Learning Substructure Invariance for Out-of-Distribution Molecular Representations

Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, Junchi Yan[*]
Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University
{yangnianzu,zengkaipeng,echo740,jiaxiaosong,yanjunchi}@sjtu.edu.cn

Abstract

Molecule representation learning (MRL) has been extensively studied and current methods have shown promising power for various tasks, e.g., molecular property prediction and target identification. However, a common hypothesis of existing methods is that either the model development or experimental evaluation is mostly based on i.i.d. data across training and testing. Such a hypothesis can be violated in real-world applications where testing molecules could come from new environments, bringing about serious performance degradation or unexpected prediction. We propose a new representation learning framework entitled MoleOOD to enhance the robustness of MRL models against such distribution shifts, motivated by an observation that the (bio)chemical properties of molecules are usually invariantly associated with certain privileged molecular substructures across different environments (e.g., scaffolds, sizes, etc.). Specifically, We introduce an environment inference model to identify the latent factors that impact data generation from different distributions in a fully data-driven manner. We also propose a new learning objective to guide the molecule encoder to leverage environment-invariant substructures that more stably relate with the labels across environments. Extensive experiments on ten real-world datasets demonstrate that our model has a stronger generalization ability than existing methods under various out-of-distribution (OOD) settings, despite the absence of manual specifications of environments. Particularly, our method achieves up to 5.9% and 3.9% improvement over the strongest baselines on OGB and DrugOOD benchmarks in terms of ROC-AUC, respectively. Our source code is publicly available at https://github.com/yangnianzu0515/MoleOOD.

1 Introduction

Predicting molecular properties plays an important role in many related applications like drug discovery [13] and material design [49]. These professional tasks conventionally take great efforts by experts e.g. in chemistry and pharmacology. Recent years have witnessed inspiring breakthroughs on building effective machine learning models for scientific discovery, and solid progress has been made along the avenue of ML-based molecule representation learning (MRL). In general, MRL aims at embedding a molecule into a vector in latent space as a foundation model, on top of which the learned representations could be used for a variety of downstream tasks, such as target identification [66], retrosynthetic analysis [62], search of antibiotics [52], virtual screening [38] for drug discovery, etc.

The challenge, however, is that existing MRL methods are mostly based on an underlying hypothesis that training and testing molecules are independently sampled from an identical environment, yet

[*] Junchi Yan is the correspondence author who is also with Shanghai AI Laboratory.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

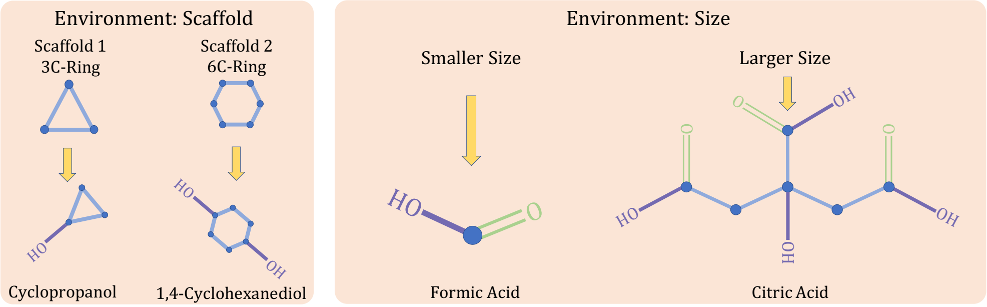


Figure 1: Two examples. **Left:** the shared substructure hydroxy $(-OH)$ invariantly contributes to the water solubility of the two molecules which contain different scaffolds, i.e. sampled from different environments by definition. **Right:** the water solubility of the two molecules with different sizes can be attributed to the shared substructure carboxy $(-COOH)$ invariantly, where different sizes are regarded as indicators to define different environments.

real-world environments are often dynamic and uncertain, which requires the model to effectively handle distribution shifts. In fact, the available experimental molecule data are rather limited while the candidate molecules to be tested are often diverse, coming from unknown environments. Taking the virtual screening [38] as an example (which is a common protocol in drug discovery and usually for target identification), the prediction model is typically trained on some known target proteins. However, some unpredictable events like COVID-19 may occur, bringing new targets from unknown distributions. Similar scenarios where training and testing data are sampled from different distributions are common in real world, posing an urgency for strengthening current MRL methods regarding out-of-distribution (OOD) generalization [39, 7, 42].

Existing methods devised for out-of-distribution generalization mostly focus on Euclidean data such as images, while few endeavors OOD generalization on non-Euclidean data [59]. In particular, molecules, as a kind of typical non-Euclidean data, i.e., graph-structured data, is different from visual data in nature. The work [26] point out that existing OOD models [3, 53, 17, 67, 48] fail to exhibit significant improvement on MRL tasks against distribution shifts and even the simple Empirical Risk Minimization (ERM) [54] method outperforms these latest methods, which is also empirically verified by [16]. We aim to develop an OOD method tailored for molecules to solve the OOD generalization problem on MRL in this paper.

We incorporate an effective prior in the molecule domain into our model design: the (bio)chemical properties of a molecule are usually associated with a few privileged molecular substructures, which has been consistently shown by studies [30, 46, 69, 28] across bio-informatics, pharmacy, and data mining. The common practice specifies environments as some prominent information of the molecules e.g. scaffold pattern [31, 22] and molecule size [26]. Fig. 1 provides two illustrative examples. Let's first take a look at the left example, where two molecules *Cyclopropanol* (C_3H_6O) and *1,4-Cyclohexanediol* ($C_6H_{12}O_2$) contain different scaffold patterns[2]: the former is *3C-ring* and the latter is *6C-ring*. Thus, the data-generating environments and the induced distributions which these two molecules are sampled from can be considered different [22]. Though sampled from different distributions, they are both readily soluble in water due to the invariant substructure hydroxy [23] shared across different environments. As for the example on the right of Fig. 1, the sizes of two molecules *Formic Acid* (CH_2O_2) and *Citric Acid* ($C_8H_8O_7$) differ a lot. Consequently, they can also be considered as being sampled from different environments. Owing to the shared invarint substructure carboxy $(-COOH)$, they are both readily soluable in water, too. Hence, a promising paradigm would be to learn the causal data-generating invariance from the substructures across environments, regarding a certain property, for the OOD generalization purpose.

Another important observation for consideration is that existing specifications for environments are often handcrafted or rule-based and not structured, which could provide insufficient information for capturing the fundamental relations across domains from the casual data-generating perspective.

[2] As a 2-D structural molecular framework [5], the scaffold reduces the chemical structure of a molecule to its core components, which can be obtained by removing side chains and only reserving the rings and parts connecting rings [64]. The scaffold can be an indicator to define a specific environment [31, 22].

Besides, some studies [26, 16] show that directly utilizing such environment labels as input when adapting existing OOD generalization methods to MRL tasks can be problematic. Furthermore, manual specifications of environments may be unavailable in reality. Hence, we aim to develop a label-free model that does not rely on the above ad-hoc environment labels. As shown later, our model can infer the environment labels in an unsupervised manner, namely for environment clustering.

To achieve robust molecule representation for OOD generalization and overcome potentially unreliable environment labels, we devise a new MRL framework without explicitly using the environment label information. We first formulate OOD generalization for molecular property prediction by introducing a latent variable for environments that affect the data generation. Then we analyze the essential cause behind the failure of existing MRL models and propose a new learning scheme based on the invariance principle [45, 42, 3, 59]. The training procedures contain two steps: 1) optimize an environment inference model from training data; 2) optimize a molecule encoder and a predictor. Our general framework can integrate existing GNN backbones and achieve improvements on four OGB molecular property prediction tasks [22], as shown in our experimental results. As for a newly released benchmark for drug-oriented OOD learning [26], even without access to environment labels, our method can still outperform state-of-the-art models that rely on environment labels for training in five out of six datasets. **The contributions of this paper are:**

- We formulate the out-of-distribution (OOD) generalization problem for molecule representation learning (MRL), by particularly incorporating an important observation that the substructure of molecule can convey invariant casual information across environments, regarding certain property prediction tasks. To our best knowledge, this is the first work that formulates the OOD problem in MRL background and proposes to leverage the invariance principle which opens a new perspective for handling substructure-aware distribution shifts.
- Under the environment-invariance principle with specific substructure invariance priors, we propose a new learning objective to learn robust representations. In particular, our model does not require environment labels which in fact can be noisy and unreliable, but instead achieve environment inference in an unsupervised manner. This design endows our model with practical applicability for molecular OOD learning where the manual specifications of the environments are often unavailable.
- We conduct extensive experiments on ten public datasets. Results demonstrate that our model yields consistent and significant improvements over various existing MRL methods as backbones and also achieves competitive or even superior prediction compared to state-of-the-art models tailored to OOD learning with environment labels used as extra inputs in both training and testing. Particularly, our method achieves up to 5.9% higher ROC-AUC on public OGB molecular property prediction benchmarks than the counterpart model trained with traditional objective. Besides, for drug-oriented benchmarks DrugOOD, when environment labels are not used, our model still outperforms several SOTA approaches tailored for general OOD learning (using environment labels as extra training information) by up to 3.9% w.r.t. ROC-AUC.

2 Backgrounds and Related Works

Out-of-Distribution Generalization. Deep neural networks are prone to suffering significant performance degradation under distribution shifts, motivating a surge of works on OOD generalization. Recent studies [47, 3, 9, 59] assume that there is a potential environment variable $\mathbf{e}$ accounting for the distribution shift between the training and testing data. In general cases the goal is to predict the target label $\mathbf{y}$ given the associated input $\mathbf{x}$. Then, the OOD problem could be formally formulated as:

$$\min_{f} \max_{e\in\mathcal{E}} \mathbb{E}_{(x,y)\sim p(\mathbf{x},\mathbf{y}|\mathbf{e}=e)}[l(f(x),y)|e], \tag{1}$$

where $\mathcal{E}$ denotes the support of environments, $f(\cdot)$ is the prediction model and $l(\cdot,\cdot)$ represents a loss function. Notice that $\mathbb{E}_{(x,y)\sim p(\mathbf{x},\mathbf{y}|\mathbf{e}=e)}[l(f(x),y)|e]$ is called the **risk function** under a given environment e and denoted as $\mathcal{R}_e(\mathbf{x}^e,\mathbf{y}^e)$ [32].

