets; Jaccard index further divides CN by the union size of the two neighbor node sets. These methods often have limited capability in capturing the higher-order structural information, leading to inconsistent performance among different datasets.

Maximum likelihood algorithms [8, 2] often start from organizing the graph's structure with a specific principle and aim at maximizing the likelihood of a target structure. Then, the existence of a link can be predicted by best fitting the structure with maximized likelihood. However, maximum likelihood algorithms often suffer from the high computation overhead, i.e., taking exponential time of iterations for convergence.

The GNN-based link prediction approaches are roughly divided into two categories, i) *node embedding-based approaches* employ a score function as decoder to evaluate a link by its two adjacent nodes' representations, such as Variational Graph Auto-Encoders [16], ARGVA [24], and GIC [19]; ii) *Subgraph classification-based approaches* extract a subgraph with respect to the target link and predict probability of the link with subgraph classification, such as SEAL [39] and WalkPooling [23], where WalkPooling achieves the state-of-the-art performance on multiple benchmarks.

In this paper, we focus on enhancing the model's performance in the following aspects. i) Compared with subgraph classification-based approaches, e.g., [39, 23], we employ contrastive learning along with graph autoencoder, which allows the GNN model to learn more discriminating representations for link prediction. ii) Moreover, our approach avoids the efforts to extract subgraph around the focal link and perform subgraph classification, such as performed in [39, 23]. iii) Compared with node embedding-based approaches, e.g., [16, 1, 24], we observe that with the proposed augmenting technique, *Self-discriminating Augmentation (SDA)*, which modifies the *easier examples* by the decoder, we are able to achieve much better performance.

Contrastive Representation Learning. Unsupervised representation learning attracts much research attention recently, in which contrastive learning [7, 41, 13] gains unprecedented popularity and achieves the state-of-the-art performance in self-supervised learning. The intuition of contrastive learning is to capture the underlying information between data and its augmented view by contrasting the positive and negative samples. The objective can be viewed as maximizing the lower bound of mutual information of input and learned representation [3, 22, 11]. In addition, various contrastive learning approaches adopt different contrastive or data augmentation strategies to boost the performance, such as Deep Graph Infomax [33], MVGRL [10], SelfGNN [14], and GCA [42].

However, unlike the other methods, we aim to utilize the GNN itself, an autoencoder-based model, to serve as the data augmenter to generate the contrastive view for the original graph, based on the accuracy of the predicted probability and trained end-to-end. Furthermore, we aim to integrate the contrastive term along with the autoencoder's generative loss, altogether as the optimization objective, in order to enhance the overall performance of our link prediction model.

3 Problem Formulation and Analysis

Problem formulation – Link Prediction. We denote the undirected input graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ denotes the node set with size N, i.e., $|\mathcal{V}| = N$, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set. We let $X \in \mathbb{R}^{N \times F}$ be the input feature matrix where x_i is the input node feature vector of v_i with dimension F. We further denote the adjacency matrix as $A \in \{0,1\}^{N \times N}$, in which $a_{i,j} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$. In the following, we denote our encoder as f and the learned output embedding from encoder as $Z = f(X, A) \in \mathbb{R}^{N \times F'}$, where F' is the output dimension of encoder f. Since there are two different graph views, we denote the augmented view's adjacency matrix and learned embedding as $A' \in \{0,1\}^{N \times N}$ and $U = f(X, A') \in \mathbb{R}^{N \times F'}$, respectively. Lastly, we denote $\mathcal{N}(i) = \{j | (v_i, v_j) \in \mathcal{E}\}$, as the index set of v_i's neighbors, i.e., adjacent nodes.

The link prediction problem is defined as follows. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{ideal})$, where $\mathcal{E}_{ideal}$ is the ideal edge set, which is composed of all the existing links on $\mathcal{G}$. However, we can only observe an incomplete set of $\mathcal{E}_{ideal}$, denoted as $\mathcal{E}$. To solve the link prediction problem, we aim at finding a predictive function $\mathcal{F}$ that takes $\mathcal{V}, \mathcal{E}$, and X as input and outputs its predicted edge set $\mathcal{E}_{\mathcal{F}}$, such that $\mathcal{E}_{\mathcal{F}}$ is as close to the ideal edge set, $\mathcal{E}_{ideal}$, as possible.

Preliminary analysis. As discussed earlier, one of the two major weaknesses of the previous contrastive learning-based approaches is *random augmentation*. We argue that such a random approach may not be the best strategy for the link prediction task, and this motivates us to take a closer look at different edge augmentation strategies.

Data augmentation is a component of crucial importance with respect to contrastive learning. We argue that a proper augmentation strategy shall retain the important structure and feature information that is critical to model learning. However, the previous contrastive learning works usually adopt a *random modification* strategy [38, 13, 41], which might accidentally corrupt the inherent graph topological information and leads to inferior performance. On the other hand, Zhu et al. [42] propose to modify the graph by first identifying those important edges and node attributes via calculating the centrality of the nodes. After that, unimportant edges and attributes are given higher probability to be dropped. Zhu et al. claim that truncating those pieces of unimportant information forces the model to learn the embedding that is insensitive to corruption on less important nodes and edges. However, calculating the centrality values of nodes

requires excessive computation overhead, which limits its scalability.

On the contrary, we aim at generating an augmented graph via a more systematical and automatic way by employing the GNN-encoder as a data augmenter as well, which automatically determines the edges to be modified. To allow the GNN-encoder to determine the suitable edges to be modified, one important research question arises:

RQ 1. *To improve the model performance, what edges should be modified in the graph augmentation step?*

Previous answers to this research question include *random modification* [38, 13, 41] and *centrality-based metric* [42], which measure the centrality to determine the importance of the edges. However, we argue that there might still be room for improvement, and thus we conduct a preliminary analysis.

