Explanation-Preserving Augmentation for Semi-Supervised Graph Representation Learning

Zhuomin Chen[1], Jingchao Ni[2], Hojat Allah Salehi[1], Xu Zheng[1], Esteban Schafir[1], Farhad Shirani [*], Dongsheng Luo[1†]

[1] Knight Foundation School of Computing and Information Sciences, Florida International University
[2] Department of Computer Science, University of Houston
zchen051@fiu.edu, jni7@uh.edu, hsalehi@fiu.edu, xzhen019@fiu.edu, escha032@fiu.edu, luodongsheng01@gmail.com

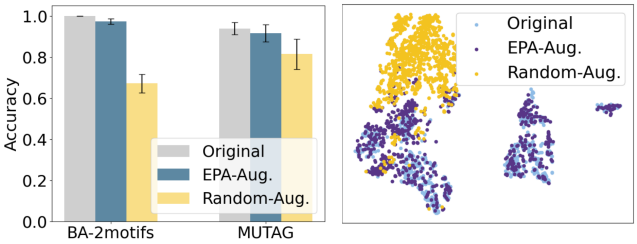
Abstract

Self-supervised graph representation learning (GRL) typically generates paired graph augmentations from each graph to infer similar representations for augmentations of the same graph, but distinguishable representations for different graphs. While effective augmentation requires both semantics-preservation and data-perturbation, most existing GRL methods focus solely on data-perturbation, leading to suboptimal solutions. To fill the gap, in this paper, we propose a novel method, Explanation-Preserving Augmentation (EPA), which leverages graph explanation for semantics-preservation. EPA first uses a small number of labels to train a graph explainer, which infers the subgraphs that explain the graph's label. Then these explanations are used for generating semantics-preserving augmentations for boosting self-supervised GRL. Thus, the entire process, namely EPA-GRL, is semi-supervised. We demonstrate theoretically, using an analytical example, and through extensive experiments on a variety of benchmark datasets, that EPA-GRL outperforms the state-of-the-art (SOTA) GRL methods that use semantics-agnostic augmentations.

Code — https://github.com/realMoana/EPA-GRL
Extended version — https://arxiv.org/abs/2410.12657

Introduction

Inspired by recent progress in self-supervised representation learning in vision and language domains (Chen et al. 2020), contrastive learning has emerged as a predominant technique for graph representation learning (GRL). Typically, two augmentations are generated for each graph with the objective of learning similar representations for augmentations of the same graph but discriminative representations for augmentations of different graphs. The success of self-supervised GRL is grounded in an effective augmentation strategy. Analogous to image data augmentation (He et al. 2020), an ideal pair of graph augmentations should concurrently be able to (1) inherit the semantics – which may be represented by signature subgraphs pertinent to the classification – of their original graph; and (2) present sufficient variance from each other (Yin et al. 2022).



(a) Classification Accuracy (b)Visualization

Figure 1: Semantics-preserving ability of different augmentations.

However, most existing works only focus on structural perturbations that add variance to the augmented graphs but largely neglect the need for preserving semantics. For example, in graph contrastive learning (GraphCL) (You et al. 2020) and JOAO (You et al. 2021), an augmented graph is typically generated by perturbing its original graph through random node/edge dropping, feature masking, and subgraph extraction. The randomness in these perturbations inevitably induces substantial alterations to some important (sub-)graph structures or features that may result in a considerable loss of semantics, which in turn, may lead to significant performance drops in downstream tasks.

Fig. 1 illustrates an experiment that evaluates the semantics-preserving ability of different augmentation techniques. A Graph Neural Network (GNN) classifier (Kipf and Welling 2017) is first trained on the training partition of a benchmark dataset (*i.e.*, BA-2motifs (Luo et al. 2020) or MUTAG (Debnath et al. 1991)), and its accuracy is then evaluated on the original graphs in the test set and the augmented graphs derived from them, respectively. Two augmentation methods are included. "Random-Aug" represents randomly dropping nodes from the original graph in a *semantics-agnostic* manner. "EPA-Aug" is an explanation-preserving augmentation (EPA) that operates in a semantics-preserving manner (which will be introduced later). From Fig. 1(a), the fully trained GNN can accurately classify the graphs in the original test set. However, it has a sharp drop in accuracy on the "Random-Aug" graphs, suggesting a significant loss of semantics (*i.e.*, class-related subgraphs).

In contrast, the accuracy on "EPA-Aug" graphs is close

[*]Farhad Shirani was affiliated with Florida International University during the preparation of this work.
[†]Corresponding author

to that of the original graphs, implying effective semantics-preservation. Fig.1(b) shows the graph embeddings in BA-2motifs. A large distribution shift from the original graphs to the "Random-Aug" graphs can be observed. In addition, "EPA-Aug" well preserves the distribution of the original embeddings while presenting sufficient variance. Thus, semantics-preserving augmentations are more suitable than semantics-agnostic ones for training a generalizable GNN using GRL.

Prior attempts to address semantics preservation either rely on domain-specific knowledge (Sun et al. 2021), which limits its generalizability, or employ parameter perturbations (Xia et al. 2022) that may implicitly alter graph structures. Some methods use unsupervised approaches to preserve structural patterns (Shi, Zhou, and Liu 2023), but without label guidance, these patterns may be irrelevant to class discrimination, limiting their effectiveness for downstream tasks.

In this paper, we aim to develop semantics-preserving augmentation techniques – using models trained by leveraging only a few labeled input samples – to enhance GRL. Inspired by recent research on explainable AI (XAI) for graphs (Ying et al. 2019; Luo et al. 2020; Yuan et al. 2022), we propose a novel approach, Explanation-Preserving Augmentation enhanced GRL (EPA-GRL), which leverages graph explanation techniques for generating augmented graphs that can bridge the gap between semantics-preservation and data-perturbation. Methods for explaining GNNs usually learn a parametric *graph explainer* that can identify a sub-structure (*e.g.*, benzene ring) in the original graph (*e.g.*, molecule) that distinguishes the graph from the graphs of other classes (Luo et al. 2020). In other words, the explainer infers semantics represented by subgraph patterns. In light of this, EPA-GRL is designed as a two-stage approach. At the pre-training stage, it learns a graph explainer using a handful of class labels. At the representation learning stage, for each input graph, EPA-GRL uses the explainer to extract a semantic subgraph and perturbs the rest of the original graphs (*i.e.*, marginal subgraph). The semantics subgraph and the perturbed marginal subgraph are combined to form an augmented graph, which is fed to a contrastive learning framework for representation learning. In this way, EPA-GRL uses a few labeled graphs and relatively more unlabeled graphs, establishing a novel label-efficient semi-supervised GRL framework. In summary, the main contributions are as follows.

