

Graph Data Augmentation with Contrastive Learning on Covariate Distribution Shift

Fanlong Zeng^a, Wensheng Gan^{a,*}

^aSchool of Intelligent Systems Science and Engineering, Jinan University, Zhuhai 519070, China

ARTICLE INFO

Keywords:
graph
out-of-distribution
covariate shift
contrastive learning
data augmentation

ABSTRACT

Covariate distribution shift occurs when certain structural features present in the test set are absent from the training set. It is a common type of out-of-distribution (OOD) problem, frequently encountered in real-world graph data with complex structures. Existing research has revealed that most out-of-the-box graph neural networks (GNNs) fail to account for covariate shifts. Furthermore, we observe that existing methods aimed at addressing covariate shifts often fail to fully leverage the rich information contained within the latent space. Motivated by the potential of the latent space, we introduce a new method called MPAIACL for More Powerful Adversarial Invariant Augmentation using Contrastive Learning. MPAIACL leverages contrastive learning to unlock the full potential of vector representations by harnessing their intrinsic information. Through extensive experiments, MPAIACL demonstrates its robust generalization and effectiveness, as it performs well compared with other baselines across various public OOD datasets. The code is publicly available at <https://github.com/fzeng1/MPAIACL>.

1. Introduction

Graph classification [13, 26] is a fundamental task in real-world graph analysis, distinguished from other classification tasks by its reliance on node and edge representations to capture the complex structure and semantics of entire graphs [13]. Graph neural networks (GNNs) [31] have recently emerged as the key framework for modeling graph-structured data. Currently, GNNs are typically designed under the assumption that the training and test sets are drawn from independent and identically distributed (I.I.D) data [2]. However, this assumption barely holds in real-world scenarios, due to the out-of-distribution (OOD) problem potentially existing during the test stage [26]. Additionally, the OOD problem induces a distribution shift between the training and test sets, leading to a significant degradation in model performance. When applied to datasets with distribution shifts, GNNs typically yield suboptimal overall graph classification performance [7]. As a result, various approaches have been proposed to address this challenge, such as invariant learning [25], architecture design [37], and data augmentation [8].

The distribution shift can be further categorized [26]. For a better understanding of these distribution shift problems, we introduce two important concepts, which are the stable feature and the environment feature. According to previous studies [16], we have the concepts below: i) Stable features, which capture the underlying patterns of the entire graph, providing a robust representation of the graph's intrinsic structure. Based on this, the relationship between stable features and labels can be considered invariant; ii) Environmental features, which are subject to variation for the label [26]. We can consider a toy example of a molecule. As

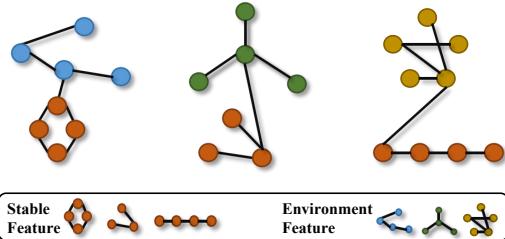
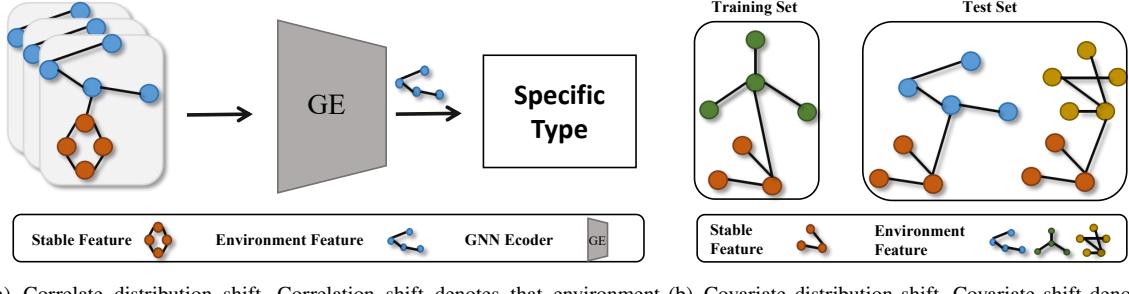


Figure 1: The stable features and environment features. Stable features capture the underlying patterns of the entire graph, providing a robust representation of the graph's intrinsic structure. Environmental features are subject to variation for the label.

illustrated in Fig. 1, the stable feature is a component that can determine the property of a molecule, like functional groups. While the environment features are various and irrelevant to the property. Noticed that functional groups of a molecule are stable enough to determine the property, while scaffolds (environmental features) are irrelevant. Due to the unstable and variability of environmental features, the distribution shifts in the graph can be further categorized into correlation shift and covariate shift: (1) Correlation shift. As illustrated in Fig. 2(a), environment features establish the spurious relation with labels. It means the GNN model learns a spurious relation between environment features and labels. Traditional GNNs assume that the training set encompasses all environment features present in the test set, but in reality, the assumption often gives rise to inconsistent statistical relationships between the training and test sets. (2) Covariate shift. Environment features are very different between the training set and test set [7, 26]. Considering an example in Fig. 2(b), when the test set contains new or unseen environmental features that are not accounted for in the training set,

*Corresponding author

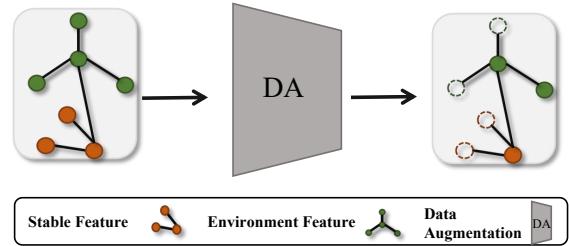
✉ fzeng1@stu.jnu.edu.cn (F. Zeng); wsgan001@gmail.com (W. Gan)
ORCID(s):

**Figure 2:** Correlate distribution shift and covariate distribution shift.

a covariate shift occurs. Covariate shift often happens in a situation where the test set has new environmental features, which do not appear in the training set. This phenomenon is frequently observed in scenarios where the training dataset is of insufficient quantity or variety.

Currently, there are two primary approaches to addressing the OOD problem: (causal) invariant learning and data augmentation [14, 26]. (i) Invariant learning approaches seek to identify a subset of graph features, referred to as the rationale (stable feature), that are most informative and predictive, thereby providing a robust explanation for the model’s predictions. However, invariant learning relies on the assumption that the environment features present during training are identical to those encountered during testing. Therefore, it neglects the potential differences in environmental features between the two phases. In other words, graph invariant learning does not consider the covariate shift situation. (ii) Data augmentation approaches utilize the perturbation at different levels, such as node features, quantity of nodes, edges, or sub-graphs to tackle the OOD problem. However, graph data augmentation often fails to distinguish stable features, inadvertently introducing noise that disrupts the entire graph, ultimately destroying the features that are essential for robust prediction. The traditional graph data augmentation methods are defective in tackling the graph OOD problem in correlation shift distribution. As illustrated in Fig. 3, these methods have unlimited data augmentation strategies, which may destroy the structure of stable features, resulting in a suboptimal performance in the situation of covariate distribution shift.

The primary challenge lies in the fact that most existing methods overlook the covariate shift issue, neglecting its significant impact on model performance. A novel approach, adversarial invariant augmentation (AIA) [26] was proposed, which not only effectively distinguishes between stable features and environmental features but also specifically targets the environmental features for perturbation. Upon re-examining this method, we discover that AIA has an insufficient latent information utilization issue where the boundary between different types of features in the latent space is not sufficiently well-defined.

**Figure 3:** Traditional graph data augmentation. Traditional data augmentation strategies may destroy the structure of stable features.

To further utilize the information in the latent space and mitigate the issue above, we utilize the manifold assumption [30], which posits that similar predictions from the network imply proximity in the manifold, to strengthen the performance of AIA. Therefore, we propose a new method—MPAIACL, for More Powerful AIA using Contrastive Learning in this paper. We leverage contrastive learning to strengthen both the stable features generator and the adversarial augmente. MPAIACL brings the embeddings of stable features closer together in latent space, while simultaneously distancing them from environment features, thereby enhancing the performance of the original model. We validated MPAIACL on various datasets, including Molbbbp, Molbace in OGB datasets [9], and MolHiv in GOOD [7]. Moreover, we validate the effectiveness of our approach across diverse datasets from different domains, varying in size and scaffold, using the same parameter settings as in [26]. Our experimental results demonstrate the superior generalization and effectiveness of our method. In summary, the main contributions of this paper are as follows:

- We uncover the under-explored representation information in AIA, which can be leveraged to further enhance the performance of the original model.
- We adopt contrastive learning to strengthen the original AIA model, strengthening both the stable feature generator and the adversarial augmente.

- We provide a theoretical analysis of our approach, offering insights into its underlying mechanisms and effectiveness.
- Through extensive experiments and analysis, we demonstrate the generalization and effectiveness of MPAIACL.

After the introduction, we reviewed the related work, including graph contrastive learning and graph invariant learning in Section 2. The motivation of MPAIACL is illustrated in Section 3. In Section 4, we illustrate the preliminaries and the notations. We briefly introduce the GNNs used in graph classification and the related definitions of OOD problems. Then we provide the problem statement in this paper. In Section 5, we explain the methodology of MPAIACL and conduct a theoretical analysis. In Section 6, we provide the experiment analysis and the limitations of our method. Finally, we conclude our work in Section 7.