We observe that, among all the training edges, some edges can be easily predicted by the link prediction model (called *easy examples* or *easy edges* hereafter), while some are much more difficult ones (called *hard examples* or *hard edges*). For the easy edges, the prediction model shall confidently determine the probability of its existence or absence (with probability clinging to zero or one). However, for those hard examples, the model might fail to accurately predict its existence or even take the wrong side. Therefore, in addition to the notions of *easy* and *hard* edges, we further propose the notion of *prediction disparity*, to quantify how *easy* (or *hard*) an edge is. The *prediction disparity* $\pi_{i,j}$ represents the absolute difference of the existence probability of an edge (v_i, v_j) on the original input graph $\mathcal{G}$ and the prediction result of the same edge from the model, i.e., $\pi_{i,j} = |A^{pred}_{i,j} - A_{i,j}|$, where $A^{pred}_{i,j}$ is the prediction output of the edge existence probability of edge (v_i, v_j). A larger $\pi_{i,j}$ indicates a more significant prediction disparity, i.e., the prediction output is further from the ground truth, considered as a *harder* edge.

Table 1. The performance of various augmentation strategies, where *gain* is the difference against modifying `harder` edges.

	`Easier Edges`	`Easy Edges`	`Hard Edges`	`Harder Edges`	`Random`	`Centrality`
AUC on *Cora* (%)	96.44	96.17	95.60	95.04	95.91	95.27
AUC gain (%)	**+1.40**	+1.13	+0.56	0.00	+0.87	+0.23
AUC on *Wisconsin* (%)	85.07	84.50	79.96	81.90	79.21	78.07
AUC gain (%)	**+3.17**	+2.60	-1.94	0.00	-2.69	-3.83
AUC on *LastFMAsia* (%)	96.63	96.63	96.53	96.32	96.52	96.24
AUC gain (%)	**+0.31**	**+0.31**	+0.21	0.00	+0.20	-0.08

With the notion of prediction disparity in hand, now we compare various strategies to identify the set of edges to be modified on the Cora, Wisconsin, and LastFMAsia datasets [27, 25, 9], which are benchmark datasets widely adopted for evaluating link prediction approaches. The compared strategies include: i) `Random`, which modifies the edges randomly, as in previous contrastive learning approaches [38, 13, 41]; ii) `Centrality`, which selects the edges to modify based on the calculated centrality [42]; In addition, we also propose and compare four augmentation strategies: iii) `Easier Edges`, which modifies the edges with prediction disparity in $[0.0, 0.2]$; iv) `Easy Edges`, which picks the edges with prediction disparity in $[0.0, 0.5]$; v) `Hard Edges` and vi) `Harder Edges`, which modify the edges with prediction disparity in $[0.5, 1.0]$ and $[0.8, 1.0]$, respectively.

We present the Area Under Curve (AUC) [6] of the above six strategies in Table 1 (the Average Precision (AP) of them shows similar results and thus are omitted). The *AUC gain* is calculated based on `Harder Edges`. The results indicate that `Easier Edges` leads to the most significant performance gain, i.e., $+1.4\%$ in Cora, $+3.17\%$ in Wisconsin, and $+0.31\%$ in LastFMAsia, outperforming `Easy Edges`, `Hard Edges`, `Harder Edges`, as well as the widely adopted `Random` [38, 13, 41] and the centrality-based approach, `Centrality` [42]. We present more detailed comparisons in Fig. 2 in Sec. 5. From the results, we observe that modifying the easier edges, i.e., the edges with the smallest prediction disparity, brings the largest performance gain in link prediction. According to the idea of curriculum learning [4, 34], we conjecture that if we modify the hard or harder edges, the model fails to recognize its demerits and tends to confidently make extreme prediction. On the contrary, by adjusting the edges with tiny prediction disparity, the overall prediction difficulty for the model is higher as the easy ones is distracted for itself. Consequently, the model can thus learn more general and robust embedding, leading to superior performance for link prediction.

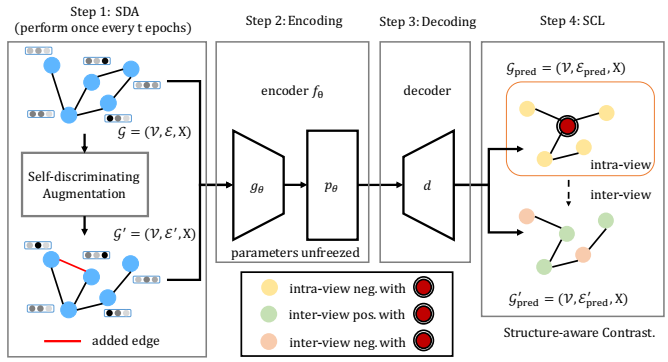


Figure 1. Model overview.

Furthermore, `Easier Edges` significantly outperforms `Random`. One reason is that the important topological information might be accidentally corrupted by `Random`, leading to inferior performance; In addition, `Easier Edges` also outperforms `Centrality`, because `Centrality` determines the importance of edges by the degree of its two adjacent nodes. Although this strategy retains edges from highly connected nodes, it might still corrupt the structural information in the sparser region.

Our answer to RQ 1. Based on the above analysis, instead of modifying the edges based on centrality [42] or randomly [38, 13, 41], *we argue that modifying the edges with small prediction disparity would be more promising for link prediction tasks.*

4 Algorithm Design

Current state-of-the-art link prediction algorithm [39, 23] solve link prediction by subgraph classification. Despite the significant performance they achieved, calculating information matrix [39] or walkprofiles [23] is still required. Therefore, to avoid the burden of constructing subgraph and extracting heuristic information, we propose *SECRET* that incorporates a novel data augmentation scheme, *SDA*, and structure-aware contrastive term, *SCL*. Our proposed model does not make prediction via subgraph classification; instead, it is an efficient dot-product decoder that reconstructs the whole edge set directly.

