GOOD: A Graph Out-of-Distribution Benchmark

Shurui Gui*
Texas A&M University
College Station, TX 77843
shurui.gui@tamu.edu

Xiner Li*
Texas A&M University
College Station, TX 77843
lxe@tamu.edu

Limei Wang
Texas A&M University
College Station, TX 77843
limei@tamu.edu

Shuiwang Ji
Texas A&M University
College Station, TX 77843
sji@tamu.edu

Abstract

Out-of-distribution (OOD) learning deals with scenarios in which training and test data follow different distributions. Although general OOD problems have been intensively studied in machine learning, graph OOD is only an emerging area of research. Currently, there lacks a systematic benchmark tailored to graph OOD method evaluation. In this work, we aim at developing an OOD benchmark, known as GOOD, for graphs specifically. We explicitly make distinctions between covariate and concept shifts and design data splits that accurately reflect different shifts. We consider both graph and node prediction tasks as