Graph Structure Refinement with Energy-based Contrastive Learning

Xianlin Zeng[1, 2], **Yufeng Wang**[1], **Yuqi Sun**[1], **Guodong Guo**[3], **Wenrui Ding**[1], **Baochang Zhang**[1]

[1]Beihang University, Beijing, P.R.China
[2]Postdoctoral Research Station at China RongTong Academy of Sciences Group Corporation Limited, Beijing, P.R.China
[3]Ningbo Institute of Digital Twin, Eastern Institute of Technology, Ningbo, P.R.China
zengxianlin@buaa.edu.cn, wyfeng@buaa.edu.cn, sunyuqi@buaa.edu.cn, guodong.guo@mail.wvu.edu,
ding@buaa.edu.cn, bczhang@buaa.edu.cn

Abstract

Graph Neural Networks (GNNs) have recently gained widespread attention as a successful tool for analyzing graph-structured data. However, imperfect graph structure with noisy links lacks enough robustness and may damage graph representations, therefore limiting the GNNs' performance in practical tasks. Moreover, existing generative architectures fail to fit discriminative graph-related tasks. To tackle these issues, we introduce an unsupervised method based on a joint of generative training and discriminative training to learn graph structure and representation, aiming to improve the discriminative performance of generative models. We propose an Energy-based Contrastive Learning (ECL) guided Graph Structure Refinement (GSR) framework, denoted as ECL-GSR. To our knowledge, this is the first work to combine energy-based models with contrastive learning for GSR. Specifically, we leverage ECL to approximate the joint distribution of sample pairs, which increases the similarity between representations of positive pairs while reducing the similarity between negative ones. Refined structure is produced by augmenting and removing edges according to the similarity metrics among node representations. Extensive experiments demonstrate that ECL-GSR outperforms *the state-of-the-art on eight benchmark datasets* in node classification. ECL-GSR achieves *faster training with fewer samples and memories* against the leading baseline, highlighting its simplicity and efficiency in downstream tasks.

Introduction

With the explosive growth of graph-structured data, Graph Neural Networks (GNNs) (Zhu et al. 2019; Zhang and Zitnik 2020; Zhang et al. 2019a; Xu et al. 2018) have emerged as a potent deep learning tool, experiencing notable advancements across diverse applications, such as node classification (Veličković et al. 2017; Kipf and Welling 2016), node clustering (Wang et al. 2019a; Zhang et al. 2019b), graph classification (Duvenaud et al. 2015; Lee, Lee, and Kang 2019), link prediction (Peng et al. 2020; Srinivasan and Ribeiro 2019), recommendation systems (Wang et al. 2019b; Yu et al. 2021b), drug discovery (Wang et al. 2021b), and anomaly detection (Ding et al. 2019). GNNs usually adopt a message-passing scheme (Gilmer et al. 2017), aggregating information from neighboring nodes within the observed topology to compute graph representations. The strong representational capacity of most GNNs hinges on the assumption that graph structure is sufficiently reliable and perfectly noise-free (Szegedy et al. 2013), considered as ground-truth information for model training. However, this assumption may hardly hold in real-world applications. This is due to the fact: i) Raw structure is typically derived from complex interactive systems, leading to inherent uncertainties and incomplete connections. Even worse, the GNN iterative mechanism with cascading effects repeatedly aggregates neighborhood features. Minor noises in the graph can propagate to adjacent nodes, influencing other node embeddings and potentially introducing further inaccuracies (Dai et al. 2018). ii) Graph representation containing explicit structure is not informative enough to improve task performance. Raw topology only incorporates necessary physical connections, such as chemical bonds in molecules, and fails to capture abstract or implicit links among nodes. Furthermore, in various graph-related tasks, such as text graph in natural language processing (Yao, Mao, and Luo 2019) or scene graph for images in computer vision (Suhail et al. 2021), the explicit structure may either be absent or unavailable.

To tackle these challenges mentioned above, Graph Structure Refinement (GSR) involves learning invariant underlying relationships by extracting general knowledge from graph data, rather than relying solely on task-specific information. Therefore, the primary concern of GSR lies in graph representation learning, which can be broadly classified into two categories (Wang et al. 2018a). On one hand, graph generative models (Grover and Leskovec 2016; Dong, Chawla, and Swami 2017; Zheng et al. 2020) assume that each node follows an inherent connectivity distribution. Edges are viewed as samples from these distributions, with the models enhancing node representations by optimizing the likelihood of these observed edges. However, most downstream tasks are inherently discriminative, such as node classification and graph prediction. The state-of-the-art generative models have significantly deviated from discriminative architectures (Grathwohl et al. 2020). On the other hand, discriminative models (Veličković et al. 2019; Wang et al. 2018b; Hassani and Khasahmadi 2020) focus on learning a classifier to predict the presence of edges directly. They output a single scalar to represent the probability between node pair, thereby differentiating the connectivity of edges. Nonethe-

less, these models may suffer from overfitting to the training data, capturing noise instead of extracting latent useful features, as well as lack the ability to generalize across different datasets and diverse graph structures. Recently, (Wang et al. 2022) and (Kim and Ye 2022) establish a crucial connection between discriminative paradigms and Energy-based Models (EBMs) (LeCun et al. 2006), creating a unified framework to generate higher-quality samples and better visual representations. Motivated by these findings, we advocate for incorporating EBMs and Contrastive Learning (CL) to unlock the potential of generative models in addressing discriminative problems of graph-related tasks.

In this paper, we explore a novel Energy-based Contrastive Learning (ECL) approach to guide the GSR framework, termed ECL-GSR, which integrates EBMs with CL for unsupervised graph representation learning. Specifically, ECL complements discriminative training loss with generative loss, supplying higher quality and more robust representations for downstream tasks. Theoretically, we demonstrate that the existing discriminative loss is merely a specific instance of the ECL loss when the generative term is disabled. Empirically, ECL can be interpreted as maximizing the joint log-likelihood of the similarity between positive sample pairs with EBMs and minimizing the similarity between negative ones with CL, indicating the augmentations of identical and different samples, respectively. In GSR, we perform edge prediction by adding or removing links based on the similarity probabilities among node representations, further refining the raw structure. Finally, we evaluate ECL-GSR on the node classification task using the refined graph. The major contributions are threefold as follows:

- We present a novel ECL-GSR framework for joint graph structure and representation learning. It is the first work to combine EBMs with CL as generative and discriminative paradigms for GSR.
- Contrary to most GSR methods, ECL-GSR is a straightforward implementation, demanding fewer training iterations, memory costs, and data samples to obtain the equivalent or better performance.
- Extensive experiments on eight benchmark datasets demonstrate the superiority of ECL-GSR over current state-of-the-art methods. Ablation studies further confirm its effectiveness, efficiency, and robustness.

Background

Graph Structure Refinement

Given a raw graph $G = (\mathcal{V}, \mathcal{E}, X) = (A, X)$ with noisy topology, where $\mathcal{V}$ is the set of $V = |\mathcal{V}|$ nodes, $\mathcal{E}$ is the set of $M = |\mathcal{E}|$ edges, $X \in \mathbb{R}^{V \times D}$ is the node feature matrix (the i^{th} entry $x^i \in \mathbb{R}^D$ represents the attribute of node v_i), and $A \in \mathbb{R}^{V \times V}$ is the adjacency matrix ($A_{i,j} > 0$ indicates $e_{i,j} = (v_i, v_j), i, j \in M$). The target of graph structure refinement (Zhu et al. 2021) is to acquire a refined graph $\widetilde{G}$ with a clean adjacency $\widetilde{A}$, along with corresponding representation $\widetilde{Z} \in \mathbb{R}^{V \times \widetilde{F}}$, $\widetilde{F} \ll D$, for downstream tasks.