- We identify a key limitation of the existing GRL augmentation methods as per the criteria – *semantics-preservation* and *data-perturbation* – of data augmentation.
- This work is the first to explore the potential of a few class labels in semantics-preservation for GRL. We propose a new semi-supervised GRL framework with a novel augmentation method EPA, which introduces XAI to GRL for semantics-preserving perturbation.
- We show theoretically, via an analytical example, that by operating on the output embeddings of a GRL model, the accuracy gap of an empirical risk minimizer under semantics-preserving and semantics-agnostic augmentations can be arbitrarily large.
- We conduct experiments on 6 benchmark datasets with extensive comparison of augmentation methods and contrastive learning frameworks. The results validate the effectiveness of the proposed EPA-GRL method, especially when labeled data are limited.

Related Work

Graph Contrastive Learning. Contrastive learning on graphs includes node-level (Zhu et al. 2020, 2021b) and graph-level tasks (Sun et al. 2020; You et al. 2020, 2021; Suresh et al. 2021; Yin et al. 2022; Xia et al. 2022). This study focuses on graph-level tasks. GraphCL (You et al. 2020) performs graph contrastive learning using four different data augmentation methods: node dropping, edge dropping, subgraph sampling, and attribute masking; JOAO (You et al. 2021) propose a unified bi-level optimization framework that automatically selects the suitable data augmentation method from GraphCL; AD-GCL (Suresh et al. 2021) utilizes adversarial augmentation methods to prevent GNNs from capturing redundant information from the original graph during training; AutoGCL (Yin et al. 2022) uses learnable graph view generators guided by an automated augmentation strategy to introduce appropriate augmentation variances during contrastive learning; SimGRACE (Xia et al. 2022) uses different graph encoders as generators of contrastive graphs and compares the semantic similarity between graphs obtained from the perturbed encoders for contrastive learning. DRGCL (Ji et al. 2024) generates augmented graphs by randomly retaining certain representation dimensions and refines them using learnable, dimension-specific weights. CI-GCL (Tan et al. 2024) proposes a community-invariant contrastive learning framework by leveraging community information to construct positive pairs.

Explainable Graph Neural Networks. Explainability in GNNs has gained significant attention, with various methods proposed to enhance transparency in graph-based tasks (Ying et al. 2019; Luo et al. 2020; Yuan et al. 2020, 2021; Wang and Shen 2023; Xie et al. 2022; Ma et al. 2022; Miao et al. 2023; Fang et al. 2023; Zheng et al. 2024; Chen et al. 2024). These approaches improve understanding of GNN decisions at both instance (Ying et al. 2019; Luo et al. 2020; Wang et al. 2021a; Zhang, Luo, and Wei 2023; Chen et al. 2024) and model levels (Yuan et al. 2020; Wang and Shen 2023; Shin, Kim, and Shin 2024). Early methods like Saliency Maps (Baldassarre and Azizpour 2019) and Grad-CAM (Pope et al. 2019) relied on gradients, while recent advances introduced perturbation-based methods (Luo et al. 2020; Wang et al. 2021a), surrogate models (Vu and Thai 2020; Duval and Malliaros 2021), and generation-based approaches (Yuan et al. 2020; Shan et al. 2021; Wang and Shen 2023). Perturbation-based methods, such as GNNExplainer (Ying et al. 2019), PGExplainer (Luo et al. 2020), Refine (Wang et al. 2021a), MixupExplainer (Zhang, Luo, and Wei 2023), and ProxyExplainer (Chen et al. 2024), generate explanations by perturbing graph features or structures to identify the most important components influencing predictions. Surrogate models (Vu and Thai 2020) approximate the original GNN with simpler models to explain local predictions, while generation-based methods (Yuan et al. 2020; Wang and Shen 2023) leverage generative models to provide both instance-level and model-level explanations.

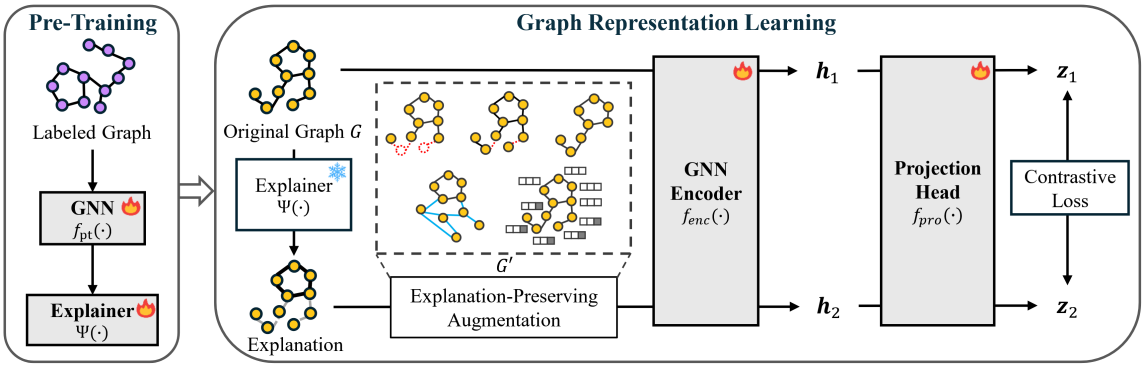


Figure 2: The Architecture of the proposed EPA-GRL method. We first pretrain a GNN model $f_{\text{pt}}(\cdot)$ and its explainer $\Psi(\cdot)$ with a small number of labeled training samples. Then in the GRL step, we use the frozen explainer $\Psi(\cdot)$ to produce augmented graphs to train a GNN encoder $f_{\text{enc}}(\cdot)$ and a projection head $f_{\text{pro}}(\cdot)$ with a contrastive loss. The output of the GNN encoder $f_{\text{enc}}(\cdot)$ will be used as graph representations.

Some recent works have attempted to use explanation to improve learning performance (Shi, Zhou, and Liu 2023; Wang et al. 2021b). For instance, ENGAGE (Shi, Zhou, and Liu 2023) proposes a Smoothed Activation Map to identify important nodes based on representation distributions and uses this to guide graph augmentation in contrastive learning. However, without access to semantic supervision, such unsupervised explanations can only capture structural patterns (e.g., nodes with similar local neighborhoods or high connectivity) that may not align with class-discriminative features. In contrast, our work leverages supervised explanation with limited labels to identify and preserve truly class-relevant semantic structures during augmentation.