Invariant Learning. There is an emerging line of research [47, 3, 10, 12] regarding invariant predictor learning, for solving the OOD generalization problem. These methods propose to find an invariant predictor that could uncover invariant relationships between inputs and targets across all environments [32]. The invariant predictor aims to learn an invariant representation satisfying such a **invariance principle**: 1) **sufficiency**: shows sufficient predictive power for the target, 2) **invariance**:

contributes to equal (optimal) performance for the downstream tasks across all environments. Recent works adopt the invariance principle as a cornerstone for handling various distribution shifts on semi-structured data like graphs [59, 6] and sequences [63]. To our knowledge, our work is a pioneering attempt that leverages the invariance principle and incorporate useful domain knowledge for handling molecular graph classification tasks under distribution shifts.

Molecule Representation Learning. Existing molecule representation learning methods can be classified into two categories. The first is SMILES-based methods where SMILES refers to Simplified Molecular Input Line Entry System [2]. They use language models to process the textual representation (SMILES) of a molecule, for example, Transformer [55] or BERT [15]. SMILES is a linear encoding for molecules and highly depends on the traverse order of molecule graphs. Therefore its expressiveness is limited for problems like medication recommendation which we believe calls for fine-grained molecular structure extraction. Beyond the above linear encoding protocol, structure-based methods are also developed, which can be further classified into fingerprint-based and graph neural networks (GNN)-based methods. The molecular fingerprint techniques date back to the Morgan fingerprints [41]. However, those fingerprint-based methods are often handcrafted and not trained in an end-to-end fashion [25]. Since molecules can be viewed as structured graphs, graph neural networks have been widely used to learn molecule representation [27, 18, 23].

Existing general OOD methods [3, 53, 17, 67, 48] are not tailored to such non-Euclidean structured data, i.e. molecules. In several recent works on molecule property classification tasks [68, 57], the importance of molecular substructures has been emphasized and such inductive bias is incorporated into the design of those models. However, they are still based on the i.i.d. assumption and do not leverage those invariant substructure across different environments to achieve robust representations. In this paper, we propose a general framework orthogonal to these MRL studies to bridge OOD and MRL, which can adopt any existing MRL methods as the backbone to improve their robustness.

3 Methodology

3.1 Problem Formulation

We propose a OOD generalization framework tailored for molecule representation learning, entitled MoleOOD. All the random variables and the corresponding realizations are denoted as bold and thin letters, respectively. We first formulate the OOD generalization problem for MRL.

OOD Generalization Problem on Molecule Representation Learning. A molecular graph can be represented as $G = (V, E)$, where V is the graph's node set corresponding to atoms constituting the molecule and E denotes the graph's edge sets corresponding to chemical bonds. The training and testing molecule graph datasets are denoted as $\mathcal{G}^{train} = \{(G_i, y_i)\}_{i=1}^{N^{train}}$ and $\mathcal{G}^{test} = \{(G_i, y_i)\}_{i=1}^{N^{test}}$. Notice that the test dataset is drawn outside the distribution of the training dataset. The goal of molecule representation learning task is to predict the target label $\mathbf{y}$ given the associated input molecule $\mathbf{G}$. Based on Eq. 1, we can formulate the OOD problem on MRL tasks as:

$$\min_f \max_{e \in \mathcal{E}} \mathbb{E}_{(G_i, y_i) \sim p(\mathbf{G}, \mathbf{y} | \mathbf{e} = e)} [l(f(G_i), y_i) | e]. \tag{2}$$

The difficulty of this problem is that the training data only cover very limited environments in $\mathcal{E}$ while the model is expected to perform well on all the environments.

We elaborate our approach in the context of molecule property classification tasks in this paper. Existing MRL methods do not differentiate invariant and spurious substructures. Hence, the spurious correlations between irrelevant substructures and the target label will be encoded to learned molecular representations. When tested on unseen environments, the downstream classifier will be easily misled by these spurious correlations [3]. With the knowledge that (bio)chemical properties of a molecule are usually associated with a few privileged substructures [30, 46, 69, 28], we aim to suppress such spurious correlations and leverage environment-invariant substructures that more stably relate with the labels across environments to learn invariant molecular representations. Notice that the learned invariant molecular representations should satisfy the invariance principle mentioned in Sec. 2. We next introduce our method formally and then give the instantiation of our model.

3.2 Model Formulation

The framework contains two parts, the fronted molecule encoder Φ for learning an "invariant representation" of the input molecule graph and the back-end predictor ω for final prediction. Solving the formulation in Eq. 2 directly is intractable in practice since we cannot know all the environments, i.e, obtain a complete support set $\mathcal{E}$. We resort to minimizing the expectation of risks from different environments known in the training data,

$$\min_{\omega,\Phi} \mathbb{E}_{\mathbf{e}}[\mathcal{R}_{\mathbf{e}}(\mathbf{G}^{\mathbf{e}}, \mathbf{y}^{\mathbf{e}})], \text{ s.t. } \mathbf{y} \perp\!\!\!\perp \mathbf{e} \mid \Phi(\mathbf{G}), \tag{3}$$

where $f = \omega \circ \Phi$ and $\perp\!\!\!\perp$ denotes probabilistic independence. All learnable parameters of the molecule encoder Φ and the predictor ω are included in θ. Different from Eq. 2, we add an extra invariance constraint $\mathbf{y} \perp\!\!\!\perp \mathbf{e} \mid \Phi(\mathbf{G})$, which is used to suppress spurious correlations [10]. Since assessing causality is challenging, we could rethink the problem on the basis of information theory. Recall that we hope to let the molecule encoder leverage environment-invariant substructures and learn a molecular representation $\Phi(G)$ given a molecule G. Our goal is to maximize the predictive power of $\Phi(\mathbf{G})$ on $\mathbf{y}$, which can be measured by mutual information between $\Phi(\mathbf{G})$ and $\mathbf{y}$. Meanwhile, probabilistic independence between $\mathbf{y}$ and $\mathbf{e}$ given $\Phi(\mathbf{G})$ can be achieved via minimizing their mutual information. For convenience, we denote $\Phi(\mathbf{G})$ as $\mathbf{z}$ and Eq. 3 can be approximately solved by:

$$\max_{\omega,\Phi} \mathrm{I}(\mathbf{z}; \mathbf{y}), \text{ s.t. } \min_{\omega,\Phi} \mathrm{I}(\mathbf{y}; \mathbf{e}|\mathbf{z}). \tag{4}$$

Treating the outputs of ω and Φ as distribution $q_\theta(\mathbf{z}|\mathbf{G})$ and $q_\theta(\mathbf{y}|\mathbf{z})$ respectively, Eq. 4 can be specified as:

$$\max_{q_\theta(\mathbf{y}|\mathbf{z}), q_\theta(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{z}; \mathbf{y}), \text{ s.t. } \min_{q_\theta(\mathbf{y}|\mathbf{z}), q_\theta(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{y}; \mathbf{e}|\mathbf{z}). \tag{5}$$

Now, we have arrived at a clearer but still intractable optimization objective. Before specifying the practical instantiation of Eq. 5, let's discuss on the environment variable $\mathbf{e}$ first.

In practice, due to the non-trivial efforts to label the molecular environments, manual specifications of the environments may be unavailable in many cases. We may directly label molecules to different environments in terms of their scaffolds when the environment label is unavailable. But this is unreasonable in practice, because the final total environment number will be too large. Taking the dataset HIV for molecule property prediction tasks released by Open Graph Benchmark [22] as an example, OGB uses scaffold to split the molecules into different environments. Assuming that we regard each scaffold as an environment directly, $41,127$ molecules in HIV are partitioned into $19,076$ environments (see details in Appendix D). This environment count is much larger than other OOD datasets from other domains, e.g. Camelyon17[3] [4], CivilComments[4] [8], etc. Even though some datasets may provide manual specifications of environments, the environment counts are also too large, which is unfriendly to existing OOD models [26, 16]. Therefore, we propose to design an environment-inference model ψ to partition the molecule into different environments with a relatively smaller environment count. We denote the environment count as a hyper-parameter k.

Given prior $p(\mathbf{e}|\mathbf{G})$, we need to maximize the log likelihood of $p_\tau(\mathbf{y}|\mathbf{G})$ and then obtain the posterior $p_\tau(\mathbf{e}|\mathbf{G}, \mathbf{y})$, which are parameterized by τ. Since there is no analytical solutions to the true posterior, here we use variational inference (VI) to approximate it. Specifically, we introduce a variational distribution $q_\kappa(\mathbf{e}|\mathbf{G}, \mathbf{y})$ parameterized by κ to approximate $p_\tau(\mathbf{e}|\mathbf{G}, \mathbf{y})$.