The proposed *SECRET* addresses the two major weaknesses identified above as follows. To deal with the **random augmentation** problem, our proposed *SDA* generates an augmented view of original input graph via the auto-encoder itself by distinguishing those edge instances with lower prediction disparity, providing an auto-

matic yet systematical augmentation strategy; On the other hand, to cope with the issue of **single positive sample contrasting** in previous node classification-based graph contrastive learning schemes, our proposed *SCL* takes the ego network of each anchor node as its positive samples, making nodes in the neighborhood structure attain representations that stay closer in the embedding space, and encodes topological information the ego network to achieve superior performance in link prediction.

The advantage of our proposed model are four-fold. i) Compared with aforementioned link prediction algorithms, we do not make prediction via subgraph classification but by an efficient dot-product decoder instead. ii) Our proposed model does not extract heuristic information but employs a GNN-based encoder to aggregate information. iii) Our data augmentation scheme is systematical yet simple, saving us from random modification or calculating importance scores, but encouraging our model to learn from harder instance to produce robust embedding. iv) We adopt a contrastive term that considers the neighborhood structure between different graph views. By maximizing the agreement of positive samples, we capture the structural information.

4.1 Overall pipeline of SECRET

To have a better picture of *SECRET*, we illustrate the overall pipeline in Fig. 1, and the details of the proposed data augmentation scheme, *SDA*, is shown in Appendix A in [37]. Specifically, in step one, i.e., *SDA*, we generate an augmented graph once in t training epochs, where t is considered as a hyperparameter in our framework. That is to say, we treat the data augmenter as a clone of the prediction model that updates its parameters with prediction model's parameters every t epochs. By adopting t as an augmentation interval, we are able to stabilize the training procedure without deteriorating the performance, as will be shown in the experiments. During the augmentation, the encoder always takes the original graph $\mathcal{G}$ as its input, and a dot-product decoder then reconstructs all the edge instances on the graph based on the output representations. Once we acquire the predicted results from the decoder, we leverage the proposed *modification disparity bound* b and *modification ratio* k along with the *prediction disparity* (as mentioned in previous section) to identify a set of edges to be modified. By flipping these edge instances in the set, we generate an augmented view of the original graph to be further used in the upcoming steps.

In step two, i.e., *encoding*, the original graph $\mathcal{G}$ and the previously generated augmented view graph $\mathcal{G}'$ are input into our GNN-based encoder and generate node embedding Z and U, respectively. In step three, i.e., *decoding*, we leverage a dot-product decoder that reconstructs all the edge instances on the original graph view and augmented graph view based on Z and U, and obtain two predicted graphs $\mathcal{G}_{pred}$ and $\mathcal{G}'_{pred}$, respectively. In step four, i.e., *SCL*, we leverage the proposed intra-view and inter-view contrastive scheme on the two graph views and incorporate a reconstruction loss along with the contrastive loss to optimize the GNN-based encoder. For the detailed pseudo code of the overall pipeline, please refer to Algorithm 1 in Appendix B in [37].

4.2 Algorithm Description

Step 1. Self-discriminating Data Augmentation (SDA). Previous works of contrastive learning on graphs usually adopt a uniform modification strategy. In our approach, according to the observation in the preliminary analysis, we propose to generate the augmented view via self-discriminating data augmentation that tends to retain those edges with larger prediction disparity while modifying those edges which can be easily and correctly predicted. The overall augmentation scheme is shown in Appendix A in [37]. Specifically, the augmentation pipeline is composed of the following steps.

i) We input the original graph into our *encoder* f to generate the latent representation Z for decoder. Note that the model is freezed during data augmentation, i.e.,

$$Z = f_\theta^{freezed}(X, A). \tag{1}$$

ii) The dot-product decoder d takes the latent representation Z as input and generates a reconstructed adjacency matrix A_{pred}, where $a_{pred_{i,j}} \in A_{pred}$ is the predicted probability of edge existence between node i and node j.

iii) In the augmentation module, we calculate the absolute difference of the reconstructed adjacency matrix and the original graph. That is, $\pi = abs(A_{pred} - A)$, where $abs(\cdot)$ denotes the function that returns a matrix composed of element-wise absolute value of its input. Here, the $N \times N$ matrix π is the prediction disparity matrix where the element $\pi_{i,j}$ represents the prediction disparity of edge connecting nodes i and j.

To determine the modified edges, we utilize two thresholds, i.e., the modification ratio k and modification disparity bound b. The modification ratio k is the upper bound for the ratio of modified edges to all the edges, and the modification disparity bound b is the upper bound for prediction disparity of modified edges. We first employ the *modification disparity bound* b to extract a set of candidate edges denoted as S by

$$S = \{(v_i, v_j) \mid \pi_{i,j} \leq b,\ i \neq j\}, \tag{2}$$

where all elements in S has prediction disparity less than b. Noted that we do not consider self-edges in S. We then use the *modification ratio* k to determine the maximum number of modified edge instances m by $m = \lfloor k \times N \times (N\text{-}1) \rfloor$.

iv) Finally, with the maximum number of modified edge instances m, we sort the set S by the edge prediction disparity in ascending order and obtain a sorted list of edge instances, $S_{sorted} = [e_1, e_2, ...]$, where e_i is the edge instance with i-th smallest prediction disparity in S. We take the first m edges in the sorted list S_{sorted} as our modified edge instances.

$$\textit{modified edges} = \{e_i \mid i \leq m,\ e_i \in S_{sorted}\}. \tag{3}$$