Notations and Problem Formulation

A graph G is defined by: i) a node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, where n is the number of nodes; ii) an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$; iii) a feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, where the i-th row $\boldsymbol{X}_i$ is the d-dimensional feature of node v_i; and iv) an adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n \times n}$, where $A_{i,j} = 1$ if $(v_i, v_j) \in \mathcal{E}$. Additionally, each graph is associated with a label $Y \in \mathcal{Y}$, where $\mathcal{Y}$ is a finite set.

Formally, we assume a pair of training sets, a (small) labeled set $\mathcal{T}_\ell = \{(G_i, Y_i) | i \in [M]\}$, and a (large) unlabeled set $\mathcal{T}_u = \{G_i | i \in [N]\}$, where $M \ll N$. Our objective is to leverage the small number of labeled graphs in $\mathcal{T}_\ell$ and relatively more unlabeled graphs in $\mathcal{T}_u$ to perform semantics-preserving representation learning. Similar to the supervised contrastive learning methods (Khosla et al. 2020; Ji et al. 2023), labels from a classification task are leveraged for model training. However, *our problem is substantially different from the existing works by only using a few labels, leading to a constrained semi-supervised GRL problem.*

The Proposed Method

In this section, we introduce the proposed EPA-GRL method. Fig. 2 is an overview. First, EPA-GRL pre-trains an explainer using the labeled graphs in $\mathcal{T}_\ell$, where the explainer $\Psi(\cdot) : G \mapsto G^{(\text{exp})}$ takes a graph G as input and outputs an explanation subgraph $G^{(\text{exp})}$ which is class-specific. Then, a semantic-preserving stochastic mapping $P_{G'|G}$ transforms the original graph G to an augmentation G' such that $G^{(\text{exp})} \subseteq G'$, where the mapping can perform random perturbations on the non-explanatory (*i.e.*, marginal) part of the input G, *i.e.*, $G \setminus G^{(\text{exp})}$; Finally, the augmented graph G' is fed to an encoder $f_{\text{enc}} : G' \mapsto \boldsymbol{h}$, where $\boldsymbol{h} \in \mathbb{R}^{d_e}$ is the d_e-dimensional embedding of G'. f_{enc} is trained via contrastive learning on both augmentations of the unlabeled training set $\mathcal{T}_u$ and the labeled training set $\mathcal{T}_\ell$.

Pre-Training GNN Explainer

In this step, the goal is to train a GNN explainer to identify the most responsible subgraph for its predictions. Extracting such key substructures enables preservation of core semantics of the input graph when perturbing the rest of the graph. We begin by training a GNN $f_{\text{pt}}(\cdot)$ using the labeled training set. The GNN learns to capture both structural and feature-based information necessary to distinguish different graph classes. The GNN is trained by minimizing the cross-entropy loss between the predicted label $f_{\text{pt}}(G)$ and the ground-truth label y. Formally, the optimization problem is:

$$\arg\min_{f_{\text{pt}}} \left(\sum\nolimits_{(G,y) \in \mathcal{T}_\ell} - y \log(f_{\text{pt}}(G)) \right). \tag{1}$$

Next, we propose to use a GNN explainer to extract subgraphs that retain the semantics necessary for classification. Augmentations can then be achieved by making controlled perturbations to rest, marginal parts of the graph. Formally, given a graph $G = (\boldsymbol{A}, \boldsymbol{X})$, the explainer generates an explanation subgraph $G^{(\text{exp})} = (\boldsymbol{A} \odot \boldsymbol{M}, \boldsymbol{X})$, where $\boldsymbol{M} \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ is a binary mask and each entry $M_{ij} = 1$ indicates edge (i,j) is retained in the subgraph $G^{(\text{exp})}$. To improve efficiency, we use a generative explainer, $\Psi(\cdot)$, which employs a parametric neural network to learn the mask $\boldsymbol{M}$ based on node embeddings (Luo et al. 2020, 2024). The explainer $\Psi(\cdot)$ is trained on the labeled dataset $\mathcal{T}_\ell$ and can be directly applied to the unlabeled graphs in $\mathcal{T}_u$ to generate new explanation subgraphs.

The objective of the GNN explainer $\Psi(\cdot)$ is to find a subgraph, denoted by $\Psi(G)$, that balances semantic preservation and compactness. This is achieved by following the Graph Information Bottleneck (GIB) principle, which has been widely used in GNN explanation methods (Yu et al. 2021; Xu et al. 2021; Suresh et al. 2021; Yin et al. 2022). The GIB principle states that an optimal subgraph should retain sufficient information for a prediction task while being as compact as possible, to avoid overfitting or including irrelevant parts of the graph. The learning objective for the explainer is defined as:

$$\arg\min_{\Psi} \left(\sum\nolimits_{(G,y)\in\mathcal{T}_\ell} \mathrm{CE}(Y; f_{\mathrm{pt}}(\Psi(G))) + \lambda|\Psi(G)| \right), \tag{2}$$

where $\mathrm{CE}(Y; f_{\mathrm{pt}}(\Psi(G)))$ is the cross-entropy loss between the label and the prediction on the subgraph $\Psi(G)$, and $|\Psi(G)|$ is the size of the subgraph, which can be measured by the number of edges or the sum of edge weights. The hyper-parameter λ controls the trade-off between the terms for information preservation and structural compactness.

Explanation-Preserving Augmentation Enhanced Graph Representation Learning

Data augmentation is essential in contrastive learning, which generates multiple augmented views of the same graph for learning invariant representations. The success of graph data augmentation attributes to its ability to preserve the core semantics of the graph while introducing variances that facilitate robust representation learning. However, a major limitation of the existing methods is their unconstrained perturbation techniques that may arbitrarily modify the graph structure or node features, and inadvertently disconnect important substructures, leading to a significant loss of semantic information.

Explanation-Preserving Augmentation. To address this limitation, we propose an EPA strategy that explicitly retains the essential part of the graph regarding its class. Specifically, we use the pre-trained graph explainer $\Psi(\cdot)$ to extract an explanation subgraph $G^{(\mathrm{exp})} = \Psi(G)$, which contains the most relevant substructures to the graph's semantics, and $G^{(\mathrm{exp})}$ will be kept intact. The remaining part of the graph, denoted as the *marginal subgraph* $\Delta G = G \setminus G^{(\mathrm{exp})}$, is perturbed to introduce necessary variance for contrastive learning. Next, we introduce EPA-based methods for graph-structured data and discuss their intuitive priors. Detailed algorithms are provided in Appendix B.