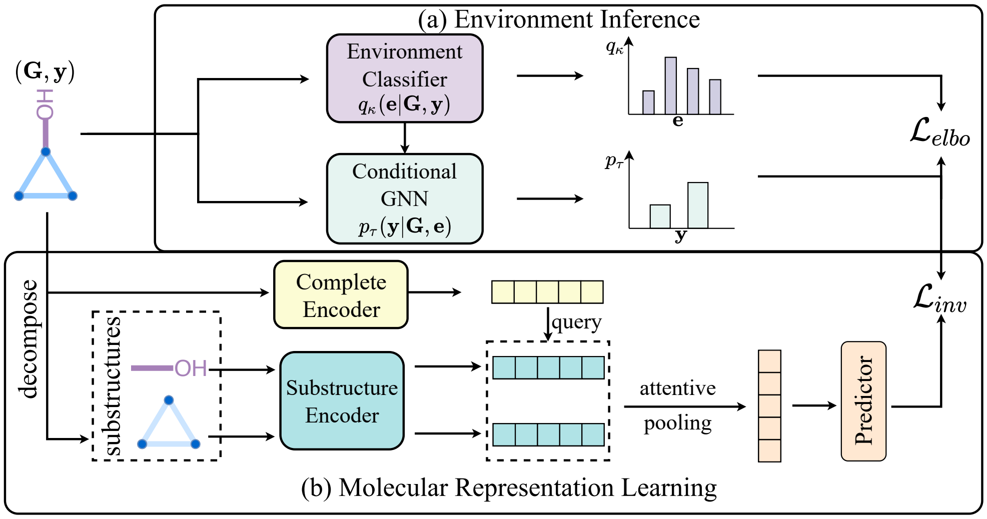
Proposition 1. *The Evidence Lower BOund (ELBO) of the observed molecule graph and corresponding label tuple (G, y): $\mathcal{L}(\tau, \kappa; (G, y)) = \mathbb{E}_{q_\kappa}[\log p_\tau(y|G, e)] - D_{KL}(q_\kappa(e|G, y) \parallel p_\tau(e|G))$.*

Our goal is to minimize the Kullback-Leibler (KL) divergence between $q_\kappa(\mathbf{e}|\mathbf{G}, \mathbf{y})$ and $p_\tau(\mathbf{e}|\mathbf{G}, \mathbf{y})$, i.e. $D_{KL}\,(q_\kappa(\mathbf{e}|\mathbf{G}, \mathbf{y}) \parallel p_\tau(\mathbf{e}|\mathbf{G}, \mathbf{y}))$, which is equivalent to maximizing the ELBO in Proposition 1. Then, the objective used to train this environment-inference model is transformed to:

$$\mathcal{L}_{elbo}(\tau, \kappa; \mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{(G,y)\in\mathcal{G}} \left[\mathbb{E}_{q_\kappa}[\log p_\tau(y|G, e)] - D_{KL}(q_\kappa(e|G, y) \parallel p(e|G))\right]. \tag{6}$$

[3] Camelyon17 is for tumor prediction, partitioning $455,954$ issue slides into 5 environments.

[4] CivilComments is for toxicity prediciton, partitioning $448,000$ online comments into 16 environments.



(a) Environment Inference
$(\mathbf{G}, \mathbf{y})$
OH
Environment Classifier $q_\kappa(\mathbf{e}|\mathbf{G}, \mathbf{y})$
Conditional GNN $p_\tau(\mathbf{y}|\mathbf{G}, \mathbf{e})$
q_κ
$\mathbf{e}$
p_τ
$\mathbf{y}$
$\mathcal{L}_{elbo}$
$\mathcal{L}_{inv}$
decompose
Complete Encoder
query
substructures
OH
Substructure Encoder
attentive pooling
Predictor
(b) Molecular Representation Learning

Figure 2: Overview of our model. The whole training procedure is divided into two stages: 1)Optimize the environment-inference model. Given an input molecule $(\mathbf{G}, \mathbf{y})$, we first infer the latent environment variable $\mathbf{e}$. This stage is trained under the guidance of $\mathcal{L}_{elbo}$. 2) Optimize the molecule encoder and the final predictor guided by $\mathcal{L}_{inv}$.

Let's look back to the objective given in Eq. 5 and give an equivalent tractable objective in practical instantiation, which involves the environment-inference model defined above:

$$\mathcal{L}_{inv}(\theta; \mathcal{G}, \tau) = \underbrace{\frac{1}{|\mathcal{G}|} \sum_{(G,y)\in\mathcal{G}} \left|\log q_\theta(y|G) - \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}[\log p_\tau(y|G, e)]\right|}_{\textcircled{1}} + \underbrace{\beta \mathbb{E}_{\mathbf{e}} \left[\frac{1}{|\mathcal{G}^e|} \sum_{(G,y)\in\mathcal{G}^e} [-\log q_\theta(y|G)]\right]}_{\textcircled{2}}, \qquad (7)$$

where $\mathcal{G}^e$ consists of the pairs of molecular graph G and corresponding label y under environment e.

Theorem 1. *With $q_\theta(\mathbf{y}|\mathbf{z})$ treated as a variational distribution, minimizing term* $\textcircled{1}$ *in Eq. 7 contributes to* $\min_{q_\theta(\mathbf{y}|\mathbf{z}), q_\theta(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{y}; \mathbf{e}|\mathbf{z})$*, letting* $\mathbf{z}$ *show equal performance for the downstream tasks across all environments, i.e.* $p(\mathbf{y}|\mathbf{z}, \mathbf{e}) = p(\mathbf{y}|\mathbf{z})$.

Theorem 2. *Regarding $q_\theta(\mathbf{y}|\mathbf{z})$ as a variational distribution, minimizing term* $\textcircled{2}$ *in Eq. 7 equals to* $\max_{q_\theta(\mathbf{y}|\mathbf{z}), q_\theta(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{z}; \mathbf{y})$*, letting* $\mathbf{z}$ *show sufficient predictive power for downstream tasks.*

Serving as theoretical justifications, Th. 1 and Th. 2 reveal that optimizing the objective in Eq. 7 forces the learned representation $\mathbf{z}$ to satisfy the invariance principle mentioned in Sec. 2, thus ensuring a valid solution for OOD problem defined in Eq. 2. Due to the limited space, the detailed proofs can be found in Appendix B.

3.3 Model Instantiations and Training

Environment-inference Module. For the approximate posterier model $q_\kappa(\mathbf{e}|\mathbf{G}, \mathbf{y})$, in principle we should design a module, entitled **Environment Classifier**, that takes $(\mathbf{G}, \mathbf{y})$ as the input and outputs the probabilistic distribution of $\mathbf{e}$. We use a Graph Isomorphism Network (GIN) [60], to learn a graph representation given $\mathbf{G}$. Then, the concatenation of this graph representation and label vector is fed to a feed-forward network to obtain a probabilistic distribution with regard to $\mathbf{e}$. We could set the prior $p_\tau(\mathbf{e}|\mathbf{G})$ to *Uniform distribution* or *Gaussian distribution*, encouraging the learned environment-partition to be uniform. As for $p_\tau(\mathbf{y}|\mathbf{G}, \mathbf{e})$, we also choose a GNN model (e.g. GIN) followed by a softmax activation function to model it. We call this module **Conditional GNN** because it conditions on $\mathbf{e}$. It takes $(\mathbf{G}, \mathbf{e})$ as the input and outputs the probabilistic distribution of $\mathbf{y}$.

The Molecule Encoder & The Final Predictor. Recall that we aim to learn an invariant substructure-aware molecular representation. Given a molecule $\mathbf{G}$, we can choose any molecule representation learning method to learn a representation $\mathbf{r}_\mathbf{G}$ for the complete molecular graph. This part is entitled **Complete Encoder**. Meanwhile, we decompose the input molecule into a set of chemical substructures using a molecule segmentation method, e.g. *breaking retrosynthetically interesting chemical substructures* (BRICS) [14], which is available as an API in RDKit [33]. For each substructure, we consider using a simple GNN to learn a corresponding representation. We call this GNN **Substructure Encoder**. Then, considering $\mathbf{r_G}$ as a query with regard to substructures, we operate attentive pooling on these substructure representations to obtain a new substructure-aware molecular representation. We then use this substructure-aware representation for downstream task. Guided by our proposed learning objective, we can encode some invariant relationships between certain substructures and target properties into this representation. The Complete Encoder, the Substructure encoder and the attentive pooling operation constitute our **Molecule Encoder** Φ. As for the **Predictor** ω, we implement it with a multi-layer perceptron, followed by a softmax function. The overview of our model is demonstrated in Fig. 2.

Training. We adopt a simple yet efficient two-stage training strategy to search for optimal parameters and the training procedure of our method is summarized in Algorithm 1:

1) optimizing the environment-inference model: $\kappa^*, \tau^* \leftarrow \arg\max_{\kappa,\tau} \mathcal{L}_{elbo}(\tau, \kappa; \mathcal{G}^{train})$.

2) optimizing the molecule encoder and the predictor: $\theta^* \leftarrow \arg\min_{\theta} \mathcal{L}_{inv}(\theta; \mathcal{G}^{train}, \tau)$.

Algorithm 1: The training procedure.

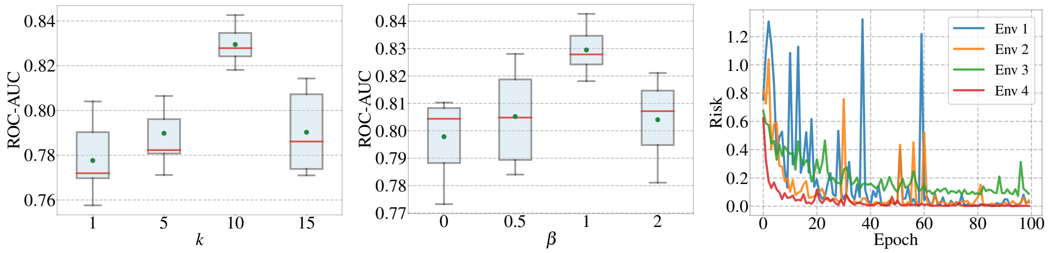
```
Input: Dataset $\mathcal{G}^{train} = \{(G_i, y_i)\}_{i=1}^{N^{train}}$; Number of training epochs for environment
       inference module $E_1$; Number of training epochs for the molecule encoder and the
       predictor $E_2$; Batch size $B$.
Output: Trained parameters $\theta$.
Initialize parameters $\theta, \tau$ and $\kappa$;
for $i \leftarrow 1$ to $E_1$ do
    Sample data batches $\mathcal{B} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k\}$ from $\mathcal{G}^{train}$ with batch size $B$;
    for $j \leftarrow 1$ to $k$ do
        Compute batch loss $\mathcal{L}_{elbo}(\tau, \kappa; \mathcal{G}_j)$ according to Eq. 6;
        Backpropagate $-\mathcal{L}_{elbo}$ and optimize parameters $\tau, \kappa$;
Freeze the parameters $\kappa, \tau$;
for $i \leftarrow 1$ to $E_2$ do
    Sample data batches $\mathcal{B} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k\}$ from $\mathcal{G}^{train}$ with batch size $B$;
    for $j \leftarrow 1$ to $k$ do
        Determine the environment of each sample $(G, y)$ in $\mathcal{G}_k$ by $\arg\max_e q_\kappa(e|G, y)$;
        Compute batch loss $\mathcal{L}_{inv}(\theta; \mathcal{G}_k, \tau)$ according to Eq. 7;
        Backpropagate $\mathcal{L}_{inv}$ and optimize parameters $\theta$;
Output the parameters $\theta$;
```

4 Experiments

Experiments are performed on 10 benchmark datasets and repeated 5 times with mean and standard deviation reported, running on a machine with i9-10920X CPU, RTX 3090 GPU and 128G RAM.

