A Details of Datasets

A.1 Dataset

In this paper, we use 18 publicly benchmark datasets, 12 of which are from GOOD [63] benchmark. They are the combination of 3 datasets (GOOD-HIV, GOOD-ZINC and GOOD-PCBA), 2 types of distribution shift (covariate and concept), and 2 environment-splitting strategies (scaffold and size). The rest 6 datasets are from DrugOOD [13] benchmark, including IC50-Assay, IC50-Scaffold, IC50-Size, EC50-Assay, EC50-Scaffold, and EC50-Size. The prefix denotes the measurement and the suffix denotes the environment-splitting strategies. This benchmark exclusively focuses on covariate shift. We use the latest data released on the official webpage[3] based on the ChEMBL 30 database[4]. We use the default dataset split proposed in each benchmark. For covariate shift, the training, validation and testing sets are obtained based on environments without interactions. For concept shift, a screening approach is leveraged to scan and select molecules in the dataset. Statistics of each dataset are in Table 4.

Table 4: Dataset statistics.

Dataset				Task	Metric	#Train	#Val	#Test	#Tasks
GOOD	HIV	scaffold	covariate	Binary Classification	ROC-AUC	24682	4133	4108	1
			concept	Binary Classification	ROC-AUC	15209	9365	10037	1
		size	covariate	Binary Classification	ROC-AUC	26169	4112	3961	1
			concept	Binary Classification	ROC-AUC	14454	3096	10525	1
	ZINC	scaffold	covariate	Regression	MAE	149674	24945	24946	1
			concept	Regression	MAE	101867	43539	60393	1
		size	covariate	Regression	MAE	161893	24945	17042	1
			concept	Regression	MAE	89418	19161	70306	1
	PCBA	scaffold	covariate	Multi-task Binary Classification	AP	262764	44019	43562	128
			concept	Multi-task Binary Classification	AP	159158	90740	119821	128
		size	covariate	Multi-task Binary Classification	AP	269990	43792	31925	128
			concept	Multi-task Binary Classification	AP	150121	32168	115205	128
DrugOOD	IC50	assay		Binary Classification	ROC-AUC	34953	19475	19463	1
		scaffold		Binary Classification	ROC-AUC	22025	19478	19480	1
		size		Binary Classification	ROC-AUC	37497	17987	16761	1
	EC50	assay		Binary Classification	ROC-AUC	4978	2761	2725	1
		scaffold		Binary Classification	ROC-AUC	2743	2723	2762	1
		size		Binary Classification	ROC-AUC	5189	2495	2505	1

A.2 The Cause of Molecular Distribution Shift

The molecule data can be divided according to different environments, and distribution shifts occur when the source environments of data are different during training and testing. In this work, we investigate three types of environment-splitting strategies, i.e., scaffold, size and assay. And the explanation of each environment are in Table 5.

B Details of Implementation

B.1 Baselines

We adopt the following methods as baselines for comparison, one group of which are common approaches for non-Euclidean data:

- **ERM** [65] minimizes the empirical loss on the training set.
- **IRM** [19] seeks to find data representations across all environments by penalizing feature distributions that have different optimal classifiers.
- **VREx** [40] reduces the risk variances of training environments to achieve both covariate robustness and invariant prediction.

[3] https://drugood.github.io/

[4] http://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBLdb/releases/chembl_30

Table 5: Description of different environment splits leading to molecular distribution shifts.

Environment	Explanation
Scaffold	Molecular scaffold is the fundamental structure of a molecule with desirable bioactive properties. Molecules with the same scaffold belong to the same environment. Distribution shift arises when there is a change in the molecular scaffold.
Size	The size of a molecule refers to the total number of atoms in the molecule. Molecular size is also an inherent structural characteristic of molecular graphs. The distribution shift occurs when size changes.
Assay	Assay is an experimental method as an examination or determination for molecular characteristics. Due to variations in assay environments and targets, activity values measured by different assays often differ significantly. Samples tested within the same assay belong to a single environment, while a change in the assay leads to a distribution shift.