- **Node Dropping** randomly removes a subset of nodes and their edges from the marginal subgraph ΔG. The assumption is that removing irrelevant nodes has a small impact on the core semantics of the graph. Each node's dropping probability follows an i.i.d. uniform distribution.
- **Edge Dropping** modifies the connectivity in G by randomly dropping edges in the marginal subgraph ΔG. It assumes the semantic meaning of the graph is robust to the changes of inessential edges. Each edge dropping follows an i.i.d. uniform distribution.
- **Attribute Masking** hides a subset of node or edge attributes in the marginal subgraph ΔG. The assumption is related to node dropping — missing unimportant node attributes has minor impacts on the recovery of essential semantics by the explanation subgraph.
- **Subgraph** further samples a subgraph from the marginal subgraph ΔG based on the assumption that sampling a connected subgraph in ΔG varies the graph structure but does not disrupt the overall semantic integrity.
- **Mixup** randomly selects another graph $\tilde{G}$ from the batch, and combines its marginal subgraph $\Delta\tilde{G}$ with the explanation subgraph $G^{(\mathrm{exp})}$ of the original graph G.

Graph Representation Learning. After generating the augmentations, we employ a contrastive learning framework to learn graph representations. Our EPA approach is versatile and can be integrated into various graph contrastive learning methods. Here, we demonstrate its flexible applicability with two popular techniques: GraphCL (You et al. 2020) and SimSiam (Chen and He 2021).

For each graph G we generate an augmented view G' with our EPA method. These graphs are then passed to a graph encoder $f_{\mathrm{enc}}(\cdot)$, which can be any suitable GNN architecture. The encoder produces graph-level representations $\boldsymbol{h}_1 = f_{\mathrm{enc}}(G)$ and $\boldsymbol{h}_2 = f_{\mathrm{enc}}(G')$ for the two views. Following the approach in (Chen et al. 2020), a projection head, implemented as an MLP, is adopted to obtain new representations for defining the self-supervised learning loss. Specifically, we compute $\boldsymbol{z}_1 = f_{\mathrm{pro}}(\boldsymbol{h}_1)$ and $\boldsymbol{z}_2 = f_{\mathrm{pro}}(\boldsymbol{h}_2)$. In line with the previous works (Chen et al. 2020; Chen and He 2021), $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are used exclusively for model training. For downstream tasks, we discard the projection head and utilize $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ as the graph representations.

GraphCL (You et al. 2020) is a contrastive learning framework developed for graphs with the aim of maximizing the mutual information between different augmented views of the same graph. It uses a noise-contrastive estimation approach to differentiate between positive and negative samples. Given a batch of N graphs, we re-annotate $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ as $\boldsymbol{z}_{i,1}$, $\boldsymbol{z}_{i,2}$ for the i-th graph in the minibatch. Then the contrastive loss (Zhu et al. 2021a) is:

$$\ell_i^{(\mathrm{graphcl})} = -\frac{1}{2}\left(\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_{i,1}, \boldsymbol{z}_{i,2})/\tau)}{\sum_{j=1}^{N} \exp(\mathrm{sim}(\boldsymbol{z}_{i,1}, \boldsymbol{z}_{j,2})/\tau)} + \log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_{i,2}, \boldsymbol{z}_{i,1})/\tau)}{\sum_{j=1}^{N} \exp(\mathrm{sim}(\boldsymbol{z}_{i,2}, \boldsymbol{z}_{j,1})/\tau)} \right) \tag{3}$$

where τ denotes the temperature parameter, $\mathrm{sim}(\cdot,\cdot)$ is the cosine similarity function. The final loss is computed across all positive pairs in the batch.

SimSiam (Chen and He 2021) learns representations by maximizing the similarity between differently augmented views of the same sample. Unlike GraphCL, it doesn't rely on negative samples. The SimSiam framework processes two augmented views of a graph through the same encoder network. After encoding, SimSiam applies an MLP predictor to one view and a stop-gradient operation to the other view. The model then maximizes the similarity between these two processed views. Specifically, for two augmented views of a graph, we have:

$$\boldsymbol{p}_1 = \mathrm{MLP}(\boldsymbol{z}_1) \qquad \boldsymbol{p}_2 = \mathrm{MLP}(\boldsymbol{z}_2), \tag{4}$$

where $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are the encoded graph representations of the two views. $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are their respective predictions after passing through the MLP. The objective is to minimize their negative cosine similarity:

$$\begin{aligned}\mathcal{D}(\boldsymbol{p}_1, \text{stopgrad}(\boldsymbol{z}_2)) &= -\frac{\boldsymbol{p}_1}{\|\boldsymbol{p}_1\|_2} \cdot \frac{\text{stopgrad}(\boldsymbol{z}_2)}{\|\text{stopgrad}(\boldsymbol{z}_2)\|_2} \\ \mathcal{D}(\boldsymbol{p}_2, \text{stopgrad}(\boldsymbol{z}_1)) &= -\frac{\boldsymbol{p}_2}{\|\boldsymbol{p}_2\|_2} \cdot \frac{\text{stopgrad}(\boldsymbol{z}_1)}{\|\text{stopgrad}(\boldsymbol{z}_1)\|_2}.\end{aligned} \tag{5}$$

The stop-gradient (stopgrad) operation is a key component of SimSiam. It blocks the gradients of $\text{stopgrad}(\boldsymbol{z}_1)$ and $\text{stopgrad}(\boldsymbol{z}_2)$ during backpropagation, treating these terms as constants when computing gradients. This process prevents direct optimization of the encoder through these paths, which is crucial for avoiding trivial solutions and encouraging the model to learn meaningful representations. The objective function is implemented as follows:

$$\ell^{(\text{simsiam})} = \frac{1}{2}\mathcal{D}(\boldsymbol{p}_1, \text{stopgrad}(\boldsymbol{z}_2)) + \frac{1}{2}\mathcal{D}(\boldsymbol{p}_2, \text{stopgrad}(\boldsymbol{z}_1)). \tag{6}$$

Theoretical Analysis

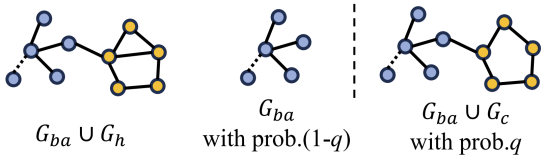
To investigate the importance of preserving semantics in augmented graphs for GRL, in this section, we theoretically analyze the errors of classifying graph embeddings produced by (1) an encoder trained with semantics-preserving augmentations, denoted by $f_{\text{enc}}^{\text{sp}}(\cdot)$; and (2) an encoder trained with semantics-agnostic augmentations, denoted by $f_{\text{enc}}^{\text{sa}}(\cdot)$. Our main theorem (Theorem 1) shows that in certain classification scenarios, the error under $f_{\text{enc}}^{\text{sp}}(\cdot)$ is close to zero, whereas for $f_{\text{enc}}^{\text{sa}}(\cdot)$, the error is close to $\frac{1}{2}$, which is equivalent to random guessing.