Energy-based Models

Given a point χ sampled from the data distribution $p_d(\chi)$, EBMs assign a scalar-valued energy function $E_\theta(\chi) \in \mathbb{R}$ by a DNN with parameters θ. The energy function define a probability distribution using the Boltzmann distribution $p_\theta(\chi) = \frac{\exp(-E_\theta(\chi))}{Z(\theta)}$, where $Z(\theta)$ is a normalizing constant or partition function ensuring p_θ integrates to 1. EBMs leverage the defined distribution p_θ to model the data distribution p_d by minimizing the negative log-likelihood of p_θ under p_d, as indicated by:

$$\min_\theta \mathbb{E}_{\chi \sim p_d}[-\log p_\theta(\chi)]. \tag{1}$$

The derivative of the negative log-likelihood $\mathcal{L}(\theta)$ is:

$$\nabla_\theta \mathcal{L}(\theta) \cong \mathbb{E}_{\chi^+ \sim p_d}[\nabla_\theta E_\theta(\chi^+)] - \mathbb{E}_{\chi^- \sim p_\theta}[\nabla_\theta E_\theta(\chi^-)]\,. \tag{2}$$

Eq. 2 decreases the energy values of positive samples χ^+ while increasing those of negative ones χ^- (Hinton 2002). However, computing $Z(\theta)$ for most parameterizations of $E_\theta(\chi)$ is intractable. We employ Stochastic Gradient Langevin Dynamics (SGLD) (Welling and Teh 2011) derived from Markov Chain Monte Carlo (MCMC) methods to reduce the mixing time of sampling procedure. Specifically, it generates p_θ as an approximation of p_d via iteratively updating χ, denoted as:

$$\chi_{k+1} = \chi_k - \frac{\lambda}{2}\nabla_\chi E_\theta(\chi_k) + \omega_k\,, \tag{3}$$

where $\omega_k \sim \mathcal{N}(0, \lambda)$. As $k \to \infty$ and $\lambda \to 0$, then p_θ converges to p_d. This process generates data samples through the energy function implicitly rather than explicitly.

Contrastive Learning

Given a set of random variables $\{\chi_n\}_{n=1}^N$, we define a data augmentation $\mathcal{T}$ to generate two distinct views $\nu_n = t(\chi_n)$, $\nu'_n = t'(\chi_n)$, i.e., $t, t' \sim \mathcal{T}$. CL constitutes an unsupervised framework for representation learning, aiming to maximize the mutual information I between the representations of two views ν_n and ν'_m w.r.t the joint distribution $p(\nu_n,\ \nu'_m)$. This is expressed as:

$$\max_{\mathcal{D}_\theta} I(z_n,\ z'_m)\,, \tag{4}$$

where $z_n = \mathcal{D}_\theta(\nu_n)$ is the representation and $\mathcal{D}_\theta(\cdot)$ is a parametric DNN. When $n = m$, the views (ν_n, ν'_m) are referred to as a positive pair with the same marginal distribution. Conversely, they are called a negative pair. In practice, each pair provides supervisory information to the other, playing a role similar to that of labels in a supervised manner. CL trains $\mathcal{D}_\theta$ to encourage z_n and its positive pair z'_n to be close in the projection space while pushing away representations of all negative pairs z'_m. This principle has been proven to be key in boosting performance (Chen et al. 2020).

Methodology

In this section, we delineate the proposed ECL-GSR framework. As shown in Fig. 1, our pipeline consists of four steps:

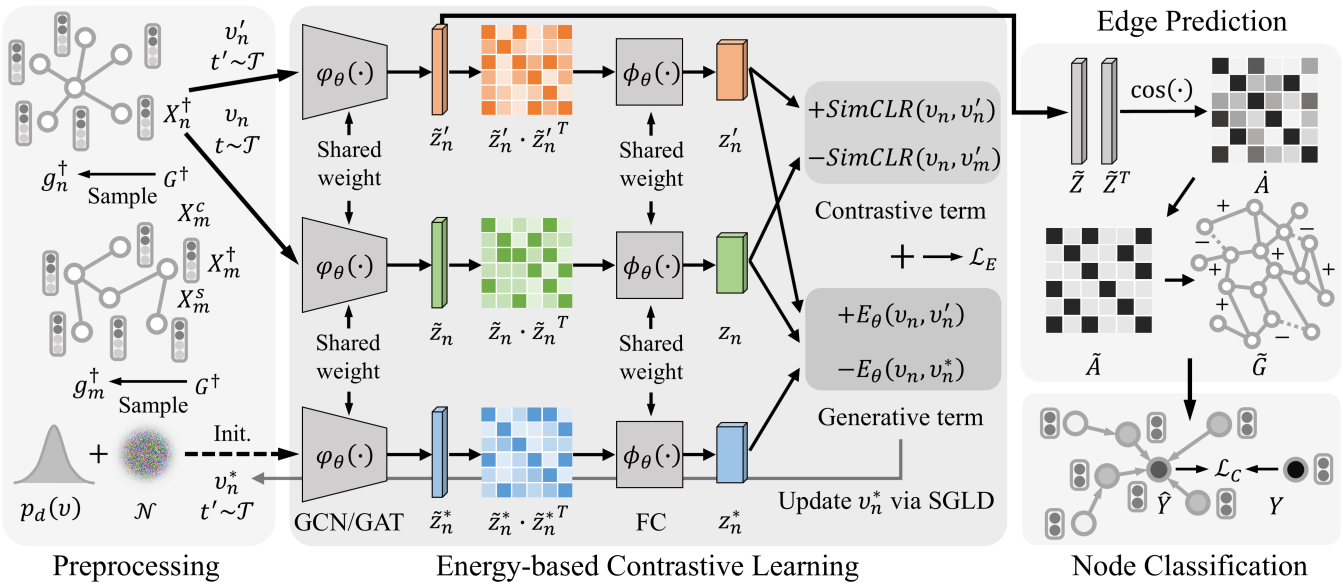

Figure 1: Illustration of the procedure of ECL-GSR. Preprocessed dual-attribute graph undergoes data augmentations, energy-based contrastive learning, and edge prediction, achieving structure refinement. Refined graph is applied in node classification.

preprocessing, energy-based contrastive learning, edge prediction, and node classification. Initially, we construct a dual-attribute graph by extracting contextual and structural information and acquiring subgraphs as input through edge sampling. Then, the ECL approach is introduced with the theorem and implementation. Next, we fine-tune the graph structure and evaluate the node classification task with raw features. Lastly, we present the final training objective.

Preprocessing

Dual-attribute graph Our framework can make full use of all trustworthy observations to maximize informativeness by constructing a dual-attribute graph. We concatenate the contextual information X^c, and structural embedding X^s as a new attribute, where X^c is derived directly from the raw node features and X^s is extracted using the DeepWalk (Perozzi, Al-Rfou, and Skiena 2014). Finally, the dual-attribute graph is represented as $G^\dagger = (X^\dagger, A)$, where $X^\dagger = [X^c, X^s]$ is the new node features.