2. Related Work

In this section, we provide a review of the related literature on graph contrastive learning and graph invariant learning. We introduce the related classical methods and then discuss the pros and cons of different methods.

2.1. Graph Contrastive Learning

Contrastive learning is a method that makes the representation of proper transformations of data agree with the positive one and be as far away from the negative one as possible. It is a traditional self-supervised learning method that has yielded impressive results in the field of computer vision [10]. With the rapid growth of interest in graph contrastive learning, a multitude of methods have emerged in recent years. Graph contrastive learning leverages multiple views of varying scales to enhance the embedding representation of similar instances. It can be broadly categorized into two paradigms: intra-scale and inter-scale contrast [13].

Intra-scale contrast refers to the process of contrasting information within the same scale, such as at the local level [17, 43], contextual level [18, 21], or graph level [40, 39]. (1) Local-level contrast focuses on learning node-level representations by comparing and aligning node-centric embeddings, thereby capturing fine-grained differences between individual nodes. GRACE [43] generates two graph views through corruption and learns node representations by maximizing the consistency. GCA [44] captures both topological and semantic aspects of the graph. B2-Sampling [17] learns to correct and refine the labels of error-prone negative pairs during training. (2) Context-level contrast refers to contrasting methods at the subgraph level. GCC [21] captures universal topological properties across multiple networks. MSSGCL [18] introduces a multi-scale subgraph contrastive learning approach that effectively captures fine-grained semantic information. (3) Graph-level contrast employs discrimination between graph representations. GraphCL [39] proposes four distinct graph augmentation methods to generate varied views. JOAO [40] simultaneously refines the

graph augmentation selection and enhances the contrastive objectives. AD-GCL [28] proactively identifies and minimizes redundant information. HGCL [12] explores the hierarchical structural semantics of a graph.

Intra-scale contrast refers to the process of contrasting information across scales, encompassing various types of contrastive relationships, such as local-global contrast [32, 32], local-context contrast [11, 19], and context-global contrast [3, 27]. (1) Local-global contrast seeks to capture the intricate relationships between local and global information, for maximizing the mutual information between these two scales. DGI [32] aligning patch representations with high-level summaries of graphs. CGKS [42] improves generalization ability and incorporates awareness of latent anatomies. (2) Local context contrast focuses on capturing the properties of subgraphs rather than the full graph. SUBG-CON [11] exploits the correlation between central nodes and their associated subgraphs. GIC [19] aims to further enrich graph representations by capturing cluster-level information content. HCHSM [29] integrates multiple levels of intrinsic graph features to capture the hierarchical relationships within the graph. (3) Context-global contrast aims to strengthen the mutual information between subgraph representations and the overall graph representation. SUGAR [27] reconstructs a sketched graph by identifying and extracting striking subgraphs. BiGI [3] encodes the global characteristics of bipartite graphs. MICRO-Graph [41] extracts informative motifs and subsequently utilizes these learned motifs to guide the sampling of informative subgraphs for contrast.

However, a significant limitation of most existing methods is that they overlook the importance of invariant graph features, which can also serve as a valuable source of contrastive information. In our method, we not only consider the stable features but also utilize the latent space information in a contrastive way.

2.2. Graph Invariant Learning

Graph invariant learning is a methodology that identifies and extracts the invariant properties of graphs, which are then leveraged to enhance the model's generalization capabilities. There also exists pioneering work [34] that leverages the invariant principle to identify stable properties to address the OOD generalization problem. For example, GRE [16] separates rationale and environment in latent spaces and performs representation learning on both real and augmented examples. CAL [25] identifies causal patterns in graph data and mitigates the confounding effects of shortcuts. FLAG [14] enhances node features with adversarial perturbations during training. AIA [26] increases the features of the environment and effectively addresses the problem of covariate distribution shift. Beyond the aforementioned methods, other generalization algorithms exist. IRM [2] estimates nonlinear, invariant, and causal predictors from multiple training environments. GroupDRO [23] learns models by minimizing the worst-case training loss across a set of predefined groups. GALA [6] learns invariant

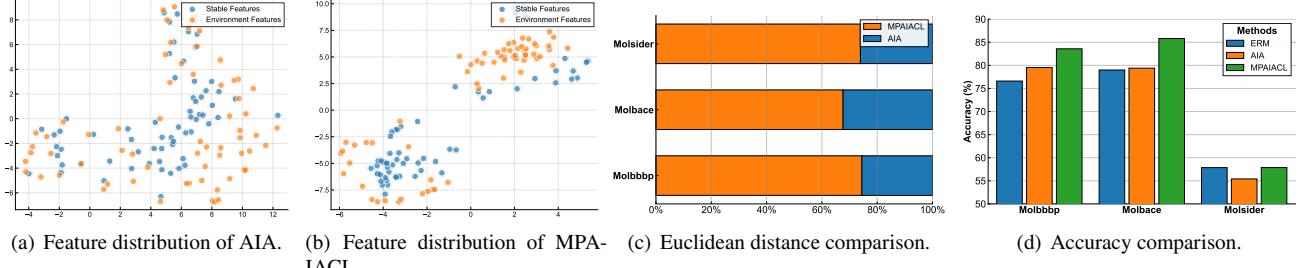


Figure 4: Result of the experiment in section 3. (a) and (b) visualize the stable feature and environment feature distribution in the latent space. (c) Visualize the average Euclidean distance of the stable feature and environment feature between AIA and MPAIACL. (d) Visualize the accuracy comparison between different datasets and methods.

Table 1
Notations with the corresponding explanations.

Notation	Explanation
\mathcal{G}	The graph data.
\mathcal{V}	The node set of a graph.
\mathcal{X}	The feature set of \mathcal{V} in a graph.
\mathcal{A}	The adjacent matrix of a graph.
\mathcal{Y}	The label of the graph.
\mathcal{D}	The dataset that includes various graph.
DS	The wasserstein distance.
\mathcal{L}	The loss value.
$\mathbb{E}(\cdot, \cdot)$	The empirical risk.
$\ell(\cdot, \cdot)$	The loss function.
$P(\cdot, \cdot)$	The probability functions.
$*_{tr}$	The * of training set.
$*_{ts}$	The * of test set.
$*_{std}^{std}, *_{env}^{std}$	The * of stable features.
$*_{DA}^{DA}, *_{env}^{DA}$	The * of augmentation features.
$*_{env}^{env}$	The * of environment features.

graph representations under the guidance of an environment assistant model. However, most existing methods overlook the importance of tackling stable features and environment features from a contrastive perspective. Our method facilitates the convergence of the stable features and promotes the divergence of environmental features.

3. Insufficient Latent Information Utilization Issue

This section introduces the insufficient latent information utilization issue and presents our approach to fully exploiting the potential of the latent space.

3.1. Cause Analysis

The insufficient latent information utilization issue in AIA refers to the phenomenon where the boundary between different types of features in the latent space is not sufficiently well-defined. We analyze this issue from two perspectives: the feature distribution in the latent space and the Euclidean distance between representations.

From the perspective of feature distribution, we refer to Fig. 4(a), which visualizes the distributions of stable and

environmental features in the latent space. The visualization is conducted under a covariate shift in the Molbbbp size domain using AIA. As illustrated in Fig. 4(a), the blue dots represent stable features, while the orange dots correspond to environment features. The boundary between these two types of features remains indistinct, with some features of different types even overlapping or connecting with each other.

From the perspective of Euclidean distance, Fig. 4(c) presents the average Euclidean distance ratio between stable and environmental features within the latent space. The visualization is obtained under covariate shift in the Molbbbp size domain using the AIA framework. As shown in Fig. 4(c), AIA exhibits a significantly smaller distance proportion compared to MPAIACL, suggesting that the stable and environmental features extracted by AIA are more closely aligned in the latent space.

In summary, the core cause of the insufficient latent information utilization issue in AIA lies in the relatively large distance between the stable and environment features within the latent space.

3.2. Solution

In this section, we provide a step-by-step analysis of how our method addresses the insufficient latent information utilization issue in AIA.

Based on the above summary, we hypothesize that leveraging the mutually exclusive relationship between stable features and environment features could further enhance the model's discriminative ability. Therefore, we utilize the manifold assumption [30] and introduce the concept of contrastive learning to enable the model to acquire a stronger discriminative capability. Based on this idea, we propose MPAIACL for More Powerful Adversarial Invariant Augmentation using Contrastive Learning. To fully unleash the potential of the information embedded in the latent space, we employ contrastive learning to push apart the stable and environmental features within this space. At the same time, we leverage contrastive learning to enhance the representation of environment features.

From the perspective of feature distribution in the latent space, Fig. 4(b) illustrates the distributions of stable and environment features under a covariate shift in the Molbbbp

size domain using AIA. As shown in Fig. 4(b), MPAIACL exhibits a more distinct and well-separated boundary between stable and environmental features compared to AIA, indicating improved feature discrimination in the latent space.

From the perspective of Euclidean distance, as shown in Fig. 4(c), the distance proportion of MPAIACL is significantly larger than that of AIA, indicating that the stable and environmental features extracted by MPAIACL are more distinct in the latent space.

Additionally, as shown in Fig. 4(d), the results indicate a positive correlation between the distinctness of stable and environmental feature proportions and the model’s accuracy—that is, the greater the separation between these features, the higher the performance achieved.