To demonstrate our analysis, we modify the widely studied benchmark BA2-Motifs (Luo et al. 2020). As shown in Fig. 3, the dataset has two classes of graphs. The first class consists of graphs generated by taking the union of a Barabási-Albert (BA) graph (Albert and Barabási 2002) and a house motif. The second class also has a BA graph as the base, which is optionally attached to a cycle motif with probability $q \in (0, 1)$. Formally, let $P_{G_{ba}}$ be a probability distribution of BA graphs, let G_h represent the house motif subgraph, with five nodes and six edges, and let G_c represent the cycle motif with five nodes and five edges. Graphs with label 0 are of the form $G_{ba} \cup G_h$, *i.e.*, attach G_h to G_{ba}, where $G_{ba} \sim P_{G_{ba}}$. Graphs with label 1 can either be G_{ba} or $G_{ba} \cup G_c$, with probabilities $1 - q$ and q, respectively.

We consider two types of edge-dropping-based augmentations as follows.

- **Semantics-Agnostic Augmentation ($P^{sa}_{G'|G}$).** Each edge in a graph is dropped independently with a probability $p \in (0, 1)$.
- **Semantics-Preserving Augmentation ($P^{sp}_{G'|G}$).** Each edge in the BA graph G_{ba} and the cycle motif G_c is dropped with probability p, but the edges of the house motif G_h are left unchanged.

Next, we use *empirical contrastive learner* (ECL), which is a generalized notion of the aforementioned encoders $f_{\text{enc}}^{\text{sp}}$ and $f_{\text{enc}}^{\text{sa}}$, to analyze the downstream classification errors using their output embeddings. Note that the proposed EPA-GRL method is an instance of the $f_{\text{enc}}^{\text{sp}}$ encoder.



(a) Label 0 (b) Label 1

Figure 3: Exemplar graphs in modified BA-2motifs.

An ECL aims to learn similar embeddings of graphs that are deemed similar by some selected distance measure. Equivalently, suppose the graphs can be partitioned into at most $\kappa = 2$ clusters based on a distance measure d_c, the goal of an ECL is to learn graph embeddings such that graphs within the same cluster have similar embeddings. To quantify the similarity of graph pairs, we define a distance measure $d_c(G, G')$ as the absolute difference in the number of cycles in graphs G and G'. For instance, $d_c(G_c, G_h) = 2$ since the cycle motif has one cycle and the house motif has three cycles. With this distance measure as an instance, the following theorem, whose proof is deferred to Appendix A , shows that under semantics-preserving augmentation $P^{sp}_{G'|G}$, the ECL can successfully learn graph embeddings such that the same labeled graphs belong to the same cluster. Using such embeddings for classification leads to zero classification error. In contrast, under semantics-agnostic augmentation $P^{sa}_{G'|G}$, the ECL fails to learn graph embeddings with such a property, leading to an error rate close to $\frac{q}{2}$.

Theorem 1. *In the modified BA-2Motifs classification task described above, consider the ECLs f^{sa}_{enc} and f^{sp}_{enc}, which correspond to the semantic-agnostic augmentation $P^{sa}_{G'|G}$ and the semantic-preserving augmentation $P^{sp}_{G'|G}$, respectively, with edge-drop probability $p > 0.3$, and as the size of the unlabeled training set $\mathcal{T}_u$ grows asymptotically large, the following hold:*
i) The error rate of an empirical risk minimization (ERM) operating on the embeddings from $f_{\text{enc}}^{\text{sa}}(G)$ converges to $\frac{q}{2}$.
ii) The error rate of an ERM operating on the embeddings from $f_{\text{enc}}^{\text{sp}}(G)$ converges to 0.

This theorem suggests that semantics-preserving augmentations can endow the ECL with the capability of producing embeddings that facilitate perfect classification in the asymptotic regime, while semantics-agnostic augmentations inevitably lead to significantly higher error rates. Although the theorem is demonstrated on a benchmark dataset, the underlying principles are potentially extendable to more real-world scenarios, where critical substructures determine graph labels (Ying et al. 2019; Yuan et al. 2021).

Experiments

In this section, we conduct extensive experiments to evaluate EPA-GRL and compare it with the widely used augmentation methods and the state-of-the-art (SOTA) GRL methods.

Augmentation Method		MUTAG	Benzene	Alkane-Car.	Fluoride-Car.	D&D	PROTEINS
Node Dropping	Vanilla	$0.803_{\pm 0.030}$	$\mathbf{0.767}_{\pm 0.049}$	$0.965_{\pm 0.038}$	$0.648_{\pm 0.067}$	$\mathbf{0.653}_{\pm 0.070}$	$0.728_{\pm 0.073}$
	EPA	$\mathbf{0.860}_{\pm 0.020}$	$0.765_{\pm 0.050}$	$\mathbf{0.979}_{\pm 0.020}$	$\mathbf{0.656}_{\pm 0.066}$	$0.649_{\pm 0.065}$	$\mathbf{0.744}_{\pm 0.077}$
Edge Dropping	Vanilla	$0.858_{\pm 0.027}$	$0.753_{\pm 0.043}$	$0.942_{\pm 0.047}$	$0.659_{\pm 0.044}$	$0.660_{\pm 0.048}$	$0.702_{\pm 0.077}$
	EPA	$\mathbf{0.861}_{\pm 0.053}$	$\mathbf{0.754}_{\pm 0.050}$	$\mathbf{0.948}_{\pm 0.026}$	$\mathbf{0.662}_{\pm 0.053}$	$\mathbf{0.665}_{\pm 0.050}$	$\mathbf{0.757}_{\pm 0.055}$
Attribute Masking	Vanilla	$0.820_{\pm 0.064}$	$\mathbf{0.762}_{\pm 0.052}$	$0.967_{\pm 0.032}$	$0.653_{\pm 0.071}$	$0.616_{\pm 0.060}$	$0.683_{\pm 0.077}$
	EPA	$\mathbf{0.850}_{\pm 0.047}$	$0.750_{\pm 0.032}$	$\mathbf{0.975}_{\pm 0.027}$	$\mathbf{0.658}_{\pm 0.046}$	$\mathbf{0.624}_{\pm 0.048}$	$\mathbf{0.715}_{\pm 0.057}$
Subgraph	Vanilla	$0.842_{\pm 0.038}$	$0.762_{\pm 0.034}$	$0.973_{\pm 0.027}$	$\mathbf{0.655}_{\pm 0.044}$	$0.651_{\pm 0.060}$	$0.704_{\pm 0.077}$
	EPA	$\mathbf{0.846}_{\pm 0.037}$	$\mathbf{0.765}_{\pm 0.056}$	$\mathbf{0.987}_{\pm 0.019}$	$0.644_{\pm 0.059}$	$\mathbf{0.663}_{\pm 0.065}$	$\mathbf{0.728}_{\pm 0.082}$
Mixup	Vanilla	$0.850_{\pm 0.024}$	$0.766_{\pm 0.040}$	$0.971_{\pm 0.019}$	$0.650_{\pm 0.050}$	$\mathbf{0.643}_{\pm 0.040}$	$0.728_{\pm 0.072}$
	EPA	$\mathbf{0.852}_{\pm 0.027}$	$\mathbf{0.769}_{\pm 0.061}$	$\mathbf{0.975}_{\pm 0.020}$	$\mathbf{0.661}_{\pm 0.060}$	$0.640_{\pm 0.032}$	$\mathbf{0.750}_{\pm 0.066}$