Edge sampling To address memory limitations, neighbor sampling techniques enable stochastic training on large graphs G by decomposing them into smaller subgraphs g. Each subgraph sequentially contributes to GNNs' optimization, executing multiple gradient descent steps within a training epoch. We independently select several edges (node pairs) from edge set $\{e_{i,j}\}_{i,j=1}^{M}$ to produce a subgraph. This process yields a mini-batch of subgraphs $\{g_n\}_{n=1}^{N}$, each with a fixed number of edges.

Energy-based Contrastive Learning

Initially, we define p_d as a distribution of graph data and $\mathcal{T}$ as a set of predetermined data augmentation operators. Given a dual-attribute subgraph $g^\dagger$ and two augmentation views $t, t' \sim \mathcal{T}$ selected uniformly at random, we propose an ECL approach to build a joint distribution $p_\theta(\nu, \nu')$ over two views $\nu, \nu' = t(g^\dagger), t'(g^\dagger)$, aiming to approximate the data distribution $p_d(\nu, \nu')$.

Definition 1. *The joint distribution $p_\theta(\nu, \nu')$ can be defined as:*

$$p_\theta(\nu, \nu') = \frac{\exp(-f_\theta(\nu, \nu'))}{Z(\theta)}, \tag{5}$$

where $Z(\theta) = \int\int \exp(-f_\theta(\nu, \nu'))d\nu d\nu'$.

Building upon the assumption that semantically similar pairs (ν, ν') have nearby projections with high p_d, while dissimilar ones would correspond to distant projections with low p_d, we solve for the distance between ν and ν'. Let $f_\theta(\cdot) = \phi_\theta(\varphi_\theta(\cdot))$, $\varphi_\theta(\cdot)$ is a GNN encoder, and $\phi_\theta(\cdot)$ is a linear projection. $z = f_\theta(\nu)$ is the corresponding representation. The term $\|z - z'\|$ indicates the inverse of semantic similarity of ν and ν'. To approximate $p_\theta(\nu, \nu')$ to $p_d(\nu, \nu')$, Eq. 1 can be rephrased as:

$$\min_\theta \mathbb{E}_{p_d}[-\log p_\theta(\nu, \nu')]\,. \tag{6}$$

Proposition 1. *The joint distribution $p_\theta(\nu, \nu')$ can be formulated as an EBM:*

$$p_\theta(\nu, \nu') = \frac{\exp(-E_\theta(\nu, \nu'))}{Z(\theta)}, \tag{7}$$

where $E_\theta(\nu, \nu') = \|z - z'\|^2/\tau$, and τ is a temperature parameter.

The gradient of the objective of Eq. 6 is expressed as:

$$\begin{aligned}\nabla_\theta \mathbb{E}_{p_d}[-\log p_\theta(\nu, \nu')] = \\ \mathbb{E}_{p_d}[\nabla_\theta E_\theta(\nu, \nu')] - \mathbb{E}_{p_\theta}[\nabla_\theta E_\theta(\nu, \nu')]\,.\end{aligned} \tag{8}$$

To avoid directly calculating $Z(\theta)$, we employ Bayes' rule (Bayes 1763) to reformulate $\mathbb{E}_{p_d}[-\log p_\theta(\nu, \nu')]$ as:

$$\begin{aligned}\mathbb{E}_{p_d}[-\log p_\theta(\nu,\nu')] = \\ \mathbb{E}_{p_d}[-\log p_\theta(\nu'|\nu)] + \mathbb{E}_{p_d}[-\log p_\theta(\nu)]\,,\end{aligned} \tag{9}$$

where $p_\theta(\nu)$ is the marginal distribution of $p_\theta(\nu, \nu')$ over ν'.

Theorem 1. *The marginal distribution $p_\theta(\nu)$ is an EBM:*

$$p_\theta(\nu) = \frac{\exp(-E_\theta(\nu))}{Z(\theta)}, \tag{10}$$

where $E_\theta(\nu) = -\log \int \exp\left(-\|z - z'\|^2/\tau\right) d\nu'$.

Its proof is detailed in Appendix A.1. The gradient of the objective of Eq. 10 is defined as:

$$\nabla_\theta \mathbb{E}_{p_d}[-\log p_\theta(\nu)] = \mathbb{E}_{p_d}[\nabla_\theta E_\theta(\nu)] - \mathbb{E}_{p_\theta}[\nabla_\theta E_\theta(\nu)]\,. \tag{11}$$

According to Eq. 9, the objective of ECL is decomposed into the generative term and discriminative term, given by:

$$\mathcal{L}_b(\theta) = \mathbb{E}_{p_d}[-\log p_\theta(\nu'|\nu)] + \alpha \mathbb{E}_{p_d}[-\log p_\theta(\nu)]\,, \tag{12}$$

where α is a hyperparameter to trade off the strength of two terms. According to Eq. 11, the gradient of Eq. 12 is written as:

$$\begin{aligned}\nabla_\theta \mathcal{L}_b(\theta) = \mathbb{E}_{p_d}[-\nabla_\theta \log p_\theta(\nu'|\nu)] + \\ \alpha \mathbb{E}_{p_d}[\nabla_\theta E_\theta(\nu)] - \alpha \mathbb{E}_{p_\theta}[\nabla_\theta E_\theta(\nu)]\,.\end{aligned} \tag{13}$$

By this way, $Z(\theta)$ ingeniously cancels itself out in the discriminative term without additional calculations. For the generative term, we merely need to sample ν^* from $p_d(\nu)$ with adding Gaussian noise $\mathcal{N}(0, \lambda)$ and iteratively optimize ν^* through SGLD, as indicated in Eq. 3.

Implementation 1. *To implement the training of ECL, we approximate the generative term and discriminative term of Eq. 12, respectively, using the empirical mean of $p_\theta(\nu)$.*

Given a mini-batch of samples $\{(\nu_n, \nu'_n)\}_{n=1}^N$, along with its representations $\{(z_n, z'_n)\}_{n=1}^N$, we have N positive and $2(N-1)$ negative samples. Therefore, the empirical mean $\hat{p}_\theta(\nu_n)$ (Kim and Ye 2022) is defined as:

$$\hat{p}_\theta(\nu_n) = \frac{1}{2N} \sum_{\nu'_m : \nu'_m \neq \nu_n}^{2(N-1)} p_\theta(\nu_n, \nu'_m)\,. \tag{14}$$

For the discriminative term, we utilize $\frac{p_\theta(\nu_n, \nu'_n)}{\hat{p}_\theta(\nu_n)}$ to approximate the conditional probability density $p_\theta(\nu'|\nu)$. According the SimCLR framework (Chen et al. 2020), $\min\limits_\theta \mathbb{E}_{p_d}[-\log \hat{p}_\theta(\nu'_n|\nu_n)]$ can be represented as:

$$\min_{z \in f_\theta(\nu)} -\log \left(\frac{\exp\left(-\|z_n - z'_n\|^2/\tau\right)}{\frac{1}{2N}\sum_{\nu'_m : \nu'_m \neq \nu_n}^{2(N-1)} \exp\left(-\|z_n - z'_m\|^2/\tau\right)} \right). \tag{15}$$