- **GroupDRO** [41] minimizes the loss on the worst-performing group, subject to a constraint that ensures the loss on each group remains close.
- **Coral** [44] encourages feature distributions consistent by penalizing differences in the means and covariances of feature distributions for each domain.
- **DANN** [43] encourages features from different environments indistinguishable by adversarially training a regular classifier and a domain classifier.
- **Mixup** [66] augments data in training through data interpolation.

And the others are graph-specific methods:

- **DIR**[5] [29] discovers the subset of a graph as invariant rationale by conducting interventional data augmentation to create multiple distributions.
- **GSAT**[6] [49] proposes to build an interpretable graph learning method through the attention mechanism and inject stochasticity into the attention to select label-relevant subgraphs.
- **GREA**[7] [47] identifies subgraph structures called rationales by environment replacement to create virtual data points to improve generalizability and interpretability.
- **CAL**[8] [28] proposes a causal attention learning strategy for graph classification to encourage GNNs to exploit causal features while ignoring the shortcut paths.
- **DisC**[9] [32] analyzes the generalization problem of GNNs in a causal view and proposes a disentangling framework for graphs to learn causal and bias substructure.
- **MoleOOD**[10] [25] investigates the OOD problem on molecules and designs an environment inference model and a substructure attention model to learn environment-invariant molecular substructures.
- **CIGA**[11] [26] proposes an information-theoretic objective to extract the desired invariant subgraphs from the lens of causality.

B.2 Implementation

Experiments are conducted on one 24GB NVIDIA RTX 3090 GPU.

[5] https://github.com/Wuyxin/DIR-GNN
[6] https://github.com/Graph-COM/GSAT
[7] https://github.com/liugangcode/GREA
[8] https://github.com/yongduosui/CAL
[9] https://github.com/googlebaba/DisC
[10] https://github.com/yangnianzu0515/MoleOOD
[11] https://github.com/LFhase/CIGA

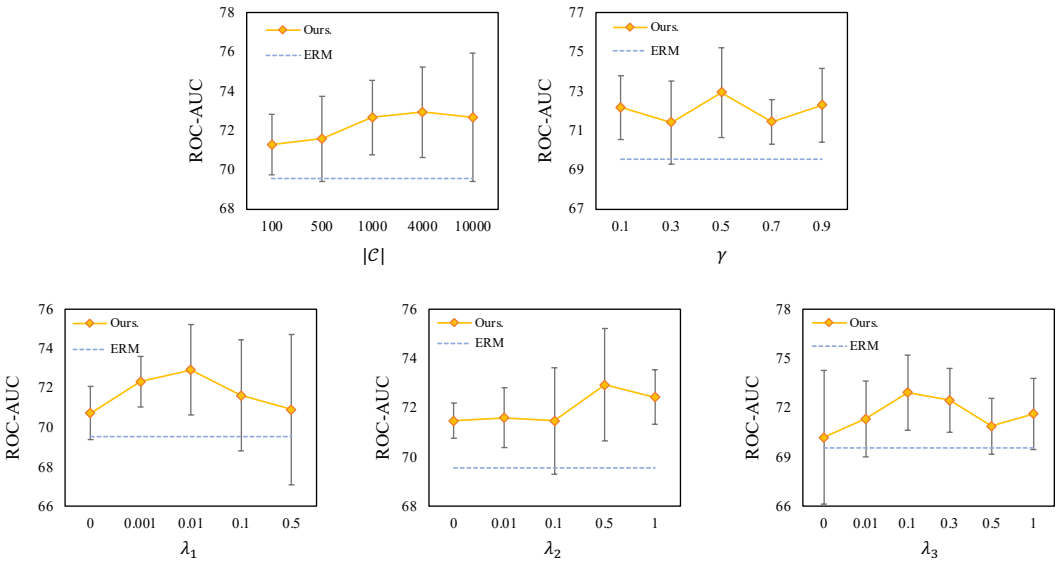

Figure 5: Hyper-parameter sensitivity analysis on the covariate-shift dataset of GOODHIV-Scaffold.