Table 1. Comparison of different graph augmentation methods using GraphCL as the GRL framework.

Augmentation Method		BA-2motifs	HIV	REDDIT-B.
Node Dropping	Vanilla	$0.739_{\pm 0.126}$	$0.634_{\pm 0.032}$	$0.779_{\pm 0.024}$
	EPA	$\mathbf{0.874}_{\pm 0.148}$	$\mathbf{0.643}_{\pm 0.028}$	$\mathbf{0.784}_{\pm 0.039}$
Edge Dropping	Vanilla	$0.603_{\pm 0.127}$	$\mathbf{0.640}_{\pm 0.034}$	$0.782_{\pm 0.045}$
	EPA	$\mathbf{0.701}_{\pm 0.197}$	$0.638_{\pm 0.030}$	$\mathbf{0.787}_{\pm 0.031}$
Attribute Mask	Vanilla	-	$0.621_{\pm 0.017}$	$0.816_{\pm 0.022}$
	EPA	-	$\mathbf{0.632}_{\pm 0.015}$	$\mathbf{0.825}_{\pm 0.026}$
Subgraph	Vanilla	$0.781_{\pm 0.145}$	$0.638_{\pm 0.033}$	$0.779_{\pm 0.036}$
	EPA	$\mathbf{0.786}_{\pm 0.180}$	$\mathbf{0.640}_{\pm 0.018}$	$\mathbf{0.786}_{\pm 0.035}$
Mixup	Vanilla	$0.691_{\pm 0.164}$	$0.632_{\pm 0019}$	$0.777_{\pm 0.026}$
	EPA	$\mathbf{0.694}_{\pm 0.143}$	$\mathbf{0.642}_{\pm 0.016}$	$\mathbf{0.784}_{\pm 0.025}$

Table 2. Comparison of different graph augmentation methods on synthetic and large datasets.

Experimental Setup

Datasets. To evaluate the performance of EPA-GRL, we use eight benchmark real-world datasets with graph-level labels, including MUTAG (Luo et al. 2020), Benzene (Agarwal et al. 2023), Alkane-Carbonyl (Agarwal et al. 2023), Fluoride-Carbonyl (Agarwal et al. 2023), D&D (Dobson and Doig 2003), and PROTEINS (Dobson and Doig 2003; Borgwardt et al. 2005). Among them, MUTAG, Benzene, Alkane-Carbonyl and Fluoride-Carbonyl also provide the ground truth subgraphs that explain the classification of every graph instance, *i.e.*, the semantics pattern. This information will be used for our analysis of semantics-preservation. Statistics and descriptions of the datasets are given in Appendix C.1.

Baselines. The key contribution of this work is a novel graph augmentation method EPA, which is agnostic to the choice of the GRL method. Therefore, in the experiment, we consider two widely used contrastive learning algorithms, *i.e.*, GraphCL (You et al. 2020) and SimSiam (Chen and He 2021), as the basic GRL method on augmented graphs, and compare EPA with its semantics-agnostic counterparts ("Vanilla"): (1) *Node-Dropping*; (2) *Edge-Dropping*; (3) *Attribute-Masking*; (4) *Subgraph-Sample*; and (5) *Mixup*. Moreover, we compare our EPA enhanced GRL, namely EPA-GRL, with other SOTA approaches for GRL, the detailed configurations and results of these compared GRL methods can be found in Appendix C.2 and Appendix D.6, respectively.

Implementation. We evaluate the classification performance based on the learned embeddings provided by different GRL methods. Specifically, following (You et al. 2020), after training a GNN by a GRL algorithm, the embeddings generated by the GNN are fed to an SVM for classification. Detailed information regarding the implementation of the experiment is delineated in Appendix C.3.

Performance Analysis of Augmentation Methods

We report the mean accuracy of graph classification over 10 random runs. Table 1 summarizes the results of comparing EPA with different augmentation techniques using GraphCL as the GRL framework. From the results, we have several observations. First, different augmentation techniques may be useful to different extents on different datasets, where the methods modifying graph structures (*e.g.*, Subgraph, Mixup, Edge Dropping, etc.) are generally better than Attribute Masking. This is because structural changes imply more variances on graphs than node features, which is a desideratum for augmenting the dataset. Second, in most cases, our method EPA outperform each Vanilla augmentation technique, with up to 7.83% relative improvement (Edge-Dropping on PROTEINS) using GraphCL, indicating its generalizability across various augmentations and datasets, and its GRL-agnositic design for plug-and-play with different GRL methods. Finally, EPA is less sensitive to the loss of semantics caused by random perturbations, such as the degraded accuracy of Vanilla Node Dropping on MUTAG in Table 1. This is because the perturbation is constrained to the marginal subgraphs outside the semantic patterns by EPA, leading to its robustness to the potentially arbitrary changes caused by perturbations.

Moreover, Appendix D.1 presents a comparison of EPA with different augmentation techniques under the SimSiam GRL framework. To enrich the baseline comparisons, we also incorporate the NodeSam augmentation (Yoo, Shim, and Kang 2022) in Appendix D.2. Further, We evaluate its performance with Graph Isomorphism Network (GIN) (Xu et al. 2019) as another base GNN model in Appendix D.3. Finally, we adopt Refine (Wang et al. 2021a) as the explainer, with results reported in Appendix D.4. These experiments demonstrate the generalizability of EPA across different datasets, GNN architectures, augmentation methods, and explainers.