4.1 Datasets and Setups

Datasets and protocols. The four datasets **BACE**, **BBBP**, **SIDER** and **HIV**, are from by Open Graph Benchmark (OGB) [22]. We use the default train/val/test split with ratio 8:1:1. Each split contains a set of scaffolds (almost) different to each other. Hence we believe that to a certain degree, it provides an OOD test-bed as different scaffold often suggest different data-generation environments. The other six datasets are generated by the dataset curator provided by DrugOOD [26]. DrugOOD provides more diverse splitting indicators than OGB, including assay, scaffold and size. To comprehensively evaluate the performance of our method under different environment definitions, we adopt these three different splitting schemes on categories IC50 and EC50 provided in DrugOOD. Then we obtain six datasets, **EC50-**$*$ and **IC50-**$*$, where the suffix $*$ specifies the splitting scheme i.e. **IC50/EC50-assay/scaffold/size**. Notice that only the six datasets from DrugOOD provide manual specified environment labels. Refer to Appendix D for more details of datasets.



(a)

(b)

(c)

Figure 3: (a) Varying the specified environment number k. (b) Varying the trading-off parameter β in Eq. 7. (c) Risk curves of environments in the training process. All results are from 'GraphSAGE + ours.' on BACE dataset.

Table 1: Performance comparison with baselines on 4 out-of-distribution molecular property prediction datasets from Open Graph Benchmark (OGB) [22] in terms of ROC-AUC (%), namely, BACE, BBBP, SIDER and HIV. The best and the runner-up results are highlighted in **bolded** and underlined respectively. We emphasize the comparison against '$*$ **+ virtual node**', a variant of the original method augmented by an additional node connecting to all nodes in the raw graphs [18, 24, 37].

Methods	**BACE**	**BBBP**	**SIDER**	**HIV**
GCN [29]	80.01 ± 3.49	67.92 ± 1.07	58.90 ± 1.30	76.35 ± 2.01
GCN + virtual node	77.51 ± 3.07	68.19 ± 1.86	60.71 ± 1.34	75.76 ± 2.21
GCN + ours.	**84.33** ± **1.07**	**70.62** ± **0.99**	**63.38** ± **0.67**	**77.73** ± **0.76**
GIN [60]	77.83 ± 3.15	66.93 ± 2.31	59.05 ± 1.47	76.58 ± 1.02
GIN + virtual node	79.64 ± 2.02	66.77 ± 0.95	59.12 ± 0.95	77.11 ± 0.96
GIN + ours.	**81.09** ± **2.03**	**69.84** ± **1.84**	**61.63** ± **1.08**	**78.31** ± **0.24**
GraphSAGE [20]	77.41 ± 1.19	70.58 ± 0.58	58.00 ± 0.95	76.98 ± 1.13
GraphSAGE + virtual node	78.34 ± 2.08	69.29 ± 0.99	59.48 ± 1.37	77.28 ± 1.53
GraphSAGE + ours.	**82.95** ± **0.85**	**71.02** ± **0.75**	**61.09** ± **0.28**	**79.39** ± **0.51**

Metric. As the concerned property prediction tasks all relate to classification, we report the ROC-AUC score which is also in line with previous MRL works [65, 61, 56].

Baselines. Ideally, any MRL method can be adapted into our method as backbone to improve their generalization ability against distribution shifts. We adapt three backbones: **GCN** [29], **GIN** [60] and **GraphSAGE** [20] into our method. We compare the adapted version with the original method. We also compare against another augmented version "**+ virtual node**" [18, 24, 37]. Furthermore, we compare our method with six OOD generalization methods on MRL tasks: **ERM** [54], **IRM** [3], **DeepCoral** [53], **DANN** [17], **MixUp** [67] and **GroupDro** [48]. Due to the fact that most of these methods require the manual specification of environments in dataset, we report this comparison on datasets from DrugOOD only. Each of the method is configured using the same parameters reported in the original paper or selected by grid search. For the sake of fairness, the embedding size of all methods are set to be equal in comparison. We specify the training details in the Appendix C.

4.2 Performance Comparison

Improvements to existing MRL methods. As demonstrated in Table 1, baselines obtain consistent improvements after adapted to our methods across all the four datasets released by OGB in terms of ROC-AUC. Our method also beats the augmented version, "+ virtual node" , of baselines on all datasets, i.e. adding a virtual node. The results indicate that, orthogonal to prior studies on MRL, our method is a general framework which can incorporate existing MRL methods and improve their generalization ability for OOD data. We attribute the superior performance of our method in molecular properties predictions under OOD setting to that, our proposed learning objective enforces the model to learn environment-invariant representations against distribution shifts.

Superiority to other OOD generalization methods. Table 2 summarizes the results in comparsion with six state-of-the-art methods tailored for OOD learning, where we obtain the following observa-

Table 2: Evaluation with other OOD generalization methods on 6 out-of-distribution datasets from DrugOOD [26] in terms of ROC-AUC (%). The best and the runner-up in each columns are highlighted in **bolded** and underlined respectively. Note the baselines except ERM and MixUp all require environment labels. All methods including ours use GIN [60] as backbones.

Dataset	IC50			EC50		
Environment	**Assay**	**Scaffold**	**Size**	**Assay**	**Scaffold**	**Size**
ERM [54]	70.93 ± 2.10	67.31 ± 1.72	67.40 ± 0.56	69.35 ± 7.38	63.92 ± 2.09	60.94 ± 1.95
IRM [3]	70.85 ± 2.41	66.06 ± 1.23	58.46 ± 2.11	69.94 ± 1.03	63.74 ± 2.15	58.30 ± 1.51
DeepCoral [53]	69.82 ± 4.23	66.36 ± 2.57	59.21 ± 2.09	69.42 ± 3.35	63.66 ± 1.87	56.13 ± 1.77
DANN [17]	70.00 ± 1.03	63.61 ± 2.32	65.77 ± 0.47	66.97 ± 7.19	64.33 ± 1.82	61.11 ± 0.64
MixUp [67]	70.22 ± 3.66	66.43 ± 1.08	**67.77 ± 0.23**	70.62 ± 2.12	64.53 ± 1.66	62.67 ± 1.41
GroupDro [48]	69.98 ± 1.74	64.09 ± 2.05	58.46 ± 2.69	70.52 ± 3.38	64.13 ± 1.81	59.06 ± 1.50
Ours.	**71.38 ± 0.68**	**68.02 ± 0.55**	66.51 ± 0.55	**73.25 ± 1.24**	**66.69 ± 0.34**	**65.09 ± 0.90**

Table 3: Ablation study on EC50-$*$ by ROC-AUC (%). We show the results of MixUp that performs best among baselines on all EC50-$*$ datasets and the naive ERM, which minimizes the average empirical loss on training data, for comparison. Notice that ERM and MixUp don't require manual specified environments labels. We also present the results of DANN, which requires manual specifications of environment and obtains competitive results with MixUp. All methods use GIN [60] as backbone.

Method	**Assay**	**Scaffold**	**Size**
ERM (GIN + ERM loss)	69.35 ± 7.38	63.92 ± 2.09	60.94 ± 1.95
MixUp	70.62 ± 2.12	64.53 ± 1.66	62.67 ± 1.41
DANN	66.97 ± 7.19	64.33 ± 1.82	61.11 ± 0.64
Our architecture + ERM loss	71.44 ± 2.02	65.99 ± 0.42	64.23 ± 0.71
GIN + new learning objective	72.07 ± 1.14	66.33 ± 1.38	64.43 ± 1.10
DANN using our inferred environment label	68.83 ± 2.44	64.95 ± 1.07	62.56 ± 1.54
Our model using given environment label	71.94 ± 2.77	66.29 ± 0.85	63.38 ± 1.20
Our full model	**73.25 ± 1.24**	**66.69 ± 0.34**	**65.09 ± 0.90**

tions. Except on IC50-size, our method outperforms all baselines across all datasets due to its ability to enforce the molecule encoder to leverage environment-invariant substructures that more stably relate with the labels across environments. Our method ranks the third on IC50-size after MixUp and ERM. Different from the other methods, MixUp constructs more training exmaples and uses more data to train the model. That's why MixUp obtains best performance among all methods on IC50-size in our analysis. As for ERM, [26, 16] have pointed out that simple ERM shows better performance compared to subsequent OOD methods when datasets have relatively large environment counts. Even though we have set the environment number k to a smaller value than the ground-truth number given by the dataset, we still need to prevent k from being too small (see discussion in Sec. 4.4), leading to our poorer performance than ERM on IC50-size.

4.3 Ablation Study of Components

We analyze the contributions of different model components to the final performance in this section. Table 3 reports detailed ablation experimental results on EC50-assay, EC50-scaffold and EC50-size.

Attention-based architecture. We study the impact of the attention-based architecture introduced in Sec. 3.3 by assembling this architecture with ERM loss. We beat ERM and MixUp only with this architecture on three datasets. The results show that learning a representation for each substructure and then attentively aggregating these learned representations to obtain a final substructure-aware representation performs better than learning a representation for a complete molecular graph directly. This verifies our assumption that the substructure perspective is of importance to boosting performance of existing MRL methods. With the aid of such a substructure-grained learning architecture, the impact of our learning objective can be further strengthened.

New learning objective. To evaluate the impact of our proposed new learning objective, we equip GIN with this new objective. We can see compared to using the substructure-grained learning architecture only, only using the proposed new learning objective can bring more significant improvement. Thus, we can attribute the main superiority of our full model to this new objective. Combined with the architecture discussed above, the new objective is able to better guide the molecule encoder to learn environment-invariant molecular representations against distribution shifts.