4. Preliminaries

In this section, we present the preliminary concepts and definitions that are used in this work. We also introduce the notations used throughout the paper. Additionally, we provide a brief overview of the graph neural network (GNN) architecture employed for graph classification and define the distribution shift. Finally, we present a detailed formulation of the problem statement, highlighting the key challenges and objectives that our proposed approach seeks to address.

In this paper, we focus on supervised graph classification and perform the task on many undirected graphs. An undirected graph is denoted as $\mathcal{G}(\mathcal{V}, \mathcal{X}, \mathcal{A})$, where \mathcal{V} denotes the node sets, \mathcal{X} is a feature set, and \mathcal{A} is an adjacent matrix. \mathcal{Y} is the label set, including the labels of each graph. We also use $\mathcal{D} = \{\mathcal{G}, \mathcal{Y}\}$ to denote datasets, where $\mathcal{D}_{tr} = \{\mathcal{G}^r, \mathcal{Y}^r\}$ denotes a training set, and $\mathcal{D}_{ts} = \{\mathcal{G}^s, \mathcal{Y}^s\}$ denotes a test set. There are two core concepts used in the entire paper, which are i) Stable features, which capture the underlying patterns of the entire graph. The relationship between stable features and labels can be considered invariant. ii) Environment features, which are subject to variation for the label [26]. The notations used throughout this paper are summarized in Table 1.

4.1. Graph Neural Networks

GNNs are a class of deep learning architectures specifically designed to process and analyze graph-structured data. They have demonstrated remarkable effectiveness across a wide range of tasks, including node-level, edge-level, and graph-level classification [33]. In this paper, we focus on the supervised graph classification task. We provide an overview of how GNNs operate for graph classification tasks below:

$$\begin{aligned} \mathbf{h}_v^{(k)} &= \text{Aggregate} \left(\mathbf{h}_v^{(k-1)}, \{\mathbf{h}_u^{(k-1)} \mid u \in \mathcal{N}(v)\} \right). \\ h_{\mathcal{G}} &= \text{READOUT} \left(\{h_v^{(K)} : v \in \mathcal{G}\} \right). \end{aligned} \quad (1)$$

Here, $h_v^{(k-1)}$ represents the embedding of the node v at the $(k-1)$ -th layer, $\mathcal{N}(v)$ denotes the set of neighbors of node v . Aggregate is a function that aggregates the features of the node with its neighbors.

In summary, Graph Neural Networks (GNNs) generally consist of three main steps: message passing, aggregation, and readout. Each node v is initially represented by a feature vector $h_v^{(0)}$. (1) During message passing, nodes exchange information with their neighbors to compute messages. (2) In the aggregation step, each node aggregates the messages from its neighbors and updates its own representation $h_v^{(t+1)}$. (3) Finally, in the readout phase, graph-level representations are derived for downstream tasks. Subsequently, the readout process yields the graph-level representation $h_{\mathcal{G}}$.

4.2. Definition

From the perspective of invariant learning and stable learning [16, 35], a fundamental assumption is that there exist stable features that are determinative of the label in classification tasks [25, 38, 35].

The relationship between stable features and labels provides a foundation for tackling OOD generalization [26]. On the contrary, the other scaffold structure is generally irrelevant to the properties, which can be considered as environmental features [9, 38]. The environmental features don’t have a direct effect on labels. Due to the inherent imprecision in data collection and the limitations of environmental features, inconsistencies inevitably arise between the training set and test set, giving rise to two primary OOD challenges: correlation shift and covariate shift [7].

Definition 1 (Correlation shift). The training set has a delusive correlation between the data and labels, which is not established in the test set. The graph correlation shift can be denoted as:

$$P_{tr}(\mathcal{G}|\mathcal{Y}) \neq P_{ts}(\mathcal{G}|\mathcal{Y}), P_{tr}(\mathcal{G}) = P_{ts}(\mathcal{G}). \quad (2)$$

Equation (2) indicates the presence of false statistical correlations in the training data, which may not hold true in the testing data. Besides, $P_{tr}(\mathcal{G}) = P_{ts}(\mathcal{G})$ indicates that the overall distribution of graph structures in the training and testing sets is the same. Notice that P denotes the probability distribution function.

Definition 2 (Covariate shift). The test set has new environment features, which do not appear in the training set. It may happen due to an insufficient quantity or variety of datasets. The graph covariate shift can be denoted as $P_{tr}(\mathcal{G}|\mathcal{Y}) = P_{ts}(\mathcal{G}|\mathcal{Y}), P_{tr}(\mathcal{G}) \neq P_{ts}(\mathcal{G})$. The equality $P_{tr}(\mathcal{G} \mid \mathcal{Y}) = P_{ts}(\mathcal{G} \mid \mathcal{Y})$ indicates that the statistical correlation between the graph structures and the labels is consistent across the training and testing sets, suggesting that the learned relationships can generalize from the training set to the test set. In contrast, $P_{tr}(\mathcal{G}) \neq P_{ts}(\mathcal{G})$ reveals that there are discrepancies in the graph structures between the training and test sets, which can be attributed to differences in environmental features. We can measure Graph Covariate Shift (GCS) as [26]:

$$GCS(P_{te}, P_{ts}) = \frac{1}{2} \int_S |P_{tr} - P_{ts}| dg. \quad (3)$$

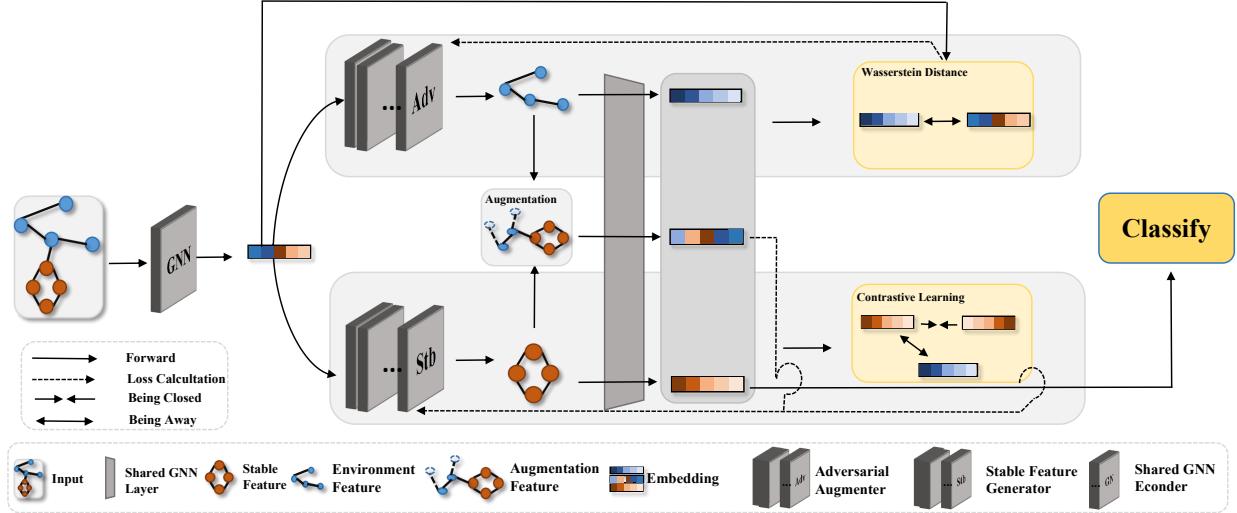


Figure 5: The overview of MPAIACL. The explanation of the notation used in the figure is located at the bottom and the bottom left. The training process consists of two distinct phases: (1) Strengthen the Stable Feature Generator (SFG), and (2) Strengthen the Adversarial Augmenter (AA). Initially, the input graph is processed through a shared GNN encoder, which generates the graph embeddings. In phrase (1), SFG generates the stable features. Then, the stable feature \mathcal{F}_s would make a data augmentation with the environment feature \mathcal{F}_e , which is generated by AA to obtain the augmentation feature \mathcal{F}_a . Subsequently, MPAIACL utilizes contrastive learning to pushing away \mathcal{F}_s from \mathcal{F}_e . Finally, the results of the contrastive and \mathcal{F}_a are used to strengthen the SFG. In phrase (2), \mathcal{F}_e uses Wasserstein distance to push away from \mathcal{F}_s and the original graph embedding. The result of the Wasserstein distance is used to strengthen AA.

where $S = \{P_{tr}(\mathcal{G}) \cdot P_{ts}(\mathcal{G}) = 0\}$, which means the features (environmental features) don't overlap in both sets. GCS is used to measure the difference in distribution between the training set and the test set. The covariance shift is a pervasive issue in real-world applications [26].

4.3. Problem Statement

The objective of graph classification under covariate distribution shift is to identify and disentangle stable features from environmental features.

Subsequently, the GNN encoder is trained in a manner analogous to traditional graph classification tasks. The problem can be defined as [26]:

$$f^* = \arg \min_f \sup_{e \in \mathcal{E}_{te}} \mathbb{E}_e[\ell(f(\mathcal{G}), y)]. \quad (4)$$

$\mathbb{E}_e[\ell(f(\mathcal{G}), y)]$ denotes the empirical risk of the environment e , \mathcal{E}_{te} is the test environment, $\ell(\cdot, \cdot)$ is the loss function, and f is the GNN encoder. The goal of our paper is to train a model f that can minimize the difference of environment between the training and the test set to alleviate the covariate shift. In this paper, we adopt contrastive learning [11] to further disentangle stable features from environmental features, therefore enhancing the capabilities of the original model.