Experiments on Synthetic and Large Datasets

To further demonstrate the robustness of EPA on different types of data, we conduct experiments with the GraphCL framework on a synthetic dataset BA-2motifs (Luo et al. 2020), a real-world dataset HIV (Luong and Singh 2024)

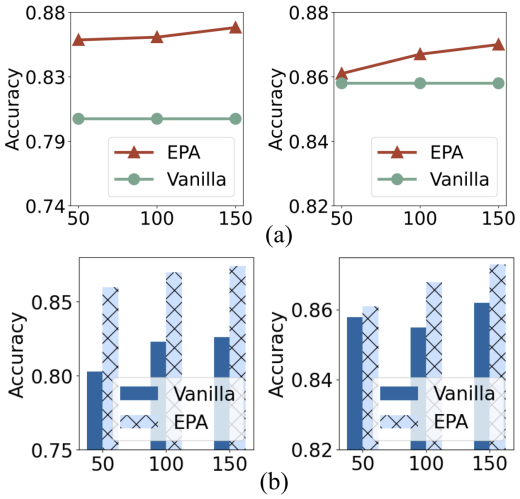


Figure 4: Accuracy with different numbers of (a) # pre-training samples and (b) # downstream training samples.

with a large number of graphs, and a real-world dataset REDDIT-BINARY (Yanardag and Vishwanathan 2015) with large scale graphs. The results are shown in Table 2. We find that our method achieves the best results across almost all data augmentation methods. Specifically, on the BA-2motifs dataset, our method improves by an average of 17.3% compared to Vanilla. Note that, since the node features in BA-2motifs have only one dimension, the feature masking augmentation method cannot be applied.

Ablation Study

In this section, we use MUTAG dataset for ablation study. To understand EPA's label-efficiency, we change the number of labeled graphs for explainer pre-training, *i.e.*, $|\mathcal{T}_\ell|$, within $\{50, 100, 150\}$ while fixing the number of training graphs for SVM as 50. Fig. 4(a) summarizes the results of using Node/Edge Dropping as the base perturbation in EPA. Extensive ablation studies of other perturbation methods are deferred to Appendix D.7. From Fig. 4(a), as $|\mathcal{T}_\ell|$ increases, EPA-GRL gains higher accuracy with different perturbation methods. The "Vanilla" method has constant results because it is semantics-agnostic. As such, EPA is capable of making effective use of more labels for better augmentations. However, as labeling is usually expensive, we restrict most of our experiments to the challenging region with only 50 labeled graphs. Additionally, we evaluate the impacts of varying numbers of labeled graphs for the downstream training of SVM. As shown in Fig. 4(b) (and Appendix D.7), both "Vanilla" and EPA augmented graphs enable better graph embeddings for effective downstream training with more labels. In particular, EPA-GRL consistently produce better embeddings than the baseline method as indicated by the clear accuracy gap.

Impact of Explanation Quality

To analyze the relationship between explanation quality and model performance, we evaluate how fidelity of explanation subgraphs influences classification accuracy. We employ two theoretically grounded fidelity metrics, $Fid_{\alpha_1,+}$ and $Fid_{\alpha_2,-}$, proposed by (Zheng et al. 2024). The metric formulations are given in Appendix D.8. A higher $Fid_{\alpha_1,+}$ or a lower $Fid_{\alpha_2,-}$ indicates better explanation quality.

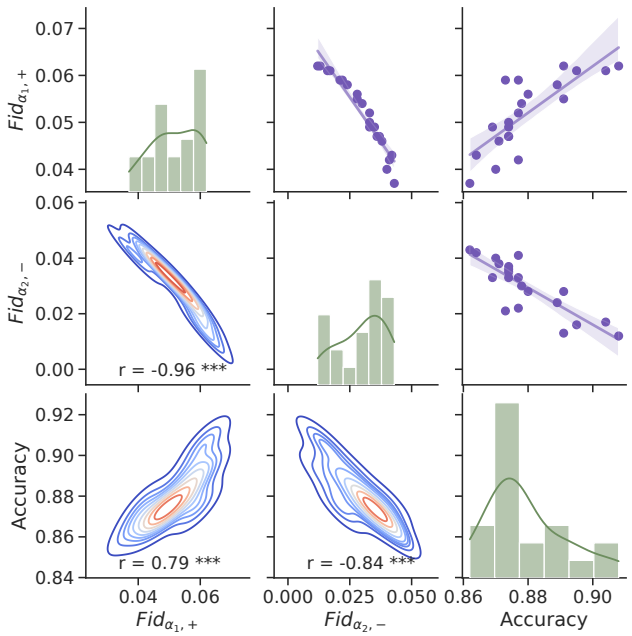


Figure 5: Correlation between $Fid_{\alpha_1,+}$, $Fid_{\alpha_2,-}$, and Accuracy on MUTAG dataset. The value r represents the Pearson correlation coefficient. Statistical significance is denoted by $*$, with $***$ indicating a p-value of $p \leq 0.01$ for testing non-correlation.

To systematically evaluate this relationship, we introduce controlled perturbations to the explanation subgraphs generated by our trained explainer, following the setting in (Zheng et al. 2024). For each explanation subgraph, we randomly remove a proportion β of its edges and replace them with the same number of randomly selected non-explanation edges, where β ranges from 0 to 1 in increments of 0.05. Fig. 5 presents the correlation between both fidelity metrics and classification accuracy on the MUTAG dataset. The strong correlation between the fidelity metrics and classification performance shows that higher-quality explanations lead to better representation learning, validating our approach of preserving semantic structures for effective graph augmentation.

Conclusion

In this paper, we study data augmentation methods for graph contrastive learning. In contrast to most of the existing methods, which focus on structural perturbations but overlook the importance of preserving semantic information, we propose a novel framework EPA-GRL to deal with both. EPA-GRL incorporates the explanatory patterns of a graph into its data augmentations by leveraging a few labeled graphs and trains a GRL model with more unlabeld graphs, establishing a semi-supervised learning paradigm. We perform theoretical analysis and conduct comprehensive experiments. The results validate the effectiveness of the proposed method.

Acknowledgments

This project was partially supported by NSF grants IIS-2529283, ECCS-2242700 and CCF-2241057. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Agarwal, C.; Queen, O.; Lakkaraju, H.; and Zitnik, M. 2023. Evaluating explainability for graph neural networks. *Scientific Data*, 10(1): 144.