Environment inference. Now we turn to investigate the performance with respect to our proposed environment-inference module. One motivation for this module is that in reality manual specifications of environments may be unavailable due to the high price for labeling environments by experts. But when environment labels are available, how will be performance be like if directly utilizing the given environment partition? An ablation study is targeted on this. Taking the EC50-assay dataset as an example, it has given the environment partition and it specifies 47 environments in total. We utilize the given environment partition directly and keep the remaining parts in line with our full model. The results show that utilizing the given environment label, our method still can beat ERM and MixUp. But compared to our full model where we set the environment number k to 20, it obtains inferior performance. Additionally, to further examine the effectiveness of our proposed environment inference method, we relabel the environment for each molecule for DANN according to our inferred environment partition. We can see that based on the new environment partition, DANN obtains better performance than using the initial given environment labels across three datasets. The reason why inferring environment instead can outperform directly using the given environment label is mainly due to the existing given partitions are often handcraftedrule-based and not structured. In contrast, letting the model learn a environment partition by itself may be more effective to some degree.

4.4 Hyper-parameter Sensitivity Study

We investigate the sensitivity of our method to these two hyper-parameters: the specified number of environments k, the trading-off parameter β in Eq. 7. Fig. 3(a) shows the performance regarding different environment number k. It shows that the performance of our methods degrades when k is too small (e.g. $k = 1, 5$) or too large (e.g. $k = 15$). When $k = 1$ i.e. we regard all training data as from only one environment, the performance is the poorest. This justifies that partitioning the training samples into different environments is necessary. Fig. 3(b) shows the results of our method by varying the trade-off parameter β. Our method obtains the worst performance when $\beta = 0$. This is mainly because Eq. 7 is reduced to the first term when $\beta = 0$. According to Theorem 2, without the second term of Eq. 7, the sufficiency condition of invariance principle cannot be satisfied, resulting in the performance degradation.

4.5 Risk Dynamics

Additionally, to shed insights of the ability of our method to lower the risks of different environments, we visualize the risk dynamic curve of some environments in Fig. 3(c). As is shown in Fig. 3(c), the difficulties of decreasing the risk on different environments are different. Though the risks of some environments vibrate violently at the beginning of training process (e.g. Env 1 and Env 2), with time elapsing, risks on all environments can decrease stably.

5 Conclusion

We have proposed a general framework which can incorporate any existing MRL method as backbone to improve their generalization ability against distribution shifts. Specifically, we devise a new learning scheme with its equivalent practical instantiation. We also develop an environment inference model to identify each molecule's corresponding environment without need of manual specifications of environments. Extensive experimental results on ten datasets demonstrate that our model yields consistent and significant improvements over various existing MRL methods as backbones. Additionally, our model achieves competitive or even superior performance compared to state-of-the-art models designed for OOD learning that require manual specified environment labels as extra inputs.

Acknowledgement

This work was partly supported by National Key Research and Development Program of China (2020AAA0107600), National Natural Science Foundation of China (61972250, 72061127003), and Shanghai Municipal Science and Technology (Major) Project (22511105100, 2021SHZDZX0102).

References

[1] L. Q. Al-Mawsawi, R. Dayam, L. Taheri, M. Witvrouw, Z. Debyser, and N. Neamati. Discovery of novel non-cytotoxic salicylhydrazide containing hiv-1 integrase inhibitors. *Bioorganic & medicinal chemistry letters*, 17(23):6472–6475, 2007.

[2] E. Anderson, G. D. Veith, and D. Weininger. *SMILES, a line notation and computerized interpreter for chemical structures*. US Environmental Protection Agency, Environmental Research Laboratory, 1987.

[3] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[4] P. Bandi, O. Geessink, Q. Manson, M. Van Dijk, M. Balkenhol, M. Hermsen, B. E. Bejnordi, B. Lee, K. Paeng, A. Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018.

[5] G. W. Bemis and M. A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[6] B. Bevilacqua, Y. Zhou, and B. Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, pages 837–851, 2021.

[7] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2178–2186, 2011.

[8] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[9] P. Bühlmann. Invariance, causality and robustness. *Statistical Science*, 35(3):404–426, 2020.

[10] S. Chang, Y. Zhang, M. Yu, and T. Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.

[11] T. Cofala and O. Kramer. An evolutionary fragment-based approach to molecular fingerprint reconstruction. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1156–1163, 2022.

[12] E. Creager, J.-H. Jacobsen, and R. Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[13] L. David, A. Thakkar, R. Mercado, and O. Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):1–22, 2020.

[14] J. Degen, C. Wegscheid-Gerlach, A. Zaliani, and M. Rarey. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem*, 2008.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] M. Ding, K. Kong, J. Chen, J. Kirchenbauer, M. Goldblum, D. Wipf, F. Huang, and T. Goldstein. A closer look at distribution shifts and out-of-distribution generalization on graphs, 2022.

[17] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[19] M. Glymour, J. Pearl, and N. P. Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[20] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[21] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

[22] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[23] S. Ishida, T. Miyazaki, Y. Sugaya, and S. Omachi. Graph neural networks with multiple feature extraction paths for chemical property estimation. *Molecules*, 26(11):3125, 2021.

[24] K. Ishiguro, S.-i. Maeda, and M. Koyama. Graph warp module: an auxiliary module for boosting the power of graph neural networks. *arXiv preprint arXiv:1902.01020*, 2019.

[25] S. Jaeger, S. Fulle, and S. Turk. Mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling*, 2018.

[26] Y. Ji, L. Zhang, J. Wu, B. Wu, L.-K. Huang, T. Xu, Y. Rong, L. Li, J. Ren, D. Xue, et al. Drugood: Out-of-distribution (ood) dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. *arXiv preprint arXiv:2201.09637*, 2022.

[27] W. Jin, C. Coley, R. Barzilay, and T. Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. *Advances in neural information processing systems*, 30, 2017.

[28] Y.-T. Kao, S.-F. Wang, M.-H. Wu, S.-H. Her, Y.-H. Yang, C.-H. Lee, H.-F. Lee, A.-R. Lee, L.-C. Chang, and L.-H. Pao. A substructure-based screening approach to uncover n-nitrosamines in drug substances. *Journal of Food & Drug Analysis*, 30(1), 2022.

[29] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[30] J. Klekota and F. P. Roth. Chemical substructures that enrich for biological activity. *Bioinformatics*, 24(21):2518–2525, 2008.

[31] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[32] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[33] G. Landrum et al. Rdkit: Open-source cheminformatics. 2006.

[34] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[35] A. Leman and B. Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.

[36] X. Q. Lewell, D. B. Judd, S. P. Watson, and M. M. Hann. Recap retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *Journal of chemical information and computer sciences*, 38(3):511–522, 1998.

[37] J. Li, D. Cai, and X. He. Learning graph-level representation for drug discovery. *arXiv preprint arXiv:1709.03741*, 2017.

[38] J. Lim, S. Ryu, K. Park, Y. J. Choe, J. Ham, and W. Y. Kim. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. *Journal of chemical information and modeling*, 59(9):3981–3988, 2019.

[39] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory (COLT)*, 2009.

[40] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, et al. Chembl: towards direct deposition of bioassay data. *Nucleic acids research*, 47(D1):D930–D940, 2019.

[41] H. L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 1965.

[42] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, pages 10–18, 2013.

[43] N. Neamati, H. Hong, J. M. Owen, S. Sunder, H. E. Winslow, J. L. Christensen, H. Zhao, T. R. Burke, G. W. Milne, and Y. Pommier. Salicylhydrazine-containing inhibitors of hiv-1 integrase: implication for a selective chelation in the integrase active site. *Journal of medicinal chemistry*, 41(17):3202–3209, 1998.

[44] J. Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19:2, 2000.

[45] J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

[46] C. Phanus-umporn, W. Shoombuatong, V. Prachayasittikul, N. Anuwongcharoen, and C. Nantasenamat. Privileged substructures for anti-sickling activity via cheminformatic analysis. *RSC advances*, 2018.

[47] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[48] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[49] B. Sanchez-Lengeling, C. Outeiral, G. L. Guimaraes, and A. Aspuru-Guzik. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (organic). 2017.

[50] J. Shen, F. Cheng, Y. Xu, W. Li, and Y. Tang. Estimation of adme properties with substructure pattern recognition. *Journal of chemical information and modeling*, 50(6):1034–1041, 2010.

[51] M. Singh, R. Divakaran, L. S. K. Konda, and R. Kristam. A classification model for blood brain barrier penetration. *Journal of Molecular Graphics and Modelling*, 96:107516, 2020.

[52] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[53] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[54] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeuIPS*, 2017.

[56] H. Wang, W. Li, X. Jin, K. Cho, H. Ji, J. Han, and M. Burke. Chemical-reaction-aware molecule representation learning. In *International Conference on Learning Representations*, 2022.

[57] S. Wang, Z. Li, S. Zhang, M. Jiang, X. Wang, and Z. Wei. Molecular property prediction based on a multichannel substructure graph. *IEEE Access*, 8:18601–18614, 2020.

[58] Y. Wang, R. Magar, C. Liang, and A. Barati Farimani. Improving molecular contrastive learning via faulty negative mitigation and decomposed fragment contrast. *Journal of Chemical Information and Modeling*, 2022.

[59] Q. Wu, H. Zhang, J. Yan, and D. Wipf. Handling distribution shifts on graphs: An invariance perspective. In *International Conference on Learning Representations*, 2022.

[60] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[61] M. Xu, H. Wang, B. Ni, H. Guo, and J. Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pages 11548–11558. PMLR, 2021.

[62] C. Yan, Q. Ding, P. Zhao, S. Zheng, J. Yang, Y. Yu, and J. Huang. Retroxpert: Decompose retrosynthesis prediction like a chemist. *Advances in Neural Information Processing Systems*, 33:11248–11258, 2020.

[63] C. Yang, Q. Wu, Q. Wen, Z. Zhou, L. Sun, and J. Yan. Towards out-of-distribution sequential event prediction: A causal treatment. In *Advances in neural information processing systems*, 2022.

[64] A. B. Yongye, J. Waddell, and J. L. Medina-Franco. Molecular scaffold analysis of natural products databases in the public domain. *Chemical biology & drug design*, 80(5):717–724, 2012.