Steps 2 and 3 – Encoding and decoding. We aim to reconstruct the edge set $\mathcal{E}_\mathcal{F}$ to be as similar to the ideal edge set, $\mathcal{E}_{ideal}$, as possible. Here, we consider all positive and negative edges; therefore, we aim at directly reconstructing an adjacency matrix $A_\mathcal{F} \in \{0, 1\}^{N \times N}$ by our prediction model $\mathcal{F}$. Similar to VGAE [16], we assume that all the edges are reconstructed by some process with an unobservable continuous random variable Z. From the perspective of coding theory, as stated in VAE [15], the random variable Z can be viewed as a latent variable that generates edges A. Therefore, we refer to an inference model $q(Z|X, A)$ as our probabilistic *encoder* f parameterized by θ with two-layer GNN g_θ followed by an MLP with batch normalization using as a projection head (denoted as p_θ), as defined below.

$$q(Z|X, A) = \prod_{i=1}^{N} q(z_i|X, A), \tag{4}$$

where z_i denotes the latent variable corresponding to node i and here, as in VGAE and VAE [16, 15], we let the distribution $q(z_i|X, A)$ be

a multivariate normal distribution with diagonal covariance,

$$q(z_i|X,A) \sim \mathcal{N}(\mu_i, diag(\sigma_i^2)), \tag{5}$$

where μ is the mean matrix from the encoder's output and σ is the standard deviation matrix output from f_θ.

Then, we refer to the generative model $p(A|Z)$ as our probability decoder, d, since we produce a probability distribution of A given latent Z. Formally, we define the generative model as follows,

$$p(A|Z) = \prod_{i=1}^{N}\prod_{j=1}^{N} p(A_{ij}|z_i, z_j), \tag{6}$$

where A_{ij} is the element of A in the i-th row and the j-th column. The generative model is a dot-product followed by a logistic sigmoid function with respect to the corresponding latent variable z_i and z_j, denoted as $p(A_{ij}|z_i, z_j) = sigmoid(z_i \cdot z_j)$.

Finally, we optimize the parameter θ in order to maximize the likelihood $p(A)$ by maximizing the variational lower bound [16], i.e., minimizing $\mathcal{L}_{recon}$, as follows.

$$\mathcal{L}_{recon} = -\mathbb{E}_{q(Z|X,A)}[\log p(A|Z)] + KL[\, q(Z|X,A) \| \, p(Z)\,], \tag{7}$$

where $KL[q(\cdot)\|p(\cdot)]$ is the Kullback-Leibler divergence between two distributions.

Step 4. Structure-aware Contrastive Learning (SCL). In each training iteration, after getting the output embedding Z and U from two different graph views in previous steps, we employ our proposed Structure-aware Contrastive Loss, which includes two terms, *Intra-view Contrastive Loss* and *Inter-view Contrastive Loss*, to enforce the model to encode the structural information as well as to maximize the representation agreement of positive samples between views.

It is worth noting that, we employ two split terms of contrastive loss, *intra-view* and *inter-view*, instead of blending all the positive and negative samples within the same graph view or different graph views altogether. This modularization helps us distinguish nodes representations on the same graph in Intra-view Contrastive loss. In addition, we can take nodes on the ego network of each anchor node to be the positive samples of it in another split-out contrastive view, i.e., Inter-view Contrastive Loss, individually dealing with the relation between original and augmented graphs' latent representations.

Intra-view Contrastive Loss. The *Intra-view Contrastive Loss* focuses on distinguishing the node representations on a single graph as we only consider to contrast the representations *within* one single graph view. For any focal (anchor) node, $v_i \in \mathcal{V}$, in the original graph view, we contrast its representation z_i with all the other nodes' representations, z_j, with $v_j \in \mathcal{V} \setminus \{v_i\}$. The contrastive loss with respect to $v_i \in \mathcal{V}$ is formulated as follows,

$$\mathcal{L}_{intra}^{ori}(v_i) = -\log \frac{e^{\theta(z_i,z_i)/\tau}}{\sum_{j=1}^{N} e^{\theta(z_i,z_j)/\tau}}, \tag{8}$$

where τ is a temperature hyperparameter used in previous works [7, 41], and $\theta(\cdot,\cdot)$ is the function that calculates the cosine similarity of input embeddings, θ: $\mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$. For $v_i \in \mathcal{V}$ on the augmented graph view with embedding U, the *Intra-view Contrastive Loss* is similarly formulated as

$$\mathcal{L}_{intra}^{aug}(v_i) = -\log \frac{e^{\theta(u_i,u_i)/\tau}}{\sum_{j=1}^{N} e^{\theta(u_i,u_j)/\tau}}. \tag{9}$$

Finally, the overall *Intra-view Contrastive Loss* takes the average among all the N nodes in $\mathcal{V}$ and is formulated as

$$\mathcal{L}_{intra} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_{intra}^{ori}(v_i) + \delta\mathcal{L}_{intra}^{aug}(v_i)), \tag{10}$$

where δ is the weight parameter for augmented graph.