5. Methodology

In this section, we first review the original adversarial invariant augmentation (AIA) model [26]. Then, we present

our contrast approach, which involves strengthening the stable feature generator and the adversarial augmenter. The overall architecture of our method is illustrated in Figure 5.

Initially, the input data is encoded by a shared GNN, then processed by the adversarial augmenter and the stable feature generator, respectively. Subsequently, we extract the stable feature and environment feature, which are then combined to generate the data augmentation version of the input data. Finally, we leverage the embeddings of the stable feature and environment features to optimize the model parameters. For the adversarial augmenter, we employ the triplet loss [24] as a new metric to measure the discrepancy between stable features and environment features. This enables us to push the environment features away from the stable features. For the stable feature generator, we employ InfoNCE [4], a contrastive learning loss, to effectively harness the vector information present in the latent space.

5.1. Original Model Introduction

In this section, we introduce the original model — AIA [26], elaborating on its architecture and the underlying principle designed to maintain feature consistency.

5.1.1. Model Architecture

Adversarial invariant augmentation (AIA) is a model that utilizes data augmentation to tackle the covariate distribution shift problem. It consists of two primary components: a stable feature generator and an adversarial augmenter: (1) The stable feature generator is designed to produce stable

features of \mathcal{G} that satisfy the stable feature consistency; (2) The adversarial augmenter aims to generate the environment features of \mathcal{G} , which adheres to environmental feature discrepancy. Then, AIA combines stable features with disturbed environment features to augment the graph \mathcal{G}' , which is named the data augmentation (DA) graph in the following section.

The stable feature generator and adversarial augmenter share the same architecture and are parameterized by θ_1 and θ_2 , respectively. Given an input graph \mathcal{G} with n nodes, the mask generation network first computes node representations using a GNN encoder $\tilde{h}(\cdot)$. To assess the importance of nodes and edges, it utilizes two MLP layers, $\text{MLP}_1(\cdot)$ and $\text{MLP}_2(\cdot)$, to generate a soft node mask matrix $M_x \in \mathbb{R}^{n \times 1}$ and an edge mask matrix $M_a \in \mathbb{R}^{n \times n}$, respectively. After masking the features, the remaining features are represented as stable or environmental features. The mask generation process can be summarized as follows:

$$Z = \tilde{h}(g), \quad M_{x_i} = \sigma(\text{MLP}_1(h_i)), \quad M_{a_{ij}} = \sigma(\text{MLP}_2([z_i, z_j])).$$

5.1.2. Proposed Principle

To maintain consistency between the stable feature and the original graph, the discrepancy between the environment features and the original graph, AIA proposed two principles: (1) the environmental feature discrepancy principle and (2) the stable feature consistency principle.

Principle 1 (Environmental feature discrepancy). Let $DAug\{\cdot\}$ denote data augmentation, and P denote the data distribution. For environment features, given a graph \mathcal{G} , the augmented graph $\mathcal{G}' = DAug\{\mathcal{G}\}$ should satisfy $GCS = \{P(\mathcal{G}), P(\mathcal{G}')\} \rightarrow 1$.

$GCS = \{P(\mathcal{G}), P(\mathcal{G}')\} \rightarrow 1$ means the environment features should keep away from the original distribution. From the viewpoint of data distribution, environment features should remain inconsistent with the original graph. In this paper, we also strengthen the discrepancy principle by keeping the environment features away from the stable feature, the original graph, and the DA graph.

Principle 2 (Stable feature consistency). Let $DAug\{\cdot\}$ denote data augmentation, \mathbb{E} denote the empirical risk, and $\mathcal{G}_{std}\{\mathcal{A}_{std}, \mathcal{X}_{std}\}$ denote a stable feature set. Then, the augmented graph $\mathcal{G}'_{std}\{\mathcal{A}'_{std}, \mathcal{X}'_{std}\} = DAug\{\mathcal{G}\}$ should meet $\mathbb{E}[\|A_{std} - A'_{std}\|_F^2] \rightarrow 0$ and $\mathbb{E}[\|X_{std} - X'_{std}\|_F^2] \rightarrow 0$.

The above formula implies that the generated stable features should preserve the consistency of the original graph. In this paper, we further strengthen the consistency between the stable features and the DA graph, thereby enhancing the model's generalization ability.

Finally, AIA utilizes cross-entropy to train the whole model [26]. AIA uses a very imaginative method to tackle the covariate distribution shift issue in graphs. However, the original approach overlooked the intrinsic correlation of the vector itself in the latent space, relying solely on label information. In contrast, we strengthen the AIA with contrastive learning to unleash the power of the vector itself.

5.2. MPAIACL

In this part, we introduce the core method. The overall architecture of our method is illustrated in Figure 5.

5.2.1. Strengthen the Stable Feature Generator

Our work is motivated by the success of contrastive learning with data augmentation in self-supervised learning, which has also been shown to be effective in graph-based applications [13]. However, there are various contrastive losses. For example, triplet loss enforces a margin between the anchor and positive samples, and the anchor and negative samples, such that the distance between the anchor and positive is always smaller. Information noise contrastive estimation (InfoNCE) [4] can learn representations of data such that positive examples are closer to each other in the feature space than negative examples (e.g., different images). Normalized temperature-scaled cross-entropy loss (NT-Xent) can build on the InfoNCE loss but introduces specific normalization and temperature scaling to improve training stability and performance. In this work, we adopt the widely used InfoNCE loss as our contrastive loss.

The original model utilizes the cross-entropy with label information to train. Here, we utilize InfoNCE [4] to unleash the power of the vector itself. InfoNCE is defined as follows:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\text{sim}(h_s^i, h_s^i)/\tau)}{\exp(\text{sim}(h_s^i, h_s^i)/\tau) + \sum_{j=1}^N \exp(\text{sim}(h_s^i, h_e^j)/\tau)} \quad (5)$$

Among the formulas, h denotes the hidden representation of \mathcal{G} in the latent space. h_s^i is the i -th graph \mathcal{G} only with stable features. h_e^i is the i -th graph that only has environment features. N denotes the total number of samples. τ is the temperature coefficient. We use the InfoNCE to bring the augmented graph closer to the stable features while pushing it further away from the environment feature. The intuition is that the similar predictions of neural networks indicate the close proximity in the manifold [30]. The InfoNCE loss encourages the representation of h_s^i to be closer to h_s^i , while pushing h_e^i away from h_s^i . This contrastive learning process enhances the model's ability to distinguish the stable feature and the environmental feature. However, relying solely on InfoNCE would result in the vectors moving towards or away from each other without any constraints, leading to uncontrolled and chaotic behavior. To prevent the vectors from moving in an uncontrolled way, we also incorporate label information serving as a ground truth boundary to guide the optimization process. The formula is defined as follows:

$$\mathcal{L}_{\text{Reg}}^{\text{std}} = - \left(\sum_i \mathcal{Y}_i \log(\text{Pred}_i^{\text{std}}) + \sum_i \mathcal{Y}_i \log(\text{Pred}_i^{\text{DA}}) \right). \quad (6)$$

Here, \mathcal{Y}_i is the label of each graph. Pred denotes the prediction of each graph. Finally, we have a new optimization

Table 2

Statistics of graph classification on the molecular datasets in the covariant shift.

Dataset		Molbace		Molbbbp		MolHiv		Molsider		Moltox21		Moltoxcast		Molclintox	
Covariant shift		size	scaffold	size	scaffold	size	scaffold	size	scaffold	size	scaffold	size	scaffold	size	scaffold
Train	Graph #	1211	1210	1633	1631	26169	24682	1143	1141	6265	6264	6862	6860	1183	1181
	Ave node #	36.66	33.60	27.02	22.49	27.87	26.25	39.47	29.97	21.31	16.54	21.56	16.68	29.90	25.52
	Ave edge #	79.05	72.59	58.71	48.43	60.20	56.68	83.47	62.81	44.72	33.74	44.66	33.54	64.17	54.10
Valid	Graph #	151	151	203	204	2773	4113	142	143	783	783	857	858	147	148
	Ave node #	23.69	37.23	12.06	33.20	15.55	24.95	10.52	43.24	7.60	26.76	7.71	26.17	11.50	32.75
	Ave edge #	52.17	81.29	24.27	71.81	32.77	54.53	20.09	91.84	14.06	53.13	14.10	56.09	22.78	71.36
Test	Graph #	151	152	203	204	3961	4108	142	143	783	784	857	858	147	148
	Ave node #	52.17	75.10	12.26	27.51	12.09	19.76	9.81	53.27	7.59	26.59	7.57	28.18	10.67	24.61
	Ave edge #	52.47	34.82	24.87	59.75	24.87	40.58	18.60	112.65	14.05	57.77	13.72	60.70	21.14	53.44
Class #		2	2	2	2	2	2	2	2	2	2	2	2	2	2

Table 3

Statistics of graph classification on the GOOD datasets in covariate shift.