Considering only N positive samples in the generative term, we simplify Eq. 14 to $\hat{p}_\theta(\nu_n) = \frac{1}{N}\sum_{n=1}^N p_\theta(\nu_n, \nu'_n)$. The approximation of $\min\limits_\theta \mathbb{E}_{p_d}[-\log \hat{p}_\theta(\nu_n)]$ is denoted as:

$$\min_{z \in f_\theta(\nu)} -\log \left(\sum_{n=1}^{N} \exp\left(-\|z_n - z'_n\|^2/\tau\right) \right). \tag{16}$$

Algorithm 1: The entire process of ECL-GSR

Input: Dual-attribute graph $G^\dagger$ with node classification label Y, EBM E_θ, augmentation operators t, t', batch size N, number of batches B, SGLD iterations K, and training epochs P

Output: The predicted label $\hat{Y}$

Construct $G^\dagger$ from G, randomly initialize $E_\theta(\bullet)$ (including $\varphi_\theta(\bullet)$, $\phi_\theta(\bullet)$) and $C_\theta(\bullet)$ with α, β, τ, μ;

for $p = 1, 2, \ldots, P$ **do**
  for $b = 1, 2, \ldots, B$ **do**
    Batch $\{g_n^\dagger\}_{n=1}^N$ from $G^\dagger$;
    Build $p_d(\nu, \nu')$ over $\nu, \nu' = t(g^\dagger), t'(g^\dagger)$;
    Sample $\{\nu_n, \nu'_n\}_{n=1}^N$ from $p_d(\nu, \nu')$;
    Calculate the discriminative term of $\mathcal{L}_b$ with Eq. 15;
    Sample $\{\nu_n^*\}_{n=1}^N$ from $p_d(\nu)$ with $\mathcal{N}(0, \lambda)$;
    for $k = 1, 2, \ldots, K$ **do**
      Sample $\omega_k \sim \mathcal{N}(0, \lambda)$;
      Update $\{\nu^*_{\mathrm{n,k+1}}\}_{n=1}^N$ from $\{\nu^*_{\mathrm{n,k}}\}_{n=1}^N$ with Eq. 3;
    Calculate the generative term of $\mathcal{L}_b$ with Eq. 16;
    Calculate $\nabla_\theta \mathcal{L}_b$ with Eq. 13 and $\mathcal{L}_E$ with Eq. 17;
    Calculate $\dot{A}$ with Eq. 18 and binarize $\dot{A}$ to yield $\widetilde{A}$;
    Predict $\hat{Y}$ with C_θ and calculate $\mathcal{L}_C$ with Y;
    Update θ_E and θ_C to minimize $\mathcal{L}$ with Eq. 19;

In summation, the final objective of ECL is:

$$\mathcal{L}_E(\theta) = \mathcal{L}_b(\theta) + \beta \mathcal{L}_r(\theta)\,, \tag{17}$$

where $\mathcal{L}_r(\theta) = \frac{1}{2N}\sum_{n \neq m} E_\theta\left(\nu_n, \nu'_m\right)^2$ is the L_2 regularization loss to prevent gradient overflow due to the excessive energy values. β is also a trade-off hyperparameter.

Edge Prediction

Upon completion of the ECL training, we are able to fine-tune the graph structure through edge prediction. The edge predictor receives the graph representation and subsequently outputs an edge probability matrix, denoted as $\dot{A}$. Each element $\dot{A}_{i,j}$ symbolizes the predicted probability of an edge existing between the pair of nodes (v_i, v_j):

$$\dot{A}_{i,j} = Norm(cos(\widetilde{z}_i, \widetilde{z}_j))\,, \tag{18}$$

where $\widetilde{z}_i, \widetilde{z}_j \in \widetilde{Z}$, $\cos(\bullet)$ is the cosine similarity function, $\widetilde{Z}$ denotes the representation output by the encoder $\varphi_\theta(\bullet)$, and $Norm(\bullet)$ is a normalization function. For the purpose of end-to-end training, we binarize $\dot{A}$ with the relaxed Bernoulli sampling (Zhao et al. 2021) on each edge to produce the final matrix $\widetilde{A}$.

Node Classification

Using $\widetilde{A}$ and X as inputs, we utilize a simple three-layer GNN as a node classifier C_θ, which can be instantiated with GCN or GAT architecture. Node representations are defined as $H = C_\theta(\widetilde{A}, X)$, and the predicted label $\hat{Y}$ aligns with the ground truth Y. For each node representation $h_i \in H$, $\hat{y}_i \in \hat{Y}$ is denoted as $Softmax(h_i)$. The node classification loss $\mathcal{L}_C(\theta)$ is the cross-entropy between $\hat{Y}$ and Y.

Method	Cora	Citeseer	Cornell	Texas	Wisconsin	Actor	Pubmed	OGB-Arxiv
GCN	81.46 ± 0.58	71.36 ± 0.31	47.84 ± 5.55	57.83 ± 2.76	57.45 ± 4.30	30.01 ± 0.77	**79.18 ± 0.29**	**70.77 ± 0.19**
GAT	81.41 ± 0.77	70.69 ± 0.58	46.22 ± 6.33	54.05 ± 7.35	57.65 ± 7.75	28.91 ± 0.83	77.85 ± 0.42	**69.90 ± 0.25**
LDS	83.01 ± 0.41	**73.55 ± 0.54**	47.87 ± 7.14	58.92 ± 4.32	61.70 ± 3.58	**31.05 ± 1.31**	OOM	OOM
GEN	80.21 ± 1.72	71.15 ± 1.81	**57.02 ± 7.19**	**65.94 ± 4.13**	**66.07 ± 3.72**	27.21 ± 2.05	78.91 ± 0.69	OOM
SGSR	83.48 ± 0.43	72.96 ± 0.25	44.32 ± 2.16	60.81 ± 4.87	56.86 ± 1.24	30.23 ± 0.38	78.09 ± 0.53	OOM
GRCN	**83.87 ± 0.49**	72.43 ± 0.61	54.32 ± 8.24	62.16 ± 7.05	56.08 ± 7.19	29.97 ± 0.71	78.92 ± 0.39	OOM
IDGL	**83.88 ± 0.42**	72.20 ± 1.18	50.00 ± 8.98	62.43 ± 6.09	59.41 ± 4.11	28.16 ± 1.41	OOM	OOM
GAuG-O	82.20 ± 0.80	71.60 ± 1.10	57.60 ± 3.80	56.90 ± 3.60	54.80 ± 5.70	25.80 ± 1.00	**79.30 ± 0.40**	OOM
SUBLIME	83.40 ± 0.42	72.30 ± 1.09	**70.29 ± 3.51**	**70.21 ± 2.32**	**66.73 ± 2.44**	30.79 ± 0.68	73.80 ± 0.60	55.50 ± 0.10
ProGNN	80.30 ± 0.57	68.51 ± 0.52	54.05 ± 6.16	48.37 ± 8.75	62.54 ± 7.56	22.35 ± 0.88	71.60 ± 0.46	OOM
CoGSL	81.76 ± 0.24	**73.09 ± 0.42**	52.16 ± 3.21	59.46 ± 4.36	58.82 ± 1.52	**32.95 ± 1.20**	OOM	OOM
STABLE	80.20 ± 0.68	68.91 ± 1.01	44.03 ± 4.05	55.24 ± 6.04	53.00 ± 5.27	30.18 ± 1.00	OOM	OOM
NodeFormer	80.28 ± 0.82	71.31 ± 0.98	42.70 ± 5.51	58.92 ± 4.32	48.43 ± 7.02	25.51 ± 1.17	78.21 ± 1.43	55.40 ± 0.23
ECL-GSR	**84.06 ± 0.84**	**73.70 ± 0.75**	**71.27 ± 2.06**	**72.97 ± 3.39**	**67.79 ± 1.03**	**33.71 ± 0.96**	**80.91 ± 1.12**	**71.09 ± 0.31**

Table 1: Node classification accuracy (mean(%)±std) with the standard splits on various benchmark datasets. The top three results are highlighted in **first best**, **second best**, and **third best**, respectively. "OOM" indicates out of memory.