[65] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

[66] X. Zeng, S. Zhu, W. Lu, Z. Liu, J. Huang, Y. Zhou, J. Fang, Y. Huang, H. Guo, L. Li, et al. Target identification among known drugs by deep learning from heterogeneous networks. *Chemical Science*, 11(7):1775–1797, 2020.

[67] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[68] X. Zhao, B. Zong, Z. Guan, K. Zhang, and W. Zhao. Substructure assembling network for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[69] J. Zhu, Y. Liu, C. Wen, and X. Wu. Dgdfs: Dependence guided discriminative feature selection for predicting adverse drug-drug interaction. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix I.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix J.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A and B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix C.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4 and Appendix D.
   (b) Did you mention the license of the assets? [Yes] See Section 4 and Appendix D.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Appendix D.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix D.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proof for Proposition 1

Proof. Our goal is to minimize the Kullback-Leibler (KL) divergence between $q_\kappa(\mathbf{e}|\mathbf{G}, \mathbf{y})$ and $p_\tau(\mathbf{e}|\mathbf{G}, \mathbf{y})$. For the observed molecule graph and corresponding label tuple (G, y),

$$\begin{aligned}
&D_{KL}\left(q_\kappa(e|G,y) \parallel p_\tau(e|G,y)\right) \\
&= \int_{q_\kappa} q_\kappa(e|G,y) \log \frac{q_\kappa(e|G,y)}{p_\tau(e|G,y)} \mathrm{d}e = \int_{q_\kappa} q_\kappa(e|G,y) \log \frac{q_\kappa(e|G,y) p_\tau(y|G) p_\tau(G)}{p_\tau(e,G,y)} \mathrm{d}e \\
&= \left(\int_{q_\kappa} q_\kappa(e|G,y) \log q_\kappa(e|G,y) \mathrm{d}e + \int_{q_\kappa} p_\kappa(e|G,y) \log p_\tau(G) \mathrm{d}e \right. \\
&\quad \left. - \int_{q_\kappa} \log p_\tau(e,G,y) \mathrm{d}e \right) + \int_{q_\kappa} q_\kappa(e|G,y) \log p_\tau(y|G) \mathrm{d}e \\
&= \int_{q_\kappa} q_\kappa(e|G,y) \log \frac{q_\kappa(e|G,y)}{p_\tau(y|G,e) p_\tau(e|G)} \mathrm{d}e + \log p_\tau(y|G) \\
&= \mathbb{E}_{q_\kappa}[\log q_\kappa(e|G,y) - \log p_\tau(y|G,z) - \log p_\tau(e|G)] + \log p_\tau(y|G) \\
&= -\mathbb{E}_{q_\kappa}[\log p_\tau(y|G,e)] + \mathbb{E}_{q_\kappa}[\log q_\kappa(e|G,y) - \log p_\tau(e|G)] + \log p_\tau(y|G) \\
&= -\underbrace{\left(\mathbb{E}_{q_\kappa}[\log p_\tau(y|G,e)] - D_{KL}(q_\kappa(e|G,y) \parallel p_\tau(e|G))\right)}_{\mathcal{L}(\tau,\kappa;G,y)} + \log p_\tau(y|G) \\
&= -\mathcal{L}(\tau,\kappa;G,y) + \log p_\tau(y|G)
\end{aligned} \tag{8}$$

Rearrange Eq. 8 and we can get,

$$\mathcal{L}(\tau,\kappa;G,y) = -D_{KL}\left(q_\kappa(e|G,y) \parallel p_\tau(e|G,y)\right) + \log p_\tau(y|G). \tag{9}$$

The defined $\mathcal{L}(\tau,\kappa;G,y)$ is called *Evidence Lower BOund* (ELBO) [21]. According to Eq. 9, maximizing this ELBO is equivalent to minimizing the KL divergence and maximizing $\log p_\tau(y|G)$. For the observed molecule graph and corresponding label tuple (G, y), we obtain the ELBO:

$$\mathcal{L}(\tau,\kappa;G,y) = \mathbb{E}_{q_\kappa}[\log p_\tau(y|G,e)] - D_{KL}(q_\kappa(e|G,y) \parallel p(e|G)). \tag{10}$$

We thus conclude the proof. □

B Proofs for Theorems

In this paper, we extend the invariance assumption [47, 3] to molecule representation learning:

Assumption 1. *Given a molecular graph G, there exists an encoder Φ yielding a graph-level representation $r_G \in \mathbb{R}^d$. Define $\mathbf{r}$ as a random variable of r_G and it satisfies: 1) (Invariance condition): $p(\mathbf{y}|\mathbf{r}, \mathbf{e}) = p(\mathbf{y}|\mathbf{r})$, and 2) (Sufficiency condition): $\mathbf{y} = h(\mathbf{r}) + \mathbf{n}$, where h is a non-linear function, $\mathbf{n}$ is a independent noise.*

With the terminology of information theory, we present a useful lemma [59] that interprets the invariance and sufficiency conditions in Assumption 1:

Lemma 1. *In terms of information theory, the two conditions in Assumption 1 can be equivalently expressed as 1) invariance: $\mathrm{I}(\mathbf{y}; \mathbf{e}|\mathbf{r}) = 0$ and 2) sufficiency: $\mathrm{I}(\mathbf{y}; \mathbf{r})$ is maximized.*

Proof. For the invariance, it can be obtained by the fact that

$$\mathrm{I}(\mathbf{y}; \mathbf{e}|\mathbf{r}) = \mathbb{E}_{p(\mathbf{e},\mathbf{r})}[D_{KL}(p(\mathbf{y}|\mathbf{e},\mathbf{r}) \| p(\mathbf{y}|\mathbf{r}))] \tag{11}$$

For the sufficiency, we first prove that every triplet $(\mathbf{G}, \mathbf{r}, \mathbf{y})$ satisfying that $\mathbf{y} = h(\mathbf{r}) + \mathbf{n}$ would also satisfy $\mathbf{r} = \arg\max_{\mathbf{r}} \mathrm{I}(\mathbf{y}; \mathbf{r})$. We prove it by contradiction. Assume that $\mathbf{r} \neq \arg\max_{\mathbf{r}} \mathrm{I}(\mathbf{y}; \mathbf{r})$ and there exists $\mathbf{r}'$ with $\mathbf{r}' = \arg\max_{\mathbf{r}} \mathrm{I}(\mathbf{y}; \mathbf{r})$ with $\mathbf{r} \neq \mathbf{r}'$. Then there exists another random variable $\tilde{\mathbf{r}}$ and a mapping function f_m such that $\mathbf{r}' = f_m(\mathbf{r}, \tilde{\mathbf{r}})$. Then we will have $\mathrm{I}(\mathbf{y}; \mathbf{r}') = \mathrm{I}(\mathbf{y}; \mathbf{r}, \tilde{\mathbf{r}}) = \mathrm{I}(h(\mathbf{r}); \mathbf{r}, \tilde{\mathbf{r}}) = \mathrm{I}(h(\mathbf{r}); \mathbf{r}) = \mathrm{I}(\mathbf{y}; \mathbf{r})$, which leads to contradiction.

Then we prove that every triplet $(\mathbf{G}, \mathbf{r}, \mathbf{y})$ satisfying that $\mathbf{r} = \arg\max_{\mathbf{r}} \mathrm{I}(\mathbf{y}; \mathbf{r})$ would also satisfy $\mathbf{y} = h(\mathbf{r}) + \mathbf{n}$ by contradiction. Suppose that $\mathbf{y} \neq h(\mathbf{r}) + \mathbf{n}$ and there exists $\mathbf{r}' \neq \mathbf{r}$ with $\mathbf{y} = h(\mathbf{r}') + \mathbf{n}$. Then the inequality $\mathrm{I}(h(\mathbf{r}'); \mathbf{r}) < \mathrm{I}(h(\mathbf{r}'); \mathbf{r}')$ holds. That means $\mathbf{r}' = \arg\max_r \mathrm{I}(\mathbf{y}; \mathbf{r})$, leading to contradiction. □

B.1 Proof for Theorem 1

Proof. According to the dependency relationship $\mathbf{z} \leftarrow \mathbf{G} \rightarrow \mathbf{y}$, we have

$$\begin{aligned}
&\mathrm{I}(\mathbf{y}; \mathbf{e}|\mathbf{z}) \\
&= D_{KL}(p(\mathbf{y}|\mathbf{z}, \mathbf{e}) \parallel p(\mathbf{y}|\mathbf{z})) \\
&= D_{KL}(p(\mathbf{y}|\mathbf{z}, \mathbf{e}) \parallel \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}|\mathbf{z}, \mathbf{e})\right] \\
&= D_{KL}\left(q(\mathbf{y}|\mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}|\mathbf{G}, \mathbf{e})\right]\right) - D_{KL}\left(q(\mathbf{y}|\mathbf{z}) \parallel p(\mathbf{y}|\mathbf{z}, \mathbf{e})\right) \\
&\quad - D_{KL}\left(\mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}|\mathbf{z}, \mathbf{e})\right] \parallel \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}|\mathbf{G}, \mathbf{e})\right]\right) \\
&\leq D_{KL}\left(q(\mathbf{y}|\mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}|\mathbf{G}, \mathbf{e})\right]\right).
\end{aligned} \tag{12}$$

Next, we have

$$\begin{aligned}
&D_{KL}\left(q(\mathbf{y}|\mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}|\mathbf{G}, \mathbf{e})\right]\right) \\
=&\mathbb{E}_{G\sim p(\mathbf{G})}\mathbb{E}_{y_G\sim p(\mathbf{y}|\mathbf{G}=G)}\mathbb{E}_{z_G\sim q(\mathbf{z}|\mathbf{G}=G)}\left[\log \frac{q(\mathbf{y}=y_G|\mathbf{z}=z_G)}{\mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}=y_G|\mathbf{G}=G, \mathbf{e}=e)\right]}\right] \\
=&\frac{1}{|\mathcal{G}|}\sum_{(G,y_G)\in\mathcal{G}} \mathbb{E}_{z_G\sim q(\mathbf{z}|\mathbf{G}=G)}\left[\log \frac{q(\mathbf{y}=y_G|\mathbf{z}=z_G)}{\mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}=y_G|\mathbf{G}=G, \mathbf{e}=e)\right]}\right].
\end{aligned} \tag{13}$$