Dataset		MolHiv		Motif		CMNIST
Covariate shift		size	scaffold	size	scaffold	color
Train	Graph #	26169	24682	18000	18000	42000
	Ave node #	27.86	26.25	16.92	17.06	75.0
	Ave edge #	60.20	56.68	43.56	48.89	1393.15
Valid	Graph #	2773	4113	3000	3000	7000
	Ave node #	15.54	24.94	39.22	15.82	75.0
	Ave edge #	32.77	54.53	107.03	33.00	1391.20
Test	Graph #	3961	4108	3000	3000	7000
	Ave node #	12.09	19.76	87.18	14.96	75.0
	Ave edge #	24.86	40.57	239.64	31.54	1394.33
Class #		2	2	3	3	10

Table 4

Statistics of graph classification on the GOOD datasets in correlation shift.

Dataset		MolHiv		Motif	
Correlation shift		size	scaffold	size	scaffold
Train	Graph #	14454	15209	12600	12600
	Ave node #	31.17	24.64	51.77	16.90
	Ave edge #	67.46	53.27	141.83	48.47
Valid	Graph #	9956	9365	6000	6000
	Ave node #	20.06	26.35	51.47	17.03
	Ave edge #	42.89	56.58	140.20	48.91
Test	Graph #	10525	10037	6000	6000
	Ave node #	19.39	26.64	51.60	17.01
	Ave edge #	41.42	57.21	141.51	48.69
Class #		2	2	3	3

target:

$$\min\{\mathcal{L}_{std} = \mathbb{E}[\mathcal{L}_{Reg}^{std} + \lambda\mathcal{L}_{InfoNCE}]\}. \quad (7)$$

The formula implies that the optimal stable feature generator is obtained by minimizing the joint empirical risk of the supervised learning and the contrastive loss term. Note that λ is a hyperparameter. Additionally, the formulation (7) not only enhances the capability of the model to distinguish the stable feature, but also prevents the representation vector from moving away without any constraint.

5.2.2. Strengthen the Adversarial Augmenter

To strengthen the adversarial augmenter, we considered utilizing contrastive learning to make environmental features different from the stable features. However, in this case, we lack a clear boundary to constrain the movement of the environmental features. Unlike the stable features, we do not have access to label information that can guide what constitutes an environmental feature. Therefore, we cannot utilize contrastive loss to guide the environment features toward their ground truth representations. We follow the method, which is Wasserstein distance [1] used in the original model as the distance metric to evaluate the distance between perturbation and the stable features.

In this paper, we further increase the discrepancy between stable features and environment features by the vector information in the latent space using triplet loss [24]. We employ it instead of the MSE, as the latter only minimizes point-wise Euclidean distances and therefore fails to capture the relative geometric relationships among samples in the latent space. In contrast, the triplet loss introduces a margin-based constraint between positive and negative pairs, simultaneously reducing intra-class distances while enlarging inter-class distances. We evaluate the Wasserstein distance between the original graph and the environmental features. Therefore, we have a new distance metric formula below:

$$\mathcal{L}_{triplet} = \frac{1}{N} \sum_{i=1}^N \max(0, d(\mathbf{h}_o^i, \mathbf{h}_{o'}^i) - d(\mathbf{h}_o^i, \mathbf{h}_e^i) + \alpha), \quad (8)$$

Here, \mathbf{h}_o^i indicates the i -th representation of the original graph; $\mathbf{h}_{o'}^i = \text{Dropout}(\mathbf{h}_o^i)$ indicates the dropout $\mathbf{h}_{o'}^i$; \mathbf{h}_e^i indicates i -th representation of the environment feature; $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$ indicates the L2 norm; and α indicates the margin in triplet loss. Therefore, we have a brand-new optimization object based on AIA below:

$$\max\{\mathcal{L}_{adv} = \mathbb{E}[\mathcal{L}_s - \alpha\mathcal{L}_{triplet} - \gamma\mathcal{L}_{reg}^{env}]\} \quad (9)$$

This formula means we should maximize the empirical risk of the Wasserstein distance, label information, and the regularization term. α and γ are the hyperparameters. The details of the regularization term \mathcal{L}_{reg}^{env} can be referred to

AIA [26]. Formula (9) maximizes the distances between h_{env} and the original graph, while maintaining the original distribution of the dataset.

5.3. Theoretical Analysis

In this section, we present a theoretical analysis of our approach, elucidating the stable features that motivate our adoption of contrastive learning to fully exploit the potential of the latent space, and explaining why this strategy is effective in achieving the objectives.

We first define h_s , h_{DA} , and h_e , which denote the stable features, data augmentation features, and environment features, respectively. To enhance the stable feature generator, we employ contrastive learning, formulated as:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\text{sim}(h_s^i, h_s^i)/\tau)}{\exp(\text{sim}(h_s^i, h_s^i)/\tau) + \sum_{j=1}^N \exp(\text{sim}(h_s^i, h_e^j)/\tau)}$$

, which yields the enhanced stable features h'_s and the augmented features h'_{DA} , respectively. After applying contrastive learning, we obtain

$$\|h'_s - h_e\| \gg \|h_s - h_e\|.$$

These relationships indicate that, following contrastive learning, the strengthened stable features become more distinct from the environment features in the latent space. This separation enhances the model's ability to capture domain-invariant representations, thereby improving its overall discriminative performance.

To enhance the adversarial augmenter, we employ the Wasserstein distance defined as

$$\mathcal{L}_{\text{triplet}} = \frac{1}{N} \sum_{i=1}^N \max(0, d(\mathbf{h}_o^i, \mathbf{h}_{o'}^i) - d(\mathbf{h}_o^i, \mathbf{h}_e^i) + \alpha),$$

and formulate the optimization objective as

$$\max \{ L_{\text{adv}} = \mathbb{E}[L_s - \alpha L_{DS} - \gamma L_{\text{reg}}^{\text{env}}] \},$$

through which we obtain the strengthened environment features h'_e . Afterward, we obtain

$$\|h_s - h'_e\| \gg \|h_s - h_e\| .$$

These expressions indicate that, after strengthening the adversarial augmenter, the strengthened environment features become further separated from both the stable features and the original graph in the latent space. This separation enhances feature discriminability, thereby improving the model's overall ability to distinguish invariant representations.

6. Experimental Results

In this section, we present the experimental results of our proposed method. We introduce the datasets and baselines employed in our experiments and provide a detailed analysis of the experimental results, offering insights into the

performance and efficacy of our proposed approach. To gain a deeper understanding of the contributions of individual components, we conduct an ablation study. Additionally, we perform a visualization analysis to provide a comparative assessment of MPAIACL and AIA. We also conduct the hyperparameter experiment of MPAIACL. It is noticed that all baselines adhere to their original settings, and all experiments are conducted on a single NVIDIA GeForce RTX 4060 Ti 16 GB GPU.

6.1. Settings

Datasets. We evaluate our model on the Open Graph Benchmark (OGB) [9] (including the Molbbbp, Molbace, Molsider, Moltox21, Moltoxcast, and Molclinxtox) and GOOD [7] datasets (including MolHiv, Motif, and CMNIST). Notice that Molbbbp, Molbace, Molsider, Moltox21, Moltoxcast, and Molclinxtox are molecular datasets collected from MoleculeNet [36]. We utilize the above datasets to conduct covariant shift experiments. We also utilize the GOOD datasets to conduct correlation shift experiments. For the molecular datasets, we adopt the covariate shifts introduced in [26], which leveraged scaffold and graph size to create diverse types of covariate shifts. Note that the scaffold-based construction employs scaffold splitting to partition the data into training, validation, and test sets. In contrast, the size-based shift involves training on large graphs, while using smaller graphs for validation and testing. For Molhiv, Motif, and CMNIST, we utilize the covariate shift in GOOD [7]. The details of the covariant datasets are shown in Table 2 and Table 3. For the correlation shift experiment, we utilize Molhiv, Motif, and CMNIST as our datasets. We utilize the correlation shift in GOOD [7]. The details of the correlation datasets are shown in Table 4.

Baselines. We compare the performance of MPAIACL against a range of representative baselines. For (I) generalization, we compare our method to (1) ERM and IRM [2], with a focus on identifying which properties of the training data correspond to spurious correlations and which properties capture the phenomenon of interest. (2) VREx [15], which proposes a penalty on the variance of training risks as a simpler alternative. (3) GroupDRO [23], combining group DRO models with enhanced regularization. We compare the above general methods because they overlook the importance of tackling stable features and environments from a contrastive perspective. For (II) graph generalization algorithms, we compare our approach with (1) CIGA [5], which aims to capture graph invariance for guaranteed out-of-distribution (OOD) generalization under diverse distribution shifts; (2) DIR-GNN [35] develops a framework for discovering invariant rationales, enabling the construction of inherently interpretable graph neural networks; (3) GSAT [20] addresses the limitations of post-hoc interpretation methods that often fail to provide stable and reliable explanations and instead extract features that are spuriously correlated with the task by attention mechanisms. (4) GALA [6] is a novel model that learns invariant graph representations under the guidance of an environment assistant model. We

compare the above graph generalization methods because they overlook the covariate shift distribution problem in OOD datasets. For (III) graph augmentation algorithms, we compare our approach with (1) FLAG [14], which iteratively augments node features with gradient-based adversarial perturbations during training; (2) GREA [16] separates rationales from environments and learns representations of real and augmented examples in latent spaces, enabling effective graph learning. We compare the above data augmentation methods because they often fail to preserve the stable features, as they lack explicit constraints and may inadvertently destroy the very information. (3) DropEdge [22] randomly removes edges from the graph. (4) GraphCL [39] employs a node-dropping and feature-masking strategy during the DA stage. Notice that we only use the mask feature strategy in GraphCL. We also compare with (5) AIA [26], which employs a data augmentation strategy to mitigate covariate shifts on graphs, since our MPAIACL is a refined model of it.