Training Objective

During the training process, we can efficiently compute the joint classification loss $\mathcal{L}_C(\theta)$ and ECL loss $\mathcal{L}_E(\theta)$ using gradient descent-based backpropagation techniques. The overall loss is:

$$\min_{\theta_E,\theta_C} \mathcal{L}(\theta) = \mathcal{L}_E(\theta) + \mu\mathcal{L}_C(\theta)\,, \tag{19}$$

where θ_E and θ_C are parameters of $E_\theta(\cdot)$ and $C_\theta(\cdot)$, respectively. The pseudocode of ECL-GSR is illustrated in Algorithm 1. The training stability is presented in Appendix A.6.

Experiments

We conduct comprehensive experiments to sequentially evaluate the proposed framework's effectiveness, complexity, and robustness, addressing five research questions: RQ1: How effective is ECL-GSR on the node classification task? RQ2: How efficient is ECL-GSR in terms of training time and space? RQ3: How do ECL architecture and its hyperparameters impact the performance of node-level representation learning? RQ4: How robust is ECL-GSR in the face of structural attacks or noises? RQ5: What kind of refined structure does ECL-GSR learn?

Experimental Setups

Datasets For extensive comparison, we execute experiments on eight benchmark datasets: four citation networks (Cora, Citeseer (Sen et al. 2008), Pubmed (Namata et al. 2012), and OGB-Arxiv (Hu et al. 2020)), three webpage graphs (Cornell, Texas, and Wisconsin (Pei et al. 2020)), and one actor co-occurrence network (Actor (Tang et al. 2009)).

Baselines To corroborate the promising performance of ECL-GSR, we compare it against 13 GSR baseline methods. There are two GNN baselines (GCN (Kipf and Welling 2016) and GAT (Veličković et al. 2017)), three adjacency matrix direct-optimization methods (NodeFormer (Wu et al. 2022), STABLE (Li et al. 2022), and ProGNN (Jin et al. 2020)), four probability estimation techniques (GEN (Wang et al. 2021a), GAuG-O (Zhao et al. 2021), SGSR (Zhao et al. 2023), and LDS (Franceschi et al. 2019)), and four metric learning approaches (SUBLIME (Liu et al. 2022b), GRCN (Yu et al. 2021a), CoGSL (Liu et al. 2022a), and IDGL (Chen, Wu, and Zaki 2020)).

Implementation details Our framework operates on an Ubuntu system with an NVIDIA GeForce 3090 GPU, employing PyTorch 1.12.1, DGL 1.1.0, and Python 3.9.16. All experiments are conducted using the reimplementation of GSLB (Li et al. 2023). We maintain the dimensions of contextual X^c and structural X^s features equal to that of raw attribute. Subgraph sampling batch size N is fixed at 64 for efficiency consideration. In ECL, the backbone $f_\theta(\cdot)$ is divided into $\varphi_\theta(\cdot)$ for encoding, utilizing three GCN layers with the hidden and output dimension $\widetilde{F}$ of 128, and $\phi_\theta(\cdot)$ for projection, comprising two fully-connected layers with an output dimension F of 128. The learned representation $\widetilde{Z}$ is produced by $\varphi_\theta(\cdot)$. Batch normalization is discarded when utilizing SGLD. The data augmentation operator $\mathcal{T}$ is a random Gaussian blur. For node classification, classifier $C_\theta(\cdot)$ mirrors the architecture of $\varphi_\theta(\cdot)$. Our model's final hyperparameters are set as: α=0.1, β=0.01, μ=0.01, and τ=0.1. We adopt the Adam optimizer with an initial learning rate of 0.001, halving every 20 epochs. The epochs P for Cora, Citeseer, Cornell, Texas, and Wisconsin are 40, and those for Actor, Pubmed, and OGB-Arxiv are 80. The number of SGLD's iterations K only takes 3 steps.

Node Classification Performance (RQ1)

Evaluation on standard splits As stated in Table 1, three key observations can be made: i) ECL-GSR shows robust performance across all benchmark datasets, demonstrating its superior generalizability to diverse data. Notably, within the ambit of eight datasets, ECL-GSR achieves the state-of-the-art with margins ranging from 0.15% to 1.61% over the second-highest approach. ii) Compared to other baselines, ECL-GSR exhibits enhanced performance stability and reduced standard deviation, particularly evident on the Cor-

Method	Cora				Citeseer			
	1%	3%	5%	10%	1%	3%	5%	10%
GCN	59.31 ± 0.29	77.14 ± 0.21	80.73 ± 0.63	83.53 ± 0.42	60.64 ± 1.07	67.60 ± 0.47	70.05 ± 0.54	74.38 ± 0.27
GAT	65.36 ± 0.99	76.36 ± 0.61	81.73 ± 0.21	83.92 ± 0.42	58.48 ± 2.35	68.41 ± 0.76	70.73 ± 0.22	**74.54 ± 0.14**
LDS	68.47 ± 1.11	78.06 ± 0.98	81.42 ± 0.66	83.87 ± 0.48	61.35 ± 1.57	67.29 ± 1.34	70.82 ± 0.79	74.54 ± 0.49
IDGL	**70.83 ± 1.21**	**78.60 ± 0.28**	**83.82 ± 0.28**	85.51 ± 0.08	60.61 ± 1.32	64.34 ± 1.61	69.39 ± 1.24	74.19 ± 0.58
SGSR	55.11 ± 0.43	77.32 ± 0.17	83.51 ± 0.22	**85.56 ± 0.25**	54.28 ± 0.47	**71.61 ± 0.17**	**72.88 ± 0.20**	74.31 ± 0.24
GRCN	68.38 ± 2.10	75.24 ± 1.06	79.16 ± 0.82	84.82 ± 0.41	59.06 ± 1.80	66.17 ± 0.75	72.11 ± 0.56	74.49 ± 0.73
CoGSL	64.43 ± 3.35	73.21 ± 1.10	79.02 ± 3.22	81.05 ± 0.53	56.41 ± 0.91	66.60 ± 0.79	69.96 ± 0.56	74.17 ± 0.53
ProGNN	70.32 ± 1.16	75.93 ± 0.78	81.35 ± 0.68	82.01 ± 0.67	56.77 ± 0.88	70.34 ± 0.66	70.67 ± 0.79	74.23 ± 0.36
SUBLIME	65.94 ± 4.90	73.37 ± 0.78	79.14 ± 0.26	82.37 ± 0.20	57.85 ± 1.64	67.67 ± 0.84	70.53 ± 0.16	71.47 ± 0.08
NodeFormer	67.11 ± 1.07	75.87 ± 0.79	82.05 ± 0.67	83.92 ± 0.45	**67.03 ± 0.89**	67.84 ± 0.60	70.65 ± 1.05	73.03 ± 0.37
ECL-GSR	**72.33 ± 0.39**	**79.99 ± 0.21**	**84.30 ± 0.18**	**85.71 ± 0.20**	**68.06 ± 0.54**	**72.18 ± 0.15**	**73.38 ± 0.22**	**74.90 ± 0.23**

Table 2: Node classification accuracy (mean(%)±std) with the different train ratios on Cora and Citeseer datasets. The top two results are highlighted in **first best** and **second best**, respectively.