Based on Jensen Inequality and Triangle Inequality, we can obtain that $D_{KL}\left(q(\mathbf{y}|\mathbf{z}) \parallel \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}\left[p(\mathbf{y}|\mathbf{G}, \mathbf{e})\right]\right)$ is upper bounded by:

$$\frac{1}{|\mathcal{G}|}\sum_{(G,y)\in\mathcal{G}} \left|\log q_\theta(y|G) - \mathbb{E}_{p(\mathbf{e}|\mathbf{G})}[\log p_\tau(y|G, e)]\right|. \tag{14}$$

Thus we can prove that minimizing term ① in Eq. 7 is equivalent to $\min_{q_\theta(\mathbf{y}|\mathbf{z}), q_\theta(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{y}; \mathbf{e}|\mathbf{z})$. □

B.2 Proof for Theorem 2

Proof. Given the dependency relationship $\mathbf{z} \leftarrow \mathbf{G} \rightarrow \mathbf{y}$, we hold $\max_{q(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{y}; \mathbf{z})$ is equivalent to $\min_{q(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{y}; \mathbf{G}|\mathbf{z})$. Also we have

$$\begin{aligned}
\mathrm{I}(\mathbf{y}; \mathbf{G}|\mathbf{z}) =& D_{KL}(p(\mathbf{y}|\mathbf{G}, \mathbf{e}) \| p(\mathbf{y}|\mathbf{z}, \mathbf{e})) \\
=& D_{KL}(p(\mathbf{y}|\mathbf{G}, \mathbf{e}) \| q(\mathbf{y}|\mathbf{z})) - D_{KL}(p(\mathbf{y}|\mathbf{z}, \mathbf{e}) \| q(\mathbf{y}|\mathbf{z})) \\
\leq& D_{KL}(p(\mathbf{y}|\mathbf{G}, \mathbf{e}) \| q(\mathbf{y}|\mathbf{z})),
\end{aligned} \tag{15}$$

Based on this, we will have

$$\mathrm{I}(\mathbf{y}; \mathbf{G}|\mathbf{z}) \leq \min_{q(\mathbf{y}|\mathbf{z})} D_{KL}(p(\mathbf{y}|\mathbf{G}, \mathbf{e}) \| q(\mathbf{y}|\mathbf{z})). \tag{16}$$

Then we can also derive the following inequality via Jensen Inequality:

$$\begin{aligned}
D_{KL}(p(\mathbf{y}|\mathbf{G}, \mathbf{e}) \| q(\mathbf{y}|\mathbf{z})) &= \mathbb{E}_{\mathbf{e}}\mathbb{E}_{G\sim p_e(\mathbf{G})}\left[\mathbb{E}_{y_G\sim p_e(\mathbf{y}|\mathbf{G}=G)}\mathbb{E}_{z\sim q(\mathbf{z}|\mathbf{G}=G)}\left[\log \frac{p_e(\mathbf{y}=y_G|\mathbf{G}=G)}{q(\mathbf{y}=y_G|\mathbf{z}=z_G)}\right]\right] \\
&\leq \mathbb{E}_{\mathbf{e}}\left[\frac{1}{|\mathcal{G}^e|}\sum_{(G,y_G)\in\mathcal{G}^e} \log \frac{p_e(\mathbf{y}=y_G|\mathbf{G}=G)}{\mathbb{E}_{z_G\sim q(\mathbf{z}|\mathbf{G}=G)}q(\mathbf{y}=y_G|\mathbf{z}=z_G)}\right] \\
&= C + \mathbb{E}_{\mathbf{e}}\left[-\frac{1}{|\mathcal{G}^e|}\sum_{(G,y_G)\in\mathcal{G}^e} \log q(\mathbf{y}=y_G|\mathbf{G}=G)\right],
\end{aligned} \tag{17}$$

where C is a constant. Then the problem $\min_{q(\mathbf{y}|\mathbf{z})} D_{KL}(p(\mathbf{y}|\mathbf{G}, \mathbf{e}) \| q(\mathbf{y}|\mathbf{z}))$ can be solve by

$$\min \mathbb{E}_{\mathbf{e}}\left[\frac{1}{|\mathcal{G}^e|}\sum_{(G,y_G)\in\mathcal{G}^e} \left[-\log q_\theta(\mathbf{y}=y_G|\mathbf{G}=G)\right]\right], \tag{18}$$

which means minimizing term ② in Eq. 7 contributes to $\max_{q_\theta(\mathbf{y}|\mathbf{z}), q_\theta(\mathbf{z}|\mathbf{G})} \mathrm{I}(\mathbf{z}; \mathbf{y})$. □

C Implementation Details

C.1 Baselines

This section describes training configurations for all baselines, which are compared in this paper.

Three backbones. We adapt three backbones into our method, namely, **GCN** [29], **GraphSAGE** [20] and **GIN** [60]. We also emphasize the comparison with their augmented versions, i.e. "**+ virtual node**" [18, 24, 37]. For GCN and GIN, we use the implementations provided by Open Graph Benchmark [22][5]. We implement GraphSAGE and its corresponding augmented version by ourselves. For these baselines, grid search of learning rate over $\{1e-2, 5e-3, 1e-3, 5e-4, 1e-4\}$ and dropout rate over $\{0.1, 0.3, 0.5\}$ is performed to select the best parameters. The embedding size of all methods including ours are all set to 256 for the sake of fairness.

- **GCN** [29] is a scalable approach on graph-structured data that is based on an efficient variant of convolutional neural networks.
- **GIN** [60] generalizes the Weisfeiler-Lehman (WL) graph isomorphism test [35] and hence achieves maximum discriminative power among GNNS.
- **GraphSAGE** [20] learns a function that generates embeddings by sampling and aggregating features from a node's local neighborhood.
- **GCN/GIN/GraphSAGE + virtual node** [18, 24, 37] is a variant of the original method augmented by an additional node connecting to all nodes in the raw graph.

Models tailored for OOD learning. We compare our method against six state-of-the-art methods: **ERM** [54], **IRM** [3], **DeepCoral** [53], **DANN** [17], **MixUp** [67] and **GroupDro** [48]. We use the implementations of these six method provided by DrugOOD[6]. We search for the optimal hyper-parameters by ranging learning rate over $\{1e-3, 5e-4, 1e-4, 5e-5, 1e-5\}$ and dropout rate over $\{0.1, 0.3, 0.5\}$. The embedding size of all models including ours are all set to 128 for fairness.

- **ERM** [54] minimizes the average empirical loss on training data.
- **IRM** [3] penalizes feature distributions for environments that have different optimum predictors. We set the penalty weight and the penalty anneal iteration to 10 and 500, respectively.
- **DeepCoral** [53] penalizes differences in the means and covariances of the feature distributions for each environment, which are exactly the distribution of last layer activations in a neural network. The penalty weight is set to 0.001.
- **DANN** [17] encourages feature representations to be consistent across domains. We set to the inverse factor to 0.2.
- **MixUp** [67] constructs additional virtual samples for training from two examples which are randomly sampled from the training data. We set the probability and interpolate strength to 0.1.
- **GroupDro** [48] minimizes the worst-case training loss over a set of pre-defined environments. The step size is set to 0.001.

C.2 Our Method

We implement our method in Pytorch. As for experiments on OGB datasets, we implement the Environment Classifier, the Conditional GNN and the Substructure Encoder which are mentioned in Sec. 3.3 all in Graph Isomorphism Network (GIN) [60]. We use grid search on validation set for hyper-parameter tuning by ranging learning rate from $\{1e-2, 5e-3, 1e-3, 5e-4, 1e-4, 5e-5, 1e-5\}$, dropout rate from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, the trading-off parameter β from $\{0.5, 1, 2, 4\}$, the environment count k from $\{5, 10, 15, 20, 40, 80\}$. As for the prior $p(\mathbf{e}|\mathbf{G})$, we set it to a *Uniform* distribution or a discrete *Gaussian* distribution. We use CrossEntropyLoss for all models and the Adam optimizer is used for gradient-based optimization.

[5] `https://github.com/snap-stanford/ogb`

[6] `https://github.com/tencent-ailab/DrugOOD`

D More Details of Datasets

In this paper, we use ten publicly available benchmark datasets in total. Four of them, namely, BACE, BBBP, SIDER and HIV are released by Open Graph Benchmark (OGB) [22]. The rest six are released by DrugOOD [26], i.e. IC50-assay, IC50-scaffold, IC50-size, EC50-assay, EC50-scaffold and EC50-size. We provide detailed descriptions for them as below.

- **BBBP** is a dataset of Brain-Blood Barrier Penetration. Each molecule has a label indicating whether it can penetrate through brain cell membrane to enter central nervous system.
- **BACE** is a dataset of binding affinity against human beta-secretas 1. Each molecule has a label indicating whether it binds to human beta-secretase 1.
- **SIDER** is a dataset of marked drugs and adverse drug reactions (ADRs). Molecules are grouped into 27 system organ classes.
- **HIV** is a dataset of HIV antiviral activity. Each molecule has an active or inactive label.
- **IC50/EC50-scaffold/assay/size** are datasets generated by the automated dataset curator provided by DrugOOD from the large-scale bioassay deposition website ChEMBL [40]. The suffix specifies the splitting scheme. These six datasets target on ligand-based affinity prediction (LBAP). Each molecule has an active or inactive label.

Notice that the phenomenon that there exist a few invariant substructures w.r.t. certain property indeed exists in the datasets we use in our paper. Taking **HIV** dataset as an example, *salicylhydrazide* substructure displays potent HIV-1 integrase (IN) inhibitory activity, which has been identified by previous studies [1, 43]. Additionally, for **BBBP** dataset, as pointed out by recent studies [50, 51], some substructures are closely related to brain-blood barrier penetration.