Evaluation metric. We adopt several evaluation metrics on different datasets. For the molecular datasets, including Molbbbp, Molbace, Molsider, Moltox21, Moltoxcast, and Molclintxtox, we employ the ROC-AUC as the evaluation metric. For the GOOD dataset, which comprises MolHiv, Motif, and CMNIST, we use ROC-AUC to evaluate MolHiv, and accuracy to evaluate both CMNIST and Motif.

6.2. Main Results

Covariate shift distribution. In the covariate shift distribution experiment, we employ the data manipulation approach introduced in [26], which induces covariate shifts in the molecular datasets by design. In this experiment, we compared our method MPAIACL with different baselines in different datasets. In Table 5, we make comparisons with various baselines in the covariate shift distribution using the OGB benchmark. Most baselines in generalization and graph generalization fail in covariate shift, such as G-DRO, GALA, etc. For the graph augmentation method, DropEdge, GraphCL fails in covariate shift compared with ERM. FLAG, AIA, and MPAIACL are obtaining an improvement to an extent. We consider Molbbbp in the size domain as an example, compared with ERM. For generalization, VREx and G-DRO obtain 0.47% and 0.98%, improvement, respectively. For graph generalization, CIGA, DIR-GNN, GALA perform ↓ 12.31%, ↓ 1.89%, and ↓ 9.35%, respectively. GSAT performs ↓ 2.66%. For graph augmentation, GREA, DropEdge, and GraphCL perform ↓ 0.95%, ↓ 2.45%, and ↓ 13.14%, respectively. FLAG, AIA, and MPAIACL, respectively, obtain 0.97%, 2.74%, and 4.47% improvement. In terms of overall results, MPAIACL demonstrates impressive performance on molecular datasets. Overall, our results demonstrate that MPAIACL is an effective variant of AIA, offering improved performance in the presence of covariate shifts.

In Table 6, we make comparisons with various baselines in covariant shift distribution using the GOOD benchmark. Many baselines fail in GOOD datasets in covariate shifts. We

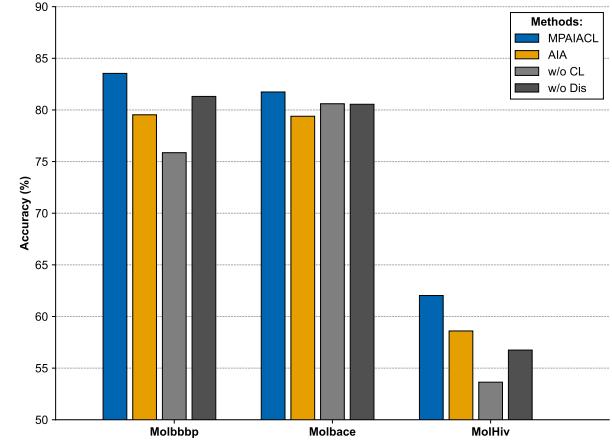


Figure 6: Ablation study on three different datasets. This demonstrates that the performance of MPAIACL outperforms others.

consider motifs in the size domain as an example, compared with ERM. For generalization, IRM performs ↓ 0.33%. VREx, and GroupDRO obtain 0.93%, and 0.21% improvement, respectively. For graph generalization, CIGA performs ↓ 2.60% DIRGNN, GSAT, and GALA obtain 0.53%, and 1.46%, 3.04% improvement, respectively. For graph augmentation, FLAG, DropEdge, and GraphCL perform ↓ 0.08%, ↓ 16.87% and ↓ 18.53%. GREA, AIA, and MPAIACL have 2.39%, 4.11%, and 11.97% improvement, respectively. Overall, MPAIACL delivers strong performance in both Motif and MolHiv, thereby validating its effectiveness.

However, in the CMNIST dataset, GALA achieves the best performance, while MPAIACL underperforms compared to GALA. In summary, MPAIACL achieves the best comprehensive performance in the GOOD dataset compared with other baselines.

Correlation shift distribution. While our primary focus is on addressing the covariate shift distribution problem, we also assess the performance of MPAIACL under correlation shift. Following the experimental setup in GOOD [7], we create correlation shift scenarios using the MolHiv (size, scaffold) and Motif (size, basis) datasets. For a comprehensive evaluation, MPAIACL was compared against 11 baseline methods: (1) IRM, (2) ERM [2], (3) VREx [15], (4) G-DRO [23], (5) CIGA [5], (6) DIR-GNN [35], (7) GSAT [20], (8) FLAG [14], (9) AIA [26], (10) DropEdge [22], and (11) GraphCL [39]. The results are presented in Table 7. We can observe that most methods are effective in handling Correlation shift distributions, with the notable exceptions of DropEdge and GraphCL, which employ random graph augmentation techniques. Comparing MPAIACL with the other baselines, we can observe that MPAIACL has superior performance across different domains in MolHiv, Motif, and CMNIST. Overall, the result also demonstrates that MPAIACL is effective in the correlation issue, which demonstrates that MPAIACL is an improved version of AIA.

Table 5

Experimental results of covariate shift distribution of molecular datasets. “-” indicates that GALA’s sampling strategy did not achieve multilabel binary classification. The best-performing result is highlighted in **bold**, while the second-best result is indicated with underlining.

Type	Method	Molbbbp		Molbace		Molsider		Moltox21		Moltocast		Molclintox	
		size	scaffold										
General generalization	IRM	77.56±2.48	67.22±1.15	77.06±1.65	69.15±2.59	54.20±1.26	55.10±1.32	71.26±0.76	68.72±1.71	57.65±3.08	57.37±0.73	62.59±8.91	58.68±3.33
	ERM	78.29±3.76	68.10±1.68	82.22±0.79	76.62±1.20	56.82±1.95	55.50±1.52	70.43±1.53	74.07±0.36	62.98±1.37	63.34±0.51	87.59±5.38	85.94±1.77
	VREx	78.76±2.37	68.74±1.03	79.67±0.23	66.70±0.14	56.82±1.95	58.79±0.23	63.22±0.25	68.94±0.76	56.79±1.44	57.83±0.92	80.59±9.56	79.10±1.13
	G-DRO	79.27±2.43	66.47±2.39	79.64±0.14	67.10±0.11	54.78±0.80	57.91±0.25	61.82±1.49	69.34±0.30	59.36±1.79	58.80±0.07	87.00±4.11	79.06±0.73
Graph generalization	CIGA	65.98±3.31	64.92±2.09	68.46±2.17	74.39±2.19	52.13±1.71	50.44±1.17	64.86±2.19	58.27±2.32	52.04±2.15	55.29±2.19	75.42±1.43	58.95±6.77
	DIR-GNN	76.40±4.43	66.86±2.25	77.48±2.28	77.98±2.81	54.02±0.61	52.47±1.57	68.52±2.13	67.91±1.21	62.75±0.65	56.18±0.86	36.88±1.24	71.51±5.18
	GSAT	75.63±3.83	66.78±1.45	78.09±2.19	73.84±3.05	56.67±2.10	61.27±0.42	68.75±4.44	73.20±0.51	62.64±1.01	61.80±0.43	87.73±3.37	89.52±1.77
	GALA	68.94±8.41	57.80±2.35	76.70±6.44	70.93±2.88	-	-	-	-	-	-	-	-
Graph augmentation	FLAG	79.26±2.26	67.69±2.36	<u>83.56±0.80</u>	76.93±0.93	52.09±1.44	58.65±2.03	65.72±1.12	<u>75.88±0.90</u>	<u>65.98±2.52</u>	62.81±0.18	84.99±5.56	86.83±1.35
	GREA	77.34±3.52	67.65±1.89	83.18±0.46	<u>78.78±1.55</u>	<u>57.40±0.64</u>	<u>59.53±1.05</u>	75.68±0.29	76.11±0.73	65.28±0.62	65.83±0.60	<u>90.40±4.02</u>	88.37±2.66
	AIA	<u>81.03±5.15</u>	<u>68.03±1.52</u>	80.76±1.19	77.74±2.36	54.41±2.53	56.32±0.32	70.29±1.04	75.37±0.68	65.13±0.45	<u>63.72±0.70</u>	82.49±15.83	87.87±1.81
	D-Edge	75.84±0.31	60.03±0.70	74.39±0.20	70.12±0.20	55.12±0.25	48.09±0.09	60.98±1.52	68.72±0.29	61.07±1.20	57.19±0.90	82.98±2.70	78.13±5.36
	GraphCL	65.15±0.87	61.13±2.26	69.29±3.43	62.95±1.37	49.48±0.08	47.24±0.10	49.09±0.15	47.07±0.54	49.26±0.06	50.06±0.10	43.36±1.71	47.56±2.31
	MPAIACL	82.76±0.03	69.64±0.70	83.68±0.04	79.11±0.54	57.54±0.02	57.86±1.02	71.53±0.62	74.36±0.31	66.25±0.34	63.34±0.36	93.75±0.01	87.90±1.40

Table 6

Experiment results of covariant shift distribution of GOOD datasets. The best-performing result is highlighted in **bold**, while the second-best result is indicated with underlining.