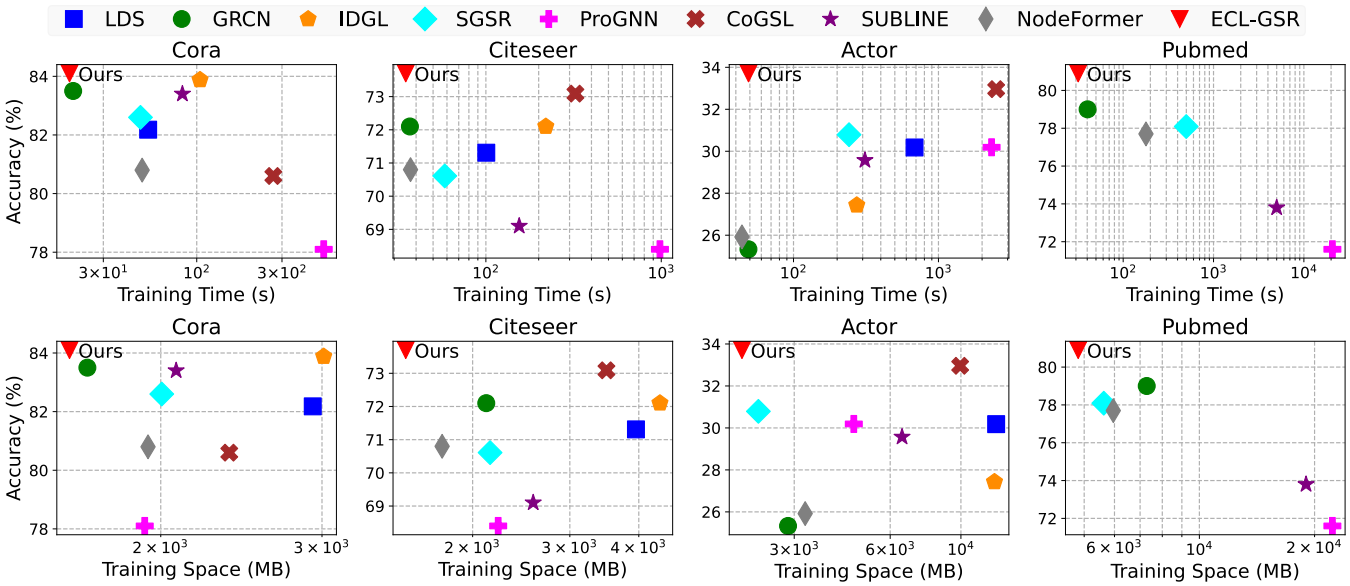

Figure 2: Training time and space analysis on Cora, Citeseer, Actor, and Pubmed datasets.

nell, Texas, Wisconsin, and Actor datasets. iii) Whereas certain competing algorithms such as CoGSL, GEN, and IDGL encounter OOM errors with Pubmed and OGB-Arxiv, our approach achieves the state-of-the-art on large benchmarks.

Evaluation on different train ratios In Table 2, we conduct experiments on Cora and Citeseer datasets with varying amounts of supervised information, specifically at training ratios of 1%, 3%, 5%, and 10%. The results indicate that our framework substantially outperforms existing baselines in terms of accuracy at a low training ratio. Among GSR approaches, we validate that ECL-GSR achieves equivalent or better performance with fewer training samples as well as maintains competitive performance at a high training ratio.

Efficiency and Scalability Analysis (RQ2)

In this section, we analyze the efficiency and scalability of ECL-GSR on Cora, Citeseer, Actor, and Pubmed datasets. As illustrated in Fig. 2, the position nearer to the figure's upper left corner signifies superior overall performance. For efficiency, the time complexity of performing an ECL-GSR is delineated by $\mathcal{O}(P{\cdot}K{\cdot}B)$, where B represents the number of batches. The higher time efficiency of our approach stems from its expedited convergence, necessitating only a limited quantity of training epochs P and iterations K. Regarding scalability, we can flexibly adapt the stochastic training by adjusting the mini-batch N, enabling to achieve an acceptable space complexity of $\mathcal{O}(N^2)$.

Conventional GSR algorithms are typically hindered by their considerable time and space demands, constraining their applicability in large-scale graphs. Some alternative solutions, such as NodeFormer and SGSR, have been recognized for their speed, albeit due to diminished classification accuracy. Methods like CoGSL and LDS are notable for their effectiveness, yet they demand considerable computational and storage requirements. Conversely, ECL-GSR

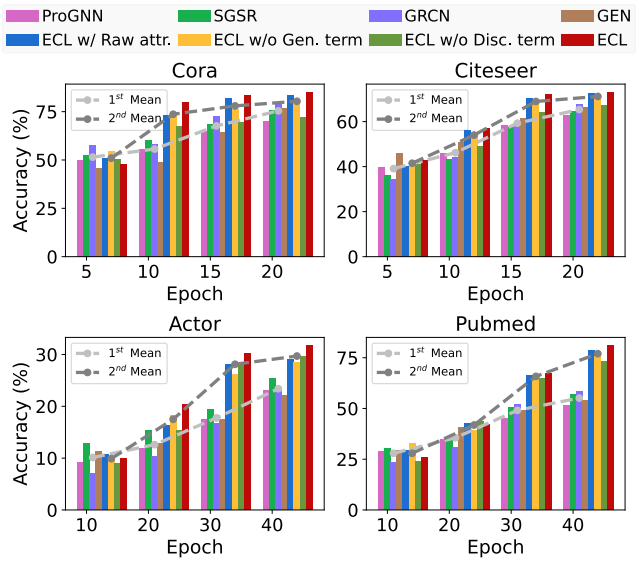

Figure 3: Performance study of ECL-GSR variants and other baselines over multiple training epochs on four datasets.

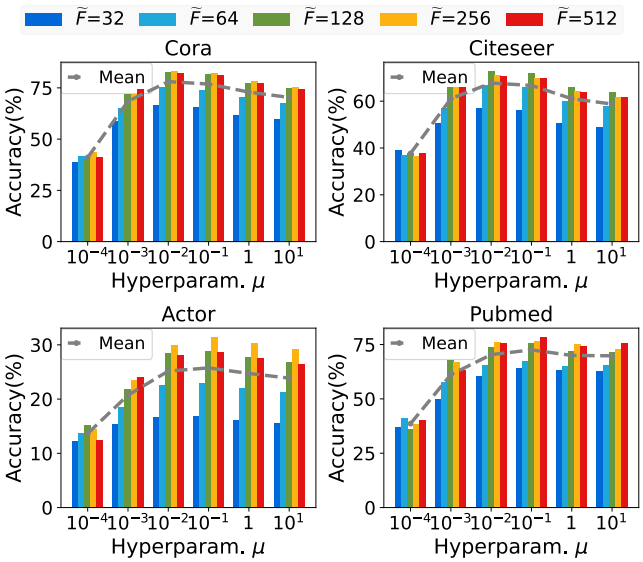

Figure 4: Hyperparameter μ and dimensionality $\widetilde{F}$ analysis of ECL-GSR on four datasets. "Mean" denotes the averages.

achieves advantages in terms of accuracy, speed, and memory usage, especially on Citeseer and Pubmed datasets.