Baselines. For datasets in the GOOD benchmark, we use the results provided in the official leaderboard[12]. For datasets in the DrugOOD benchmark, we use the official benchmark code[13] to get the results for ERM, IRM, Coral, and Mixup on the latest version of datasets. The results for GroupDRO and DANN are not reported due to an error occurred while the code was running. For some baselines that do not have reported results, we implement them using public codes. All of the baselines are implemented using the GIN-Virtual [3, 35] (on GOOD) or GIN [35] (on DrugOOD) as the GNN backbone that is parameterized according to the guidance of the respective benchmark. And we conduct a grid search to select hyper-parameters for all implemented baselines.

Our method. We implement the proposed iMoLD in Pytorch [69] and PyG [70]. For all the datasets, we select hyper-parameters by ranging the code book size $|\mathcal{C}|$ from $\{100, 500, 1000, 4000, 10000\}$, threshold γ from $\{0.1, 0.5, 0.7, 0.9\}$, λ_1 from $\{0.001, 0.01, 0.1, 0.5\}$, λ_2 from $\{0.01, 0.1, 0.5, 1\}$, λ_3 from $\{0.01, 0.1, 0.3, 0.5, 1\}$, and batch size from $\{32, 64, 128, 256, 512\}$. For datasets in DrugOOD, we also select dropout rate from $\{0.1, 0.3, 0.5\}$. The maximum number of epochs is set to 200 and the learning rate is set to 0.001. Please refer to Table 6 for a detailed hyper-parameter configuration of various datasets. The hyper-parameter sensitivity analysis is in Appendix C.1.

C Additional Experimental Results

C.1 Hyper-parameter Sensitivity Analysis

Take the dataset of covariate-shift split of GOODHIV-Scaffold as an example, we conduct extensive experiments to investigate the hyper-parameter sensitivity, and the results are shown in Figure 5. We observe that the performance tends to improve first and then decrease slightly as the size of the codebook $|\mathcal{C}|$ increases. This is because a small codebook limits the expressivity of the model, while too large one cuts the advantage of the discrete space. The effect of the threshold γ is insignificant and there is no remarkable trend. As the λ_1, λ_2 and λ_3 increase, the performance shows a tendency to increase first and then decrease, indicating that $\mathcal{L}_{\text{inv}}$, $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{cmt}}$ are effective and can improve performance within a reasonable range. We also observe that the standard deviation of performance increases as λ_1 increases, which may be due to the fact that too much weight on the self-supervised invariant learning objective may enhance or affect the performance. The standard deviation is the smallest when $\lambda_2 = 0$, suggesting that neural networks may have more stable outcomes by learning adaptively when there are no constraints, but it is difficult to obtain higher performance. While the

[12] https://good.readthedocs.io/en/latest/leaderboard.html
[13] https://github.com/tencent-ailab/DrugOOD

Table 6: Hyper-parameter configuration.

				γ	$\lvert\mathcal{C}\rvert$	batch-size	λ_1	λ_2	λ_3	dropout
GOOD	HIV	scaffold	covariate	0.8	4000	128	0.01	0.5	0.1	-
			concept	0.7	4000	256	0.01	0.5	0.1	-
		size	covariate	0.7	4000	256	0.01	0.5	0.1	-
			concept	0.9	4000	1024	0.01	0.5	0.1	-
	ZINC	scaffold	covariate	0.3	4000	32	0.01	0.5	0.1	-
			concept	0.5	4000	256	0.01	0.5	0.1	-
		size	covariate	0.3	4000	256	0.01	0.5	0.1	-
			concept	0.3	4000	64	0.0001	0.5	0.1	-
	PCBA	scaffold	covariate	0.9	10000	32	0.0001	1	0.1	-
			concept	0.9	10000	32	0.0001	1	0.1	-
		size	covariate	0.9	10000	32	0.0001	1	0.1	-
			concept	0.9	10000	32	0.0001	1	0.1	-
DrugOOD	IC50	assay		0.7	1000	128	0.001	0.5	0.1	0.5
		scaffold		0.9	1000	128	0.0001	0.5	0.1	0.5
		size		0.7	1000	128	0.01	0.5	0.1	0.1
	EC50	assay		0.7	500	128	0.01	0.5	0.1	0.5
		scaffold		0.3	500	128	0.001	0.5	0.1	0.3
		size		0.7	500	128	0.001	0.5	0.1	0.1

standard deviation is the largest when $\lambda_3 = 0$, indicating that the commitment loss in VQ can increase the performance stability.