Type	Method	Motif		MolHiv		CMNIST	
		size	basis	size	scaffold	color	
General generalization	IRM	51.41±3.78	61.52±7.11	59.00±2.92	67.97±1.84	27.83±2.13	
	ERM	<u>51.74±2.88</u>	<u>68.66±4.25</u>	<u>59.94±2.37</u>	69.58±2.51	28.60±1.87	
	VREx	52.67±5.54	40.49±5.66	58.53±2.88	66.62±2.55	28.48±2.87	
	G-DRO	51.95±5.86	68.24±8.92	58.98±2.16	69.17±0.85	29.07±3.14	
Graph generalization	CIGA	49.14±8.34	66.43±11.31	59.55±2.56	69.40±2.39	32.22±2.67	
	DIR-GNN	52.27±4.56	62.07±8.75	58.08±2.31	68.07±2.29	33.20±6.17	
	GSAT	53.20±8.35	62.80±11.41	58.06±1.98	68.66±1.35	28.17±1.26	
	GALA	54.78±6.05	55.03±10.00	58.76±1.86	67.37±4.58	53.30±2.32	
Graph augmentation	FLAG	51.66±4.14	61.12±5.39	59.54±1.27	68.45±2.30	32.30±2.69	
	GREA	<u>54.13±10.02</u>	<u>56.74±9.23</u>	52.77±0.18	67.79±2.56	29.02±3.26	
	AIA	55.85±7.98	<u>68.93±0.21</u>	56.30±1.42	71.15±1.81	36.37±4.44	
	D-Edge	34.87±1.08	33.90±0.10	55.76±3.02	56.32±0.97	15.07±1.99	
	GraphCL	33.21±2.51	33.11±7.85	48.87±2.27	50.71±3.71	10.39±1.10	
	MPAIACL	63.71±5.02	69.77±0.79	60.39±4.01	69.69±2.11	<u>42.35±0.14</u>	

6.3. Ablation Study

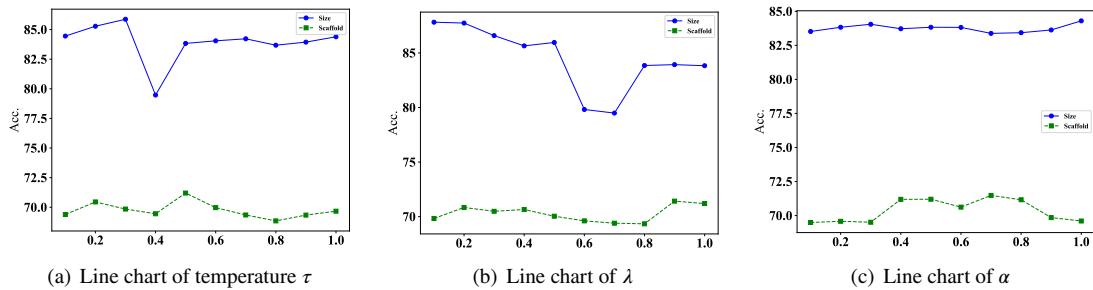
We conduct an ablation study to investigate the contributions of two key components in our approach: contrastive learning in the stable features generator and the use of Wasserstein distance in the adversarial augmentation. As illustrated in Fig. 6, we denote the following variants of MPAIACL: (1) “w/o cl”: MPAIACL without contrastive learning in the stable features generator; (2) “w/o dis”: MPAIACL without the Wasserstein distance-based regularization between stable features, original graph, and environment features in the adversarial augmenter. We observe that the

performance of each component degrades significantly when used independently, often resulting in worse performance than even AIA. Taking Molbbbp as an example, “w/o cl” is even worse than AIA, which performs ↓ 3.68%. Although “w/o dis” 1.78% improvement, it perform ↓ 2.23% than MPAIACL, respectively. In contrast, the combined version of MPAIACL achieves superior performance, obtaining 4.01% improvement over AIA, highlighting the importance of integrating these components for effective covariate shift adaptation. Without incorporating vector information in the latent space, the vectors would approach each other

Table 7

Experiment results of correlation shift distribution. The best-performing result is highlighted in **bold**, while the second-best result is indicated with underline.

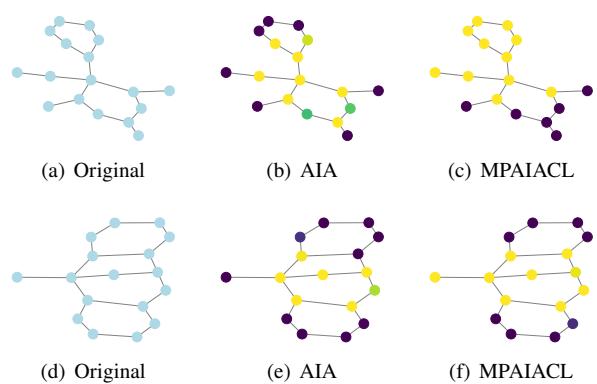
Type	Method	Motif		MolHiv	
		size	basis	size	scaffold
General generalization	IRM	64.68±0.75	74.79±1.01	53.97±1.77	69.07±0.35
	ERM	65.59±0.14	81.37±0.33	52.24±1.73	70.27±1.49
	VREx	51.90±3.85	56.18±2.32	71.17±8.39	65.68±1.76
	G-DRO	64.69±0.98	75.30±0.23	50.81±1.45	70.28±1.02
Graph generalization	CIGA	51.95±2.45	68.87±7.22	73.20±0.34	71.95±0.87
	DIR-GNN	47.12±7.42	75.07±6.66	69.72±2.28	67.40±2.23
	GSAT	37.33±1.01	52.20±0.20	49.08±5.07	67.56±0.82
Graph augmentation	FLAG	<u>66.78±0.70</u>	<u>79.04±0.43</u>	73.17±1.84	69.37±0.84
	AIA	65.42±2.54	76.26±5.02	70.37±0.43	69.81±1.18
	D-Edge	37.31±1.79	34.90±3.40	59.60±1.25	66.60±1.13
	GraphCL	36.05±1.50	37.68±1.57	57.03±2.57	58.88±5.41
	MPAIACL	69.44±1.71	81.64±1.65	70.40±1.33	<u>71.78±0.87</u>

**Figure 7:** Hyperparameter analysis.

without limitation, leading to suboptimal performance. This highlights the importance of utilizing vector information to regulate the latent space. Without utilizing the Wasserstein distance between stable features and environment features, the performance of our approach still falls short of the combined version. To achieve optimal fine-tuning performance, it is also essential to strengthen the adversarial augmenter.

6.4. Visualization

Here, we provide a visualization of MPAIACL and AIA to intuitively demonstrate the ability of MPAIACL to capture stable features. We utilize *NetworkX* to generate visualizations of our results. As shown in Fig. 8, the visualization is organized as follows: the left column displays the original graphs selected from the Molbbb training set, the middle column shows the stable features captured by AIA, and the right column presents the stable features captured by MPAIACL. In these figures, the color intensity represents the stable degree; the brighter the color, the greater the stability of the features. We can see that MPAIACL is generally a match for AIA, which indicates that MPAIACL is an effective method for capturing stable features. Furthermore,

**Figure 8:** Visualization.

MPAIACL appears to exhibit a more optimistic outlook on stable features and tends to capture more structure as stable features. While MPAIACL outperforms AIA in terms of accuracy, a crucial consideration is whether it also captures more consistent and stable features. As a variant of AIA,

Table 8

Complexity comparison between AIA and MPAIACL.

Dataset	Domain	AIA		MPAIACL (ours)	
		Time consumption	Memory consumption	Time consumption	Memory consumption
MolHiv	size	00h 41m 50s	395.50 MB	0 h 47 m 29 s	456.41 MB
	scaffold	00h 39m 12s	365.86 MB	0 h 44 m 20 s	430.39 MB
Motif	size	00h 14m 32s	223.65 MB	0 h 16 m 10 s	268.88 MB
	basis	00h 14m 08s	225.23 MB	0 h 15 m 29 s	268.29 MB
Molbbbp	size	00h 02m 46s	355.89 MB	0 h 2 m 48 s	409.58 MB
	scaffold	00h 02m 12s	290.09 MB	0 h 2 m 24 s	338.61 MB
Molbace	size	00h 02m 04s	414.96 MB	0 h 2 m 15 s	492.09 MB
	scaffold	00h 02m 01s	399.21 MB	0 h 2 m 11 s	462.71 MB
CMMNIST	color	02h 23m 08s	2201.90 MB	03h 17m 14s	2704.77 MB

MPAIACL leverages latent information to enhance performance, but its approach is more empirical, lacking a solid theoretical foundation. We will investigate this question further in our future work.

6.5. Hyperparameter Analysis

In this section, we conduct experiments to investigate hyperparameters' impact on our method's performance. Specifically, we analyze the influence of the temperature parameter τ in Equation 5, the regularization strength λ in Equation 7, and the hyperparameter α in Equation 9. We use Molbbbp as our dataset with different domains (size, scaffold) and explore the performance when the hyperparameter varies from 0 to 1 in increments of 0.1. It is noticed that the default hyperparameter settings are $\tau = 0.5$, $\lambda = 1$, and $\alpha = 0.5$.