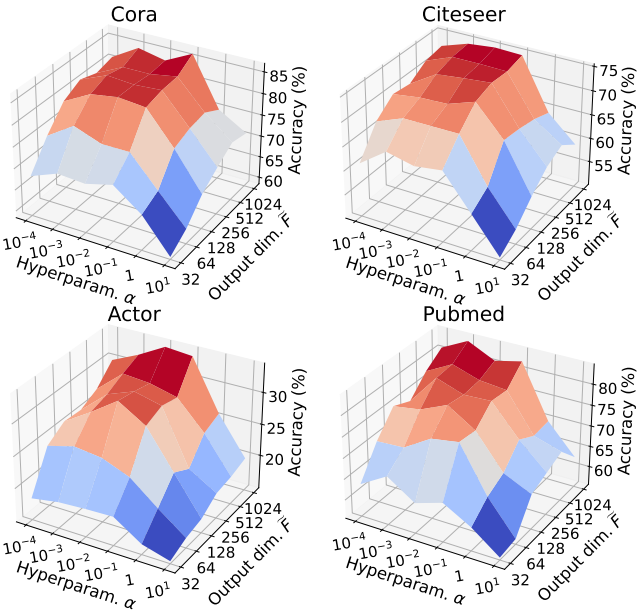

Figure 5: Hyperparameter α and dimensionality $\widetilde{F}$ analysis of ECL-GSR on four datasets.

Ablation Study (RQ3)

Component analysis As illustrated in Fig. 3, we investigate the impact of various configurations on Cora, Citeseer, Actor, and Pubmed datasets, evaluating the performance of GEN, GRCN, SGSR, ProGNN, ECL with raw attr., ECL without gen. term, ECL without disc. term, and full ECL over a range of training epochs. "1^{st} and 2^{nd} Mean" are averages of baselines and variations, respectively. Our findings indicate that: i) Utilizing raw graph attributes without structural embeddings marginally reduces the accuracy of ECL-GSR. (ii) When either generative or discriminative terms are absent, a notable decrease in performance suggests a vital role for the combination of them. (iii) All variants reach their peak performance within fewer training epochs, highlighting our framework's swift convergence compared to other GSR.

Parameter analysis The impacts of varying hyperparameter α, μ and output dimension $\widetilde{F}$ on Cora, Citeseer, Actor, and Pubmed datasets are indicated in Fig. 4 and Fig. 5, respectively. With respect to μ, a low value weakens the constraint of classification loss, whereas a high value leads our framework to degrade to baseline, thereby diminishing the role of ECL. Regarding α, we explore the importance of the generative term relative to the discriminative term in ECL. It suggests that setting α to 1.0 or higher yields suboptimal results. The performance peaks at 0.1 and then experiences a marginal decline as α is decreased further. For selecting $\widetilde{F}$, it is crucial for balancing representational adequacy and preventing overfitting. Lower dimensions compromise performance due to insufficient representation, while higher dimensions maintain performance but add model complexity.

Robustness Analysis (RQ4)

To evaluate the robustness of ECL-GSR, we randomly add or remove edges from the raw graph on Cora and Citeseer datasets and then evaluate the performance of various algorithms on the corrupted graph. We change the ratios of modified edges from 0 to 0.8 to simulate different noise intensities and compare our framework with GCN, GRCN, LDS, ProGNN, and IDGL. As revealed in Fig. 6, the performance of models generally shows a downward trend with increased attack intensity. Nonetheless, GSR approaches commonly

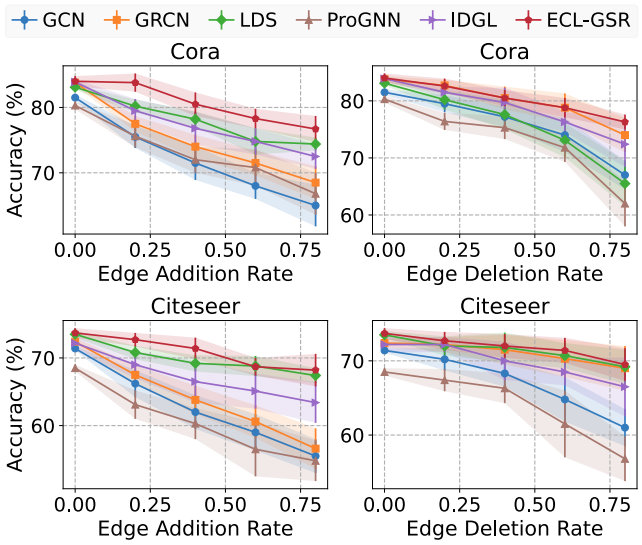

Figure 6: Robustness analysis by randomly adding and removing edges on Cora and Citeseer datasets.

exhibit better stability than GCN baseline. When edge addition and deletion rates increase, ECL-GSR consistently achieves better or comparable results in both scenarios, indicating its robustness in severe structural attacks.

Structure Visualization (RQ5)

To enhance the comprehension of refined topology, we present the visualization results of edge weights for both the raw and refined graph, as depicted in Fig. 7. We adhere to the previous strategy (Li et al. 2023) to select several subgraphs from Cora and Citeseer datasets. Randomly sampling 20 labeled (L) and 20 unlabeled (U) nodes, we extract four subgraphs and separate them with red lines. A subgraph contains two classes, each with 10 nodes. Intra- and inter-class connections are separated by green lines. The diagonal elements represent self-loops. Comparing the sparse intra- and inter-class connections of raw graph, the refined graph shows a significantly denser structure. However, a denser graph does not necessarily equate to improved performance. We find that ECL-GSR maintains a lower frequency of inter-class connections than intra-class ones. This observation aligns with the basic principle of ECL, which is to pull close similar semantic information and push away dissimilar ones.

Conclusion

In this paper, we advance an Energy-based Contrastive Learning approach to guide GSR and introduce a novel ECL-GSR framework, which jointly optimizes graph structure and representation. ECL is capable of approximating the joint distribution to learn good representations by combining generative and discriminative paradigms. We evaluate the proposed method on the graph node classification task. Experimental results verify its superior effectiveness, efficiency, and robustness.

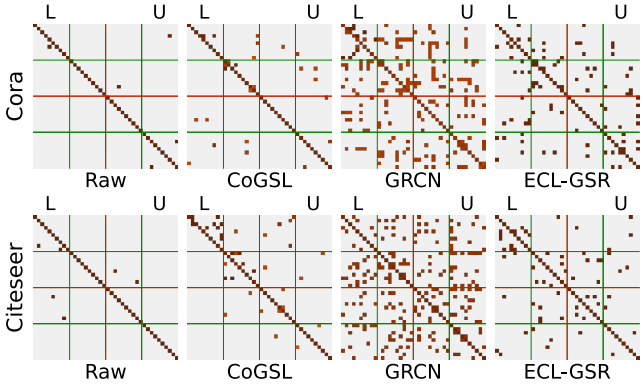

Figure 7: Visualization of the refined adjacency matrixes by various GSR algorithms on Cora and Citeseer datasets.