Albert, R.; and Barabási, A.-L. 2002. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1): 47.

Baldassarre, F.; and Azizpour, H. 2019. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*.

Borgwardt, K. M.; Ong, C. S.; Schönauer, S.; Vishwanathan, S.; Smola, A. J.; and Kriegel, H.-P. 2005. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1): i47–i56.

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*, 1597–1607. PMLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15750–15758.

Chen, Z.; Zhang, J.; Ni, J.; Li, X.; Bian, Y.; Islam, M. M.; Mondal, A.; Wei, H.; and Luo, D. 2024. Generating In-Distribution Proxy Graphs for Explaining Graph Neural Networks. In *ICML*.

Debnath, A. K.; Lopez de Compadre, R. L.; Debnath, G.; Shusterman, A. J.; and Hansch, C. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2): 786–797.

Dobson, P. D.; and Doig, A. J. 2003. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4): 771–783.

Duval, A.; and Malliaros, F. D. 2021. Graphsvx: Shapley value explanations for graph neural networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*, 302–318. Springer.

Fang, J.; Wang, X.; Zhang, A.; Liu, Z.; He, X.; and Chua, T.-S. 2023. Cooperative Explanations of Graph Neural Networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 616–624.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Ji, J.; Jia, H.; Ren, Y.; and Lei, M. 2023. Supervised contrastive learning with structure inference for graph classification. *IEEE Transactions on Network Science and Engineering*, 10(3): 1684–1695.

Ji, Q.; Li, J.; Hu, J.; Wang, R.; Zheng, C.; and Xu, F. 2024. Rethinking dimensional rationale in graph contrastive learning from causal perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12810–12820.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33: 18661–18673.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Luo, D.; Cheng, W.; Xu, D.; Yu, W.; Zong, B.; Chen, H.; and Zhang, X. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33: 19620–19631.

Luo, D.; Zhao, T.; Cheng, W.; Xu, D.; Han, F.; Yu, W.; Liu, X.; Chen, H.; and Zhang, X. 2024. Towards inductive and efficient explanations for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Luong, K.-D.; and Singh, A. K. 2024. Fragment-based pretraining and finetuning on molecular graphs. *Advances in Neural Information Processing Systems*, 36.

Ma, J.; Guo, R.; Mishra, S.; Zhang, A.; and Li, J. 2022. Clear: Generative counterfactual explanations on graphs. *Advances in Neural Information Processing Systems*, 35: 25895–25907.

Miao, S.; Luo, Y.; Liu, M.; and Li, P. 2023. Interpretable Geometric Deep Learning via Learnable Randomness Injection. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Pope, P. E.; Kolouri, S.; Rostami, M.; Martin, C. E.; and Hoffmann, H. 2019. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10772–10781.

Shan, C.; Shen, Y.; Zhang, Y.; Li, X.; and Li, D. 2021. Reinforcement learning enhanced explainer for graph neural networks. *Advances in Neural Information Processing Systems*, 34: 22523–22533.

Shi, Y.; Zhou, K.; and Liu, N. 2023. Engage: Explanation guided data augmentation for graph representation learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 104–121. Springer.

Shin, Y.; Kim, S.; and Shin, W. 2024. PAGE: Prototype-Based Model-Level Explanations for Graph Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(10): 6559–6576.

Sun, F.; Hoffmann, J.; Verma, V.; and Tang, J. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Sun, M.; Xing, J.; Wang, H.; Chen, B.; and Zhou, J. 2021. MoCL: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 3585–3594.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 15920–15933.

Tan, S.; Li, D.; Jiang, R.; Zhang, Y.; and Okumura, M. 2024. Community-invariant graph contrastive learning. *arXiv preprint arXiv:2405.01350*.

Vu, M.; and Thai, M. T. 2020. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33: 12225–12235.

Wang, X.; and Shen, H. 2023. GNNInterpreter: A Probabilistic Generative Model-Level Explanation for Graph Neural Networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Wang, X.; Wu, Y.; Zhang, A.; He, X.; and Chua, T.-S. 2021a. Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, 34: 18446–18458.

Wang, Y.; Min, Y.; Shao, E.; and Wu, J. 2021b. Molecular graph contrastive learning with parameterized explainable augmentations. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1558–1563. IEEE.

Xia, J.; Wu, L.; Chen, J.; Hu, B.; and Li, S. Z. 2022. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*, 1070–1079.

Xie, Y.; Katariya, S.; Tang, X.; Huang, E.; Rao, N.; Subbian, K.; and Ji, S. 2022. Task-agnostic graph explanations. *Advances in Neural Information Processing Systems*, 35: 12027–12039.

Xu, D.; Cheng, W.; Luo, D.; Chen, H.; and Zhang, X. 2021. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 30414–30425.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Yanardag, P.; and Vishwanathan, S. 2015. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1365–1374.

Yin, Y.; Wang, Q.; Huang, S.; Xiong, H.; and Zhang, X. 2022. Autogcl: Automated graph contrastive learning via learnable view generators. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 8892–8900.

Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

Yoo, J.; Shim, S.; and Kang, U. 2022. Model-agnostic augmentation for accurate graph classification. In *Proceedings of the ACM Web Conference 2022*, 1281–1291.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph contrastive learning automated. In *International Conference on Machine Learning*, 12121–12132. PMLR.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph Contrastive Learning with Augmentations. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 5812–5823. Curran Associates, Inc.

Yu, J.; Xu, T.; Rong, Y.; Bian, Y.; Huang, J.; and He, R. 2021. Graph Information Bottleneck for Subgraph Recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Yuan, H.; Tang, J.; Hu, X.; and Ji, S. 2020. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 430–438.

Yuan, H.; Yu, H.; Gui, S.; and Ji, S. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5782–5799.

Yuan, H.; Yu, H.; Wang, J.; Li, K.; and Ji, S. 2021. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*, 12241–12252. PMLR.

Zhang, J.; Luo, D.; and Wei, H. 2023. Mixupexplainer: Generalizing explanations for graph neural networks with data augmentation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3286–3296.

Zheng, X.; Shirani, F.; Wang, T.; Cheng, W.; Chen, Z.; Chen, H.; Wei, H.; and Luo, D. 2024. Towards Robust Fidelity for Evaluating Explainability of Graph Neural Networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Zhu, Y.; Xu, Y.; Liu, Q.; and Wu, S. 2021a. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021b. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*, 2069–2080.