All these ten datasets do not contain personally identifiable information or offensive content. Table 4 shows the detailed statistics of datasets. For all datasets, we adopt the default training-validation-test split as shown in Table 4. We use all molecules in the training set to optimize the model parameters. Then, we select hyper-parameters using the validation set, and we report the results on test molecule set for the model that achieves the best results on the validation set.

Table 4: **Summary of datasets used in this paper.** #Train/#Valid/#Test denotes the number of samples in the training/validation/test set, respectively. #Total is the sum of #Train, #Valid and #Test. #Tasks is the output dimensionality required for prediction. Additionally, we also list which split scheme is adopted and whether the manual specification of environments is available for each dataset.

	Dataset	#Train	#Valid	#Test	#Total	#Tasks	Split Scheme	Specify Environments?
OGB	**BACE**	1,210	151	152	1,513	1	Scaffold	✗
	BBBP	1,631	204	204	2,039	1	Scaffold	✗
	SIDER	1,141	143	143	1,427	27	Scaffold	✗
	HIV	32,901	4,113	4,113	41,127	1	Scaffold	✗
DrugOOD	**EC50-assay**	4,540	2,572	2,490	9,602	1	Assay	✓
	EC50-scaffold	2,570	2,532	2,533	7,635	1	Scaffold	✓
	EC50-size	4,684	2,313	2,398	9,395	1	Size	✓
	IC50-assay	34,179	19,028	19,028	72,235	1	Assay	✓
	IC50-scaffold	21,519	19,041	19,048	59,608	1	Scaffold	✓
	IC50-size	36,597	17,660	16,415	70,672	1	Size	✓

Table 5: We count the number of scaffolds that contain 1, 2, 3, 4 and 5 samples, respectively.

Size	Number
1	$14,295$
2	$2,330$
3	862
4	449
5	255

Next, let's discuss on the details of HIV dataset, which is released by Open Graph Benchmark (OGB) [22]. OGB adopts scaffold splitting scheme to split the HIV into train/validation/test set. We count the number of scaffolds that only contain 1, 2, 3, 4 and 5 molecules, respectively, and summarize the statistics in Table 5. Notice that HIV has $19,076$ scaffolds in total. We can see there are $18,191$ scaffolds containing less or equal to 5 molecules, accounting for 95.45% of the total scaffold count. HIV has a great deal of environments that contains few samples, which poses great challenge to directly applying some existing OOD generalization methods to datasets like HIV [26]. Thus, for datasets released by OGB, partitioning the mococules into different environments according to their scaffolds may not be suitable in practice. Such a observation motivates us to propose the environment-inference model.

E Understanding the Data-generating Process

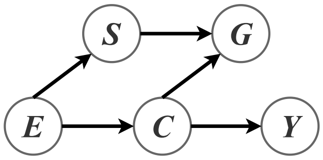


Figure 4: SCM

We provide a causal perspective to understand the data-generating process. Recalling the two molecules *Cyclopropanol* (C_3H_6O) and *1,4-Cyclohexanediol* ($C_6H_{12}O_2$) used for illustration in Sec. 1, they are sampled from different environments. Because both of them contain the hydroxy $(-OH)$, which we can call invariant or causal substructure in this case, these two molecules are readily soluble in water. We formalize such a date-generating process of molecule property prediction with a general Structural Causal Model (SCM) [19, 44] in Fig. 4. The abstract data variables are denoted by the nodes and the directed arrows represent the causalities. This SCM illustrates the causalities among variables: ***E*** as the environment, ***S*** as the spurious substructures, ***C*** as the invariant/causal substructures w.r.t ***Y***, ***G*** as the instance molecule graph, ***Y*** as the ground-truth label.

- ***S*** ← ***E*** → ***C***: the environmental variable impacts the underlying data generating distribution. Furthermore, substructures could be divided into causal and spurious ones across all environments.
- ***S*** → ***G*** ← ***C***: an instance molecule graph is made up of the causal and spurious substructures.
- ***C*** → ***Y***: ***Y***, the ground-truth label, is only determined by ***C***. This causation is the focus of our work.

Taking *Cyclopropanol* (C_3H_6O) as an example, we can specify ***E*** as the *3C-ring* scaffold, ***C*** as the substructure hydroxy $(-OH)$, ***S*** as the substructures aside from hydroxy, ***G*** as *Cyclopropanol*, ***Y*** as good water solubility. The good water solubility ***Y*** is only attributed to the invariant substructure hydroxy, i.e. ***C***, rather than other spurious substructures ***S***.

Existing MRL methods do not differentiate invariant and spurious substructures. Hence, the spurious correlations between irrelevant substructures ***S*** and the target label ***Y*** will be encoded to learned molecular representations. When tested on unseen environments, the downstream classifier will be easily misled by these spurious correlations [3].

F Notations

We summarize the notations used in this paper in Table 6.

G Sensitivity to Molecule Segmentation Method

For all experiments in our original paper, we all adopt *breaking retrosynthetically interesting chemical substructures* (BRICS) to segment molecule into substructures, which is widely used in other works related to molecules, e.g., [58, 11]. To investigate the sensitivity of our method to different decomposing strategies, we adopt another molecule segmentation method called *retrosynthetic combinatorial analysis procedure* (RECAP) [36], which is also available as an API in RDKit package. RECAP and BRICS decompose molecules based on two different rules. We conduct experiments on EC50-assay/scaffold/size three datasets and the comparisions are summarized in Table 7. We can

Table 6: Notations.

Notation	Description
e	an environment instance
$\mathbf{e}$	a random variable of e
$\mathcal{E}$	the support of environments
$l(\cdot)$	the loss function
$\mathcal{R}(\cdot)$	the risk function
G	a molecular graph instance
$\mathbf{G}$	a random variable of G
y	a ground-truth label instance
$\mathbf{y}$	a random variable of y
$\mathcal{G}$	a dataset set, i.e. $\{(G, y)\}$
ψ	the environment-inference model
Φ	the molecule encoder
ω	the final predictor
$\mathbf{z}$	the denotation of $\Phi(\mathbf{G})$
f	$\omega \circ \Phi$
κ	the learnable parameters of the Environment Classifier
τ	the learnable parameters of the Conditional GNN
θ	the learnable parameters of Φ and ω
k	hyper-parameter: the environment count
β	hyper-parameter: the trading-off parameter in Eq. 7

Table 7: Comparisons on 3 out-of-distribution datasets in terms of ROC-AUC (%). The best and the runner-up in each columns are highlighted in **bolded** and underlined respectively. Note the baselines except ERM and MixUp all require environment labels. All methods including ours use GIN [60] as backbones. Each experiment is repeated 5 times with mean and standard deviation reported.

Dataset	**EC50**		
Environment	**Assay**	**Scaffold**	**Size**
ERM [54]	69.35 ± 7.38	63.92 ± 2.09	60.94 ± 1.95
IRM [3]	69.94 ± 1.03	63.74 ± 2.15	58.30 ± 1.51
DeepCoral [53]	69.42 ± 3.35	63.66 ± 1.87	56.13 ± 1.77
DANN [17]	66.97 ± 7.19	64.33 ± 1.82	61.11 ± 0.64
MixUp [67]	70.62 ± 2.12	64.53 ± 1.66	62.67 ± 1.41
GroupDro [48]	70.52 ± 3.38	64.13 ± 1.81	59.06 ± 1.50
Ours + RECAP	72.72 ± 3.94	66.34 ± 0.52	**65.48 ± 1.10**
Ours + BRICS	**73.25 ± 1.24**	**66.69 ± 0.34**	65.09 ± 0.90

see that RECAP and BRICS show competitive performance on our model and both outperform the baselines by large margins.

H Future Direction

Sometimes, bio-chemical properties are affected by interactions between substructures. To encode such interactions between substructures into the final learned molecular representation, we utilize the permutation equivariant Set Attention Block (SAB) proposed in Set Transformer [34]. SAB takes a representation set of any size as input and outputs a representation set of equal size. SAB is able to encode pairwise and higher-order interactions between elements in input sets into outputs. We add such a SAB after the Substructure Encoder. For each molecule, we feed the represention of its substructures to SAB to obtain new substruture representations. In this way, the final molecule representation could model interactions between substructures. We conduct experiments on EC50-assay/scaffold/size to examine the performance of adding such a SAB. As demonstrated in Table 8,

Table 8: Comparisons on 3 out-of-distribution datasets in terms of ROC-AUC (%). The best and the runner-up in each columns are highlighted in **bolded** and underlined respectively. Note the baselines except ERM and MixUp all require environment labels. All methods including ours use GIN [60] as backbones. Each experiment is repeated 5 times with mean and standard deviation reported.

Dataset	**EC50**		
Environment	**Assay**	**Scaffold**	**Size**
ERM [54]	69.35 ± 7.38	63.92 ± 2.09	60.94 ± 1.95
IRM [3]	69.94 ± 1.03	63.74 ± 2.15	58.30 ± 1.51
DeepCoral [53]	69.42 ± 3.35	63.66 ± 1.87	56.13 ± 1.77
DANN [17]	66.97 ± 7.19	64.33 ± 1.82	61.11 ± 0.64
MixUp [67]	70.62 ± 2.12	64.53 ± 1.66	62.67 ± 1.41
GroupDro [48]	70.52 ± 3.38	64.13 ± 1.81	59.06 ± 1.50
Ours	**73.25 ± 1.24**	66.69 ± 0.34	**65.09 ± 0.90**
Ours + SAB	73.15 ± 2.69	**67.26 ± 1.54**	64.83 ± 1.07

we can see that adding such a SAB further improves our model on EC50-scaffold. This design is a naive attempt but brings us some valuable insights.

I Limitations

Some studies [26, 16] have empirically shown that existing models designed for OOD learning may fail to outperform the simple ERM [54] model when the environment count is large. Though we can relabel the environment for each molecule according to the new environment partition inferred by our devised environment-inference module, we still need to set the environment count k to a relatively larger value than that of other OOD datasets from other domain, e.g. Camelyon17 [4], which only contains five environments. Thus, using our inferred environment partition, existing models designed for OOD learning might still be inferior to ERM in some cases.

J Potential Negative Impacts

As far as we are concerned, we have not identified any negative social impact of this work.