Acknowledgments

The work was supported by the National Key Research and Development Program of China (Grant No. 2023YFC3306401). This work was also supported by the National Natural Science Foundation of China (Grant No. U20B2042 and 62076019).

References

Bayes, T. 1763. LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. *Philosophical Transactions of the Royal Society of London*, (53): 370–418.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*. PMLR.

Chen, Y.; Wu, L.; and Zaki, M. 2020. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings. *Advances in Neural Information Processing Systems*, 33: 19314–19326.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial Attack on Graph Structured Data. In *International Conference on Machine Learning*. PMLR.

Ding, K.; Li, J.; Bhanushali, R.; and Liu, H. 2019. Deep Anomaly Detection on Attributed Networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 135–144.

Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; and Adams, R. P. 2015. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *Advances in Neural Information Processing Systems*, 28.

Franceschi, L.; Niepert, M.; Pontil, M.; and He, X. 2019. Learning Discrete Structures for Graph Neural Networks. In *International Conference on Machine Learning*. PMLR.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*. PMLR.

Grathwohl, W.; Wang, K.-C.; Jacobsen, J.-H.; Duvenaud, D.; Norouzi, M.; and Swersky, K. 2020. Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One. In *International Conference on Learning Representations*.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive Multi-view Representation Learning on Graphs. In *International Conference on Machine Learning*, 4116–4126. PMLR.

Hinton, G. E. 2002. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8): 1771–1800.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *Advances in Neural Information Processing Systems*, 33: 22118–22133.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph Structure Learning for Robust Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Kim, B.; and Ye, J. C. 2022. Energy-Based Contrastive Learning of Visual Representations. *Advances in Neural Information Processing Systems*, 35: 4358–4369.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*.

LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; and Huang, F. 2006. A Tutorial on Energy-based Learning. *Predicting Structured Data*, 1(0).

Lee, J.; Lee, I.; and Kang, J. 2019. Self-attention Graph Pooling. In *International Conference on Machine Learning*. PMLR.

Li, K.; Liu, Y.; Ao, X.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2022. Reliable Representations Make a Stronger Defender: Unsupervised Structure Refinement for Robust GNN. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Li, Z.; Wang, L.; Sun, X.; Luo, Y.; Zhu, Y.; Chen, D.; Luo, Y.; Zhou, X.; Liu, Q.; Wu, S.; et al. 2023. GSLB: The Graph Structure Learning Benchmark. *arXiv preprint arXiv:2310.05174*.

Liu, N.; Wang, X.; Wu, L.; Chen, Y.; Guo, X.; and Shi, C. 2022a. Compact Graph Structure Learning via Mutual Information Compression. In *Proceedings of the ACM Web Conference 2022*.

Liu, Y.; Zheng, Y.; Zhang, D.; Chen, H.; Peng, H.; and Pan, S. 2022b. Towards Unsupervised Deep Graph Structure Learning. In *Proceedings of the ACM Web Conference 2022*.

Namata, G.; London, B.; Getoor, L.; Huang, B.; and Edu, U. 2012. Query-driven Active Surveying for Collective Classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. *arXiv preprint arXiv:2002.05287*.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. In *Proceedings of the Web Conference 2020*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Magazine*, 29(3): 93–93.

Srinivasan, B.; and Ribeiro, B. 2019. On the Equivalence between Node Embeddings and Structural Graph Representations. *arXiv preprint arXiv:1910.00452*.

Suhail, M.; Mittal, A.; Siddiquie, B.; Broaddus, C.; Eledath, J.; Medioni, G.; and Sigal, L. 2021. Energy-based Learning for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199*.

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social Influence Analysis in Large-scale Networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph Attention Networks. *arXiv preprint arXiv:1710.10903*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *International Conference on Learning Representations*.

Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; and Zhang, C. 2019a. Attributed Graph Clustering: A Deep Attentional Embedding Approach. *arXiv preprint arXiv:1906.06532*.

Wang, H.; Wang, J.; Wang, J.; Zhao, M.; Zhang, W.; Zhang, F.; Xie, X.; and Guo, M. 2018a. GraphGAN: Graph Representation Learning with Generative Adversarial Nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Wang, H.; Zhang, F.; Hou, M.; Xie, X.; Guo, M.; and Liu, Q. 2018b. Shine: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 592–600.

Wang, R.; Mou, S.; Wang, X.; Xiao, W.; Ju, Q.; Shi, C.; and Xie, X. 2021a. Graph Structure Estimation Neural Networks. In *Proceedings of the Web Conference 2021*.

Wang, S.; Hu, L.; Wang, Y.; Cao, L.; Sheng, Q. Z.; and Orgun, M. 2019b. Sequential Recommender Systems: Challenges, Progress and Prospects. *arXiv preprint arXiv:2001.04830*.

Wang, Y.; Min, Y.; Chen, X.; and Wu, J. 2021b. Multi-view Graph Contrastive Representation Learning for Drug-Drug Interaction Prediction. In *Proceedings of the Web Conference 2021*.

Wang, Y.; Wang, Y.; Yang, J.; and Lin, Z. 2022. A Unified Contrastive Energy-based Model for Understanding the Generative Ability of Adversarial Training. *arXiv preprint arXiv:2203.13455*.

Welling, M.; and Teh, Y. W. 2011. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*.

Wu, Q.; Zhao, W.; Li, Z.; Wipf, D. P.; and Yan, J. 2022. Nodeformer: A Scalable Graph Structure Learning Transformer for Node Classification. *Advances in Neural Information Processing Systems*, 35: 27387–27401.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful are Graph Neural Networks? *arXiv preprint arXiv:1810.00826*.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph Convolutional Networks for Text Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

Yu, D.; Zhang, R.; Jiang, Z.; Wu, Y.; and Yang, Y. 2021a. Graph-revised Convolutional Network. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*. Springer.

Yu, J.; Yin, H.; Li, J.; Wang, Q.; Hung, N. Q. V.; and Zhang, X. 2021b. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *Proceedings of The Web Conference 2021*.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019a. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Zhang, X.; Liu, H.; Li, Q.; and Wu, X.-M. 2019b. Attributed Graph Clustering via Adaptive Graph Convolution. *arXiv preprint arXiv:1906.01210*.

Zhang, X.; and Zitnik, M. 2020. GNNGUARD: Defending Graph Neural Networks against Adversarial Attacks. *Advances in Neural Information Processing Systems*, 33: 9263–9275.

Zhao, J.; Wen, Q.; Ju, M.; Zhang, C.; and Ye, Y. 2023. Self-Supervised Graph Structure Refinement for Graph Neural Networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*.

Zhao, T.; Liu, Y.; Neves, L.; Woodford, O.; Jiang, M.; and Shah, N. 2021. Data Augmentation for Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35.

Zheng, S.; Zhu, Z.; Zhang, X.; Liu, Z.; Cheng, J.; and Zhao, Y. 2020. Distribution-induced Bidirectional Generative Adversarial Network for Graph Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7224–7233.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust Graph Convolutional Networks against Adversarial Attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Zhu, Y.; Xu, W.; Zhang, J.; Du, Y.; Zhang, J.; Liu, Q.; Yang, C.; and Wu, S. 2021. A Survey on Graph Structure Learning: Progress and Opportunities. *arXiv preprint arXiv:2103.03036*.