As illustrated in Fig. 7, we have three different line charts for τ , λ , and α , respectively, in different domains. The performance varies significantly with different values of the hyperparameters.

The impact of varying τ on domain size is illustrated in Fig. 7(a). The accuracy increases steadily as τ ranges from 0 to 0.3. However, a sudden drop in accuracy occurs at $\tau = 0.4$. Beyond this point, the accuracy remains relatively stable, with minimal fluctuations between τ equals 0.5 and 1. In the domain scaffold, the accuracy exhibits slight fluctuations throughout the entire range of τ values.

As shown in Fig. 7(b), the impact of varying λ on domain size reveals a general downward trend in overall accuracy as λ increases from 0.1 to 1. Notably, the accuracy experiences a sharp decline at $\lambda = 0.6$ but recovers slightly at $\lambda = 0.8$. In the domain scaffold, the accuracy remains relatively stable, with only minor fluctuations observed throughout the entire range of λ values.

For the change of α in domain both size and scaffold, the accuracy slightly fluctuates throughout the entire range. Across both domain size and scaffold illustrated in Fig. 7(c), the accuracy remains relatively stable with respect to the

changes of α , exhibiting only minor fluctuations throughout the entire range.

6.6. Complexity Analysis

We further provide an analysis of the time complexity of our proposed method to assess its efficiency. We also present a comparison of the time and max GPU memory consumption of various datasets used in our experiments in Table 8, providing an overview of their computational requirements. As shown, the time and memory consumption of our proposed method are higher than those of AIA. This is because our method, MPAIACL, incurs additional computational costs due to the calculation of contrastive loss in the stable feature generator and the computation of Wasserstein distance in the adversarial augmenter.

To calculate the time complexity of our proposed method based on the architecture mentioned in Section 5.1.1, we first define n and e as the total number of nodes and edges, respectively. Let B denote the batch size. Let l , l_s , and l_a denote the layers of the GNN backbone, stable feature generator, and adversarial augmenter, respectively. Let h , h_s , and h_a denote the hidden layers of the GNN backbone, stable feature generator, and adversarial augmenter, respectively. Utilizing the definition above, the time complexity of the stable feature generator is $O(B(2leh + l_s eh_s))$. $2leh$ denotes the hidden embedding through two GNN layers with message-passing. $l_s eh_s$ denotes generating masks for nodes and edges of graphs. The time complexity regulation term is $O(2Bn)$, since we utilize the result of the stable generator to calculate the label information. The time complexity of the adversarial augmenter is $O(B(2leh + l_a eh_a))$, the same as the stable generator. Its corresponding regularization term is $O(B(n + e))$, since we calculate the Wasserstein distance of h_{ori} with h_{adv} . For convenience, we assume that $l_s = l_a$, $h_s = h_a$. The total time complexity of MPAIACL is then $O(2B(2leh + l_s eh_s + 1.5n + e))$.

6.7. Limitation

Although MPAIACL outperforms various baselines in our experiments, we acknowledge and reflect on the limitations of our approach. Although our approach is grounded in the manifold assumption [30] and intuition behind contrastive learning, we acknowledge that our work relies more heavily on empirical evidence and experimental experience. The strengthened strategy introduces a degree of randomness into the model’s performance, particularly in the adversarial augmenter, since we lack knowledge of the ground truth environment embedding and only make the environment embedding far away from the stable one and the augmentation one. Another limitation is that the augmented environment features may not accurately reflect real-world situations, particularly after contrastive learning, which can exacerbate this discrepancy. We will explore more theoretical ways to make the process more stable.

7. Conclusion

In this paper, we investigate graph classification under covariate distribution shift. We discover that the representation information in the latent space remains under-explored. Consequently, we harness the representative information in the latent space, thereby unlocking its potential to improve the performance of graph classification. We propose MPAIACL, a method that employs contrastive learning to fine-tune the existing model by fully releasing the power of the information in the latent space. Experiment results demonstrate the effectiveness of MPAIACL as it performs well compared to other baselines in different OOD datasets. Moreover, our in-depth study reveals that leveraging the latent space information is effective, providing valuable insight into the potential benefits of this approach. In future work, we plan to develop a theoretically grounded framework for augmenting environmental features in a principled manner, with the goal of improving model generalization in diverse and dynamic settings. Additionally, we intend to investigate self-supervised learning techniques to effectively mitigate the challenges posed by covariate distribution shifts, particularly in scenarios where labeled data is scarce or unavailable.

References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.
- [2] Martin Arjovsky, Léon Bottou, Ishaaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.
- [3] Jiangxia Cao, Xixun Lin, Shu Guo, Luchen Liu, Tingwen Liu, and Bin Wang. Bipartite graph embedding via mutual information maximization. In *The ACM International Conference on Web Search and Data Mining*, pages 635–643, 2021.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference On Machine Learning*, pages 1597–1607. PMLR, 2020.
- [5] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, MA Kaili, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems*, 35:22131–22148, 2022.
- [6] Yongqiang Chen, Yatao Bian, Kaiwen Zhou, Binghui Xie, Bo Han, and James Cheng. Does invariant graph learning via environment augmentation learn invariance? *Advances in Neural Information Processing Systems*, 36, 2024.
- [7] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. GOOD: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems*, 35:2059–2073, 2022.
- [8] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-Mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pages 8230–8248. PMLR, 2022.
- [9] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- [10] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9865–9874, 2019.
- [11] Yizhu Jiao, Yun Xiong, Jiawei Zhang, Yao Zhang, Tianqi Zhang, and Yangyang Zhu. Sub-graph contrast for scalable self-supervised graph representation learning. In *The International Conference on Data Mining*, pages 222–231. IEEE, 2020.
- [12] Wei Ju, Yiyang Gu, Xiao Luo, Yifan Wang, Haochen Yuan, Huasong Zhong, and Ming Zhang. Unsupervised graph-level representation learning with hierarchical contrasts. *Neural Networks*, 158:359–368, 2023.
- [13] Wei Ju, Yifan Wang, Yifang Qin, Zhengyang Mao, Zhiping Xiao, Junyu Luo, Junwei Yang, Yiyang Gu, Dongjie Wang, Qingqing Long, et al. Towards graph contrastive learning: A survey and beyond. *arXiv preprint arXiv:2405.11868*, 2024.
- [14] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Robust optimization as data augmentation for large-scale graphs. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 60–69, 2022.
- [15] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation. In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.
- [16] Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. In *the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1069–1078, 2022.
- [17] Mengyue Liu, Yun Lin, Jun Liu, Bohao Liu, Qinghua Zheng, and Jin Song Dong. B2-sampling: Fusing balanced and biased sampling for graph contrastive learning. In *the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1489–1500, 2023.
- [18] Yanbei Liu, Yu Zhao, Xiao Wang, Lei Geng, and Zhitao Xiao. Multi-scale subgraph contrastive learning. *arXiv preprint arXiv:2403.02719*, 2024.
- [19] Costas Mavromatis and George Karypis. Graph infoclust: Leveraging cluster-level node information for unsupervised graph representation learning. *arXiv preprint arXiv:2009.06946*, 2020.
- [20] Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, pages 15524–15543. PMLR, 2022.
- [21] Jiezhang Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1150–1160, 2020.
- [22] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node

- classification. In *8th International Conference on Learning Representations*. OpenReview.net, 2020.
- [23] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*.
- [24] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition*, pages 815–823, 2015.
- [25] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In *the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1696–1705, 2022.
- [26] Yongduo Sui, Qitian Wu, Jiancan Wu, Qing Cui, Longfei Li, Jun Zhou, Xiang Wang, and Xiangnan He. Unleashing the power of graph data augmentation on covariate distribution shift. *Advances in Neural Information Processing Systems*, 36, 2024.
- [27] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Yuanxing Ning, Philip S Yu, and Lifang He. SUGAR: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *the Web Conference*, pages 2081–2091, 2021.
- [28] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34:15920–15933, 2021.
- [29] Wenzuan Tu, Sihang Zhou, Xinwang Liu, Chunpeng Ge, Zhiping Cai, and Yue Liu. Hierarchically contrastive hard sample mining for graph self-supervised pretraining. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [30] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.
- [31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.
- [32] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *the International Conference on Learning Representations*. OpenReview.net, 2019.
- [33] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*, pages 23341–23362. PMLR, 2022.
- [34] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. In *International Conference on Learning Representations*, 2022.
- [35] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. In *International Conference on Learning Representations*, 2022.
- [36] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- [37] Chenxiao Yang, Qitian Wu, Jiahua Wang, and Junchi Yan. Graph neural networks are inherently good generalizers: Insights by bridging gnns and mlps. In *International Conference on Learning Representations*, 2023.
- [38] Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. Learning substructure invariance for out-of-distribution molecular representations. *Advances in Neural Information Processing Systems*, 35:12964–12978, 2022.
- [39] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.
- [40] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021.
- [41] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. Motif-driven contrastive learning of graph representations. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [42] Yifei Zhang, Yankai Chen, Zixing Song, and Irwin King. Contrastive cross-scale graph knowledge synergy. In *the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3422–3433, 2023.
- [43] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.
- [44] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *the Web Conference*, pages 2069–2080, 2021.