Inter-view Contrastive Loss. The *inter-view contrastive loss* focuses on the contrasting representations *between* the original and augmented graphs. We further consider the relation between connected nodes by taking the neighborhood structure around the focal node into account. More specifically, for $v_i \in \mathcal{V}$, the positive samples for z_i are defined to be not only the corresponding node representation u_i on augmented graph but also the representations of v_i's neighbors, i.e., $v_j \in \mathcal{N}(i)$. Taking these neighbor nodes as positive samples, we further encode the structural information around the focal node, and the *Inter-view Contrastive Loss* is formulated as

$$\mathcal{L}_{inter}(v_i) = -\log \frac{\sum_{j}^{j\in i\cup\mathcal{N}(i)} e^{\theta(z_i,u_j)/\tau}}{\sum_{j=1}^{N} e^{\theta(z_i,u_j)/\tau}}. \tag{11}$$

Finally, we take the average among all the N nodes in $\mathcal{V}$ and present the overall Inter-view Contrastive Loss:

$$\mathcal{L}_{inter} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{inter}(v_i). \tag{12}$$

Overall Contrastive Loss. With Intra-view Contrastive Loss and Inter-view Contrastive Loss defined in Eqs. 10 and 12, respectively, the overall contrastive loss for SCL is defined as

$$\mathcal{L}_{SCL} = \beta\mathcal{L}_{intra} + \gamma\mathcal{L}_{inter}, \tag{13}$$

where β and γ are hyperparameters for the weights of Intra-view Contrastive and Inter-view Contrastive losses, respectively. Note that we denote the term *contrastive objective*, $\mathcal{J}_{SCL}$, as our target to maximize in terms of SCL. In contrast, $\mathcal{L}_{SCL}$ is the *contrastive loss* to be minimized during model optimization with respect to SCL.

Model training. We train our prediction model jointly with the above-mentioned reconstruction loss, Intra-view Contrastive Loss, and Inter-view Contrastive Loss to learn node latent representation for link prediction. Specifically, the overall optimized loss function can be formulated as

$$\mathcal{L} = \alpha(\mathcal{L}_{recon}^{ori} + \delta\mathcal{L}_{recon}^{aug}) + \mathcal{L}_{SCL}, \tag{14}$$

where α is a hyperparameter that controls the balance between reconstruction loss and SCL loss, and δ is the weight parameter for the augmented graph.

5 Experimental Results

The codes, models, and other reproducibility materials could be found at [37].

5.1 Settings

Datasets. We conduct experiments on 11 benchmark datasets with node attributes, including four citation networks: *Cora*, *Cora_ML*, *Citeseer*, *PubMed* [27, 5]; Three webpage datasets: *Cornell*, *Texas*, and *Wisconsin* [25]; One Wikipedia network *Squirrel* [26, 20]; Two co-purchase network *Amazon Photo* and *Amazon Computers* [28]; One music website *LastFMAsia* [9]. The statistics of the datasets are summarized in Appendix C in [37].

Training setting. For data splits, we follow the experimental protocol as in [16, 23], which splits the positive edges into three parts, 10% for testing, 5% for validation, and the rest for training. We perform 10 random splits on each dataset and report the average and standard deviation of the performance. We implement our model via Pytorch 1.11.0, employ Adam as our optimizer, and set the learning rate as 0.001 with 1000 training epochs. We adopt APPNP [17] as

Table 2. Link prediction accuracy evaluated by AUC (%) on 11 datasets. *Best* results are in **bold**, and *runner-ups* are underlined.

	Cora	Citeseer	PubMed	Cora_ML	Cornell	Texas	Wisconsin	Squirrel	Amazon Photo	Amazon Com.	LastFMAsia
GCA (WWW'21)	91.72±1.37	80.50±5.23	94.46±0.75	83.98±2.70	57.19±21.35	65.78±13.79	71.82±8.52	94.84±0.25	90.90±1.02	91.84±1.18	93.04±0.88
VGAE (NIPS'16)	91.98±0.54	91.21±1.14	96.51±0.14	94.08±0.47	70.59±9.03	73.71±9.29	75.05±6.88	97.54±0.05	96.35±0.89	96.39±0.18	94.56±1.13
VGNAE (CIKM'21)	95.37±0.11	96.30±0.34	97.63±0.03	95.57±0.56	34.81±3.39	61.37±5.33	81.80±1.36	96.32±0.01	96.70±0.02	95.81±0.10	94.41±0.24
ARGVA (IJCAI'18)	92.45±1.11	91.71±1.38	96.62±0.12	95.24±0.48	81.73±4.82	68.05±8.29	75.69±7.91	97.12±0.20	97.21±0.17	96.49±0.14	94.65±0.36
GIC (TechRep'20)	93.68±0.59	95.03±0.65	93.00±0.36	94.04±0.74	63.32±7.47	65.43±10.39	74.74±6.28	94.15±0.27	95.09±0.35	93.65±0.26	94.03±0.29
SEAL (NIPS'18)	79.49±0.74	78.98±1.33	90.21±0.90	85.18±0.89	73.49±5.56	81.05±2.92	76.71±2.50	98.03±0.07	97.73±0.09	97.09±0.10	90.61±0.49
WP-only (ICLR'22)	93.29±0.88	88.42±1.28	97.21±0.23	93.39±0.90	78.72±8.03	74.02±6.39	77.94±10.15	97.02±0.06	96.79±0.22	96.03±0.24	94.98±0.70
GIC+WP (ICLR'22)	95.90±0.50	95.94±0.53	**98.72±0.10**	94.99±0.70	80.69±7.25	74.49±6.85	82.27±6.27	97.98±0.04	96.74±0.19	96.48±0.35	95.42±0.57
Ours	**96.18±0.27**	**97.53±0.17**	98.31±0.34	**96.80±0.18**	**87.51±2.34**	**83.13±4.71**	**85.37±2.51**	**98.26±0.07**	**98.34±0.11**	**97.67±0.08**	**96.59±0.17**

our GNN backbone. The weight decay is set to 5×10^{-4} and dropout rate is set to 0.3. Generally, the hidden dimension for representation is set to 256 and the output dimension of encoder is set to 64 across all datasets. However, the hidden dimension for webpage datasets (*Cornell*, *Texas*, and *Wisconsin*) are adjusted to be smaller due to the smaller scale of these datasets.

Baselines and settings. We compare our model with 8 state-of-the-art baselines, including: i) GCA [42], a contrastive representation learning approach; four unsupervised GNN-based models: ii) VGAE [16], iii) VGNAE [1], iv) ARGVA [24], v) GIC [19]; two link prediction models based on subgraph classification: vi) SEAL [39], vii) WalkPooling [23], where we adopt two variations of it, which are the combination of GIC and WalkPooling (GIC+WP) and WalkPooling itself (WP-only). The detailed descriptions of the baselines and experiment settings can be found in Appendix C in [37].

Performance metrics. We follow previous works and measure the Area Under Curve (AUC) [6]. A higher value of AUC indicates better performance.

5.2 Performance Evaluation

Link prediction performance. Table 2 presents the AUC of all the approaches. Numbers marked in bold indicate the *best performance* among all 9 approaches. From Table 2, the results indicate that by leveraging our proposed modules, *Self-discriminating Augmentation (SDA)* and *Structure-aware Contrastive Learning (SCL)*, our proposed method achieves promising performance on link prediction among various datasets. Compared to another contrastive representation learning method, GCA, our method is more stable and achieves superior performance across all 11 datasets in both AUC.This indicates that our *Self-discriminating Augmentation* works effectively on link prediction tasks, and our proposed *Structure-aware Contrastive Learning* jointly trained with reconstruction loss indeed enhances the performance.

Compared with autoencoder-based methods, i.e., VGAE, VGNAE, and ARGVA, our approach also achieves better performance in most benchmark datasets. Compared with GIC, the clustering-based unsupervised representation learning baseline, our performance outperforms GIC on all 11 datasets in both AUC. Compared with the state-of-the-art link prediction baselines, i.e., subgraph classification-based methods, SEAL, WP and WalkPooling equipped with GIC, GIC+WP, we are also able to achieve better performance.

Ablation Study. To show the effectiveness of the proposed *SDA* and *SCL*, we conduct an ablation study on the Cora dataset, and the results are presented in Table 3. The term w/o SDA denotes not using *SDA* but a random edge modification. The term w/o SCL-intra denotes not using the Intra-view Contrastive Loss $\mathcal{L}_{intra}$, while w/o SCL-inter denotes not using the Inter-view Contrastive Loss $\mathcal{L}_{inter}$, and w/o SCL denotes not using both. The improvements on AUC by equipping these modules indicate the effectiveness of the proposed SDA and SCL in link prediction. Specifically, if both SDA and SCL are equipped, the performance increases by 1.39% in AUC.When both Intra-view and Inter-view Contrastive Losses are employed, the performance of our model boosts by 1.26% in AUC.

Sensitivity tests – k and b. We aim to investigate the relations between our augmentation strategy and the overall performance. Here, k is the modification ratio and b is the modification disparity bound. Note that we take the modification disparity bound b as an interval (i.e., 0.1 stands for modifying edges with disparity in $[0.0, 0.1)$, 0.2 stands for modifying edges with disparity in $[0.1, 0.2)$, etc.) here instead of an upper bound to prevent modifying the whole graph when b is set to 1.0. This is also because we aim at observing the performance change when edges with different *intervals* of disparity are modified.

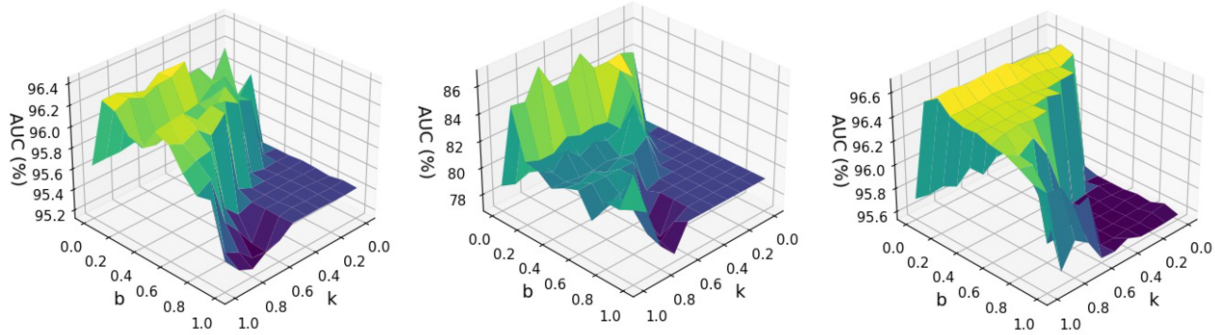
The results are shown in Fig. 2. As we claimed in the preliminary analysis, we observe a performance drop when increasing b, i.e., comparing to modifying those hard edges, modifying those easy edges obtains a higher prediction accuracy. We thus claim that our proposed model can discriminate the hard samples and easy samples and focus on harder ones as the simpler ones are distracted by our augmentation scheme. Also, it is worth noting that the influence of k is reduced when b is set smaller (modifying easy/easier edges) as the performance retains high with various k, indicating that modifying more easy samples does not rapidly corrupt the original graph's inherent information.

Furthermore, if we modify a bunch of hard/harder edges, the performance arrives at its valley bottom as observed when k and b are set higher with values around 1.0. On the other hand, we observe a steep rise in AUC when adjusting b from 0.0 to 0.1, apparently indicating the effectiveness of our augmentation scheme; However, if the modification ratio k is not bounded and gradually increases, the performance would experience another steep drop, indicating that modifying too many edges will corrupt the original graph's structural information and thus there shall be a budget for numbers modification edges.

Table 3. Ablation study with or without SDA and SCL on Cora.

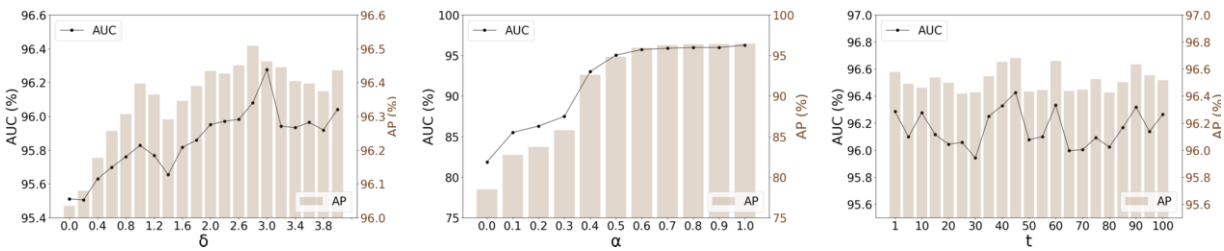
	AUC (%)	AUC Gain (%)
SECRET w/o SDA & SCL	94.89	0.00
SECRET w/o SCL	95.02	+0.13
SECRET w/o SCL-intra	95.65	+0.76
SECRET w/o SCL-inter	95.77	+0.88
SECRET w/o SDA	95.91	+1.02
SECRET	96.28	+1.39

Sensitivity tests – δ, α, and t. The parameter δ is the weight parameter for the augmented graph's loss. The results are shown in Fig. 3(a). The performance ascends as δ increases in both AUC, indicat-



(a) AUC on Cora. (b) AUC on Wisconsin. (c) AUC on LastFMAsia.

Figure 2. Performance of adjusting k and b.



(a) Varying δ. (b) Varying α. (c) Varying t.

Figure 3. Performance of adjusting δ, α, and t.

ing that considering an augmented along with the original graph indeed improves the prediction accuracy. As the value of δ reaches 1.0, the performance begins to stabilize and achieves optimal accuracy at around 3.0.

The parameter α is the weight parameter for reconstruction loss. Here, we vary its value in range $[0.0, 1.0]$ with a step size of 0.1 to show the importance of reconstruction loss to the overall optimization scheme. The result is shown in Fig. 3(b). We observe that the reconstruction loss $\mathcal{L}_{recon}$ is of crucial importance to the whole optimization scheme as both AUC and AP experience a steep drop to around 80% when α decreases. On the other hand, the performance gain become unnoticeable gradually when α is set to around 0.5.

The augmentation period t is the period for re-generating a new augmented graph, i.e., update the augmented graph. Here, we vary its value in the range $[1, 100]$ with a step size of 5 epochs. The results are shown in Fig. 3(c). Unlike the previous two parameters δ and α, the influence of the augmentation period is inconspicuous when the augmented graph is re-generated in various period and the prediction performance steadily achieves 96.0% in AUC. and 96.4% in AP.

Efficiency tests. We also compare the efficiency in terms of space and time with other baselines on Cora and employ the same parameters reported in the experiment setting section. The results are presented in Table 4. The host memory usage for two subgraph classification-based methods, `WP-only`, `GIC+WP`, is generally higher than representation decoder-based methods, where `WP-only` requires around 8,000 MiB in host memory and around 2,300 MiB in GPU memory, `GIC+WP` requires around 4,950 MiB in host memory and around 2,393 MiB in GPU. In summary, our approach achieves the best prediction performance (i.e., an 96.18% AUC) agasint all these compared approaches while requiring much less run time compared to `WP-only` and `GIC+WP`.

Table 4. Efficiency tests on Cora.

	Run Time (Sec)	AUC (%)	GPU Mem. Usage (MiB)	Host Mem. Usage (MiB)
`GCA`	41.15	91.72	1,519	3,273
`VGAE`	36.54	91.98	2,107	4,911
`VGNAE`	11.69	95.37	1,905	4,355
`ARGVA`	38.07	92.45	1,903	4,346
`GIC`	12.72	93.68	2,067	4,692
`SEAL`	223.41	79.49	1,213	3,765
`WP-only`	932.40	93.29	2,313	8,174
`GIC+WP`	939.05	95.90	2,393	4,950
`Ours`	56.95	96.18	2,056	4,783

6 Conclusion

In this paper, we propose *SECRET* that adopts *SDA* for augmentation by discriminating modified edges based on prediction disparity and trains the model jointly with a neighborhood structure-aware contrastive scheme, *SCL*. Experimental results show that *SECRET* outperforms other state-of-the-art baselines in multiple benchmark datasets. In our future work, we plan to explore the diversity of the augmented graphs to improve the overall model performance.

Acknowledgments

This work was supported in part by the National Science and Technology Council (NSTC), Taiwan, under grants NSTC 111-2636-E-007-019-, NSTC 112-2221-E-007-088-MY3, and NSTC 111-2410-H-030-006-MY2.

References

[1] Seong Jin Ahn and MyoungHo Kim, 'Variational graph normalized autoencoders', in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2827–2831, (2021).

[2] Edo M Airoldi, David Blei, Stephen Fienberg, and Eric Xing, 'Mixed membership stochastic blockmodels', *Advances in neural information processing systems*, **21**, (2008).

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter, 'Learning representations by maximizing mutual information across views', *Advances in neural information processing systems*, **32**, (2019).

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, 'Curriculum learning', in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, (2009).

[5] Aleksandar Bojchevski and Stephan Günnemann, 'Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking', in *Proceedings of the International Conference on Learning Representations (ICLR)*, (2018).

[6] Andrew P Bradley, 'The use of the area under the roc curve in the evaluation of machine learning algorithms', *Pattern recognition*, **30**(7), 1145–1159, (1997).

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, 'A simple framework for contrastive learning of visual representations', in *International conference on machine learning*, pp. 1597–1607. PMLR, (2020).

[8] Aaron Clauset, Cristopher Moore, and Mark EJ Newman, 'Hierarchical structure and the prediction of missing links in networks', *Nature*, **453**(7191), 98–101, (2008).

[9] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King, 'Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding', in *Proceedings of The Web Conference 2020*, pp. 2331–2341, (2020).

[10] Kaveh Hassani and Amir Hosein Khasahmadi, 'Contrastive multi-view representation learning on graphs', in *International Conference on Machine Learning*, pp. 4116–4126. PMLR, (2020).

[11] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio, 'Learning deep representations by mutual information estimation and maximization', in *Proceedings of the International Conference on Learning Representations (ICLR)*, (2019).

[12] Bay-Yuan Hsu, Chih-Ya Shen, and Ming-Yi Chang, 'Wmego: Willingness maximization for ego network data extraction in online social networks', in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 515–524, (2020).

[13] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan, 'Multi-scale contrastive siamese networks for self-supervised graph representation learning', in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, (2021).

[14] Zekarias T Kefato and Sarunas Girdzijauskas, 'Self-supervised graph neural networks without explicit negative sampling', in *Proceedings of The Web Conference Workshops*, (2021).

[15] Diederik P Kingma and Max Welling, 'Auto-encoding variational bayes', *arXiv preprint arXiv:1312.6114*, (2013).

[16] Thomas N Kipf and Max Welling, 'Variational graph auto-encoders', *arXiv preprint arXiv:1611.07308*, (2016).

[17] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann, 'Predict then propagate: Graph neural networks meet personalized pagerank', in *Proceedings of the International Conference on Learning Representations (ICLR)*, (2019).

[18] Keng-Pei Lin, Yi-Wei Chang, Zhi-Hong Wei, Chih-Ya Shen, and Ming-Yi Chang, 'A smart contract-based mobile ticketing system with multi-signature and blockchain', in *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pp. 231–232. IEEE, (2019).

[19] Costas Mavromatis and George Karypis, 'Graph infoclust: Leveraging cluster-level node information for unsupervised graph representation learning', *arXiv preprint arXiv:2009.06946*, (2020).

[20] Péter Mernyei and Cătălina Cangea, 'Wiki-cs: A wikipedia-based benchmark for graph neural networks', in *Proceedings of ICML Workshop on Graph Representation Learning and Beyond*, (2020).

[21] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell, 'Representation learning via invariant causal mechanisms', in *Proceedings of the International Conference on Learning Representations (ICLR)*, (2020).

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

[23] Liming Pan, Cheng Shi, and Ivan Dokmanić, 'Neural link prediction with walk pooling', in *Proceedings of the International Conference on Learning Representations (ICLR)*, (2022).

[24] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang, 'Adversarially regularized graph autoencoder for graph embedding', in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, (2018).

[25] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang, 'Geom-gcn: Geometric graph convolutional networks', in *Proceedings of the International Conference on Learning Representations (ICLR)*, (2020).

[26] Benedek Rozemberczki, Carl Allen, and Rik Sarkar, 'Multi-scale attributed node embedding', *Journal of Complex Networks*, **9**(2), cnab014, (2021).

[27] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad, 'Collective classification in network data', *AI magazine*, **29**(3), 93–93, (2008).

[28] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann, 'Pitfalls of graph neural network evaluation', in *Proceedings of NeurIPS Workshops*, (2018).

[29] Chih-Ya Shen, Shao-Heng Ko, Guang-Siang Lee, Wang-Chien Lee, and De-Nian Yang, 'Density personalized group query', *Proceedings of the VLDB Endowment*, **16**(4), 615–628, (2022).

[30] Chih-Ya Shen, Hong-Han Shuai, De-Nian Yang, Guang-Siang Lee, Liang-Hao Huang, Wang-Chien Lee, and Ming-Syan Chen, 'On extracting socially tenuous groups for online social networks with k k-triangles', *IEEE Transactions on Knowledge and Data Engineering*, **34**(7), 3431–3446, (2020).

[31] Chih-Ya Shen, De-Nian Yang, and Ming-Syan Chen, 'Collaborative and distributed search system with mobile devices', *IEEE Transactions on Mobile Computing*, **11**(10), 1478–1493, (2011).

[32] Chih-Ya Shen, De-Nian Yang, Wang-Chien Lee, and Ming-Syan Chen, 'Activity organization for friend-making optimization in online social networks', *IEEE Transactions on Knowledge and Data Engineering*, **34**(1), 122–137, (2020).

[33] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm, 'Deep graph infomax', in *Proceedings of the International Conference on Learning Representations (ICLR)*, (2019).

[34] Xin Wang, Yudong Chen, and Wenwu Zhu, 'A survey on curriculum learning', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2021).

[35] Chen-Hsu Yang and Chih-Ya Shen, 'Enhancing machine learning approaches for graph optimization problems with diversifying graph augmentation', in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2191–2201, (2022).

[36] Chen-Hsu Yang, Hong-Han Shuai, Chih-Ya Shen, and Ming-Syan Chen, 'Learning to solve task-optimized group search for social internet of things', *IEEE Transactions on Knowledge and Data Engineering*, **34**(11), 5429–5445, (2021).

[37] Hao-Wei Yang, Ming-Yi Chang, and Chih-Ya Shen, '(Technical appendix and reproducibility materials) Enhancing link prediction with self-discriminating augmentation for structure-aware contrastive learning', in *https://bit.ly/3CAJBdw*, (2023).

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

[39] Muhan Zhang and Yixin Chen, 'Link prediction based on graph neural networks', *Advances in neural information processing systems*, **31**, (2018).

[40] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng, 'Graph debiased contrastive learning with joint representation clustering', in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3434–3440, (2021).

[41] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang, 'Deep graph contrastive representation learning', in *Proceedings of ICML Workshop on Graph Representation Learning and Beyond*, (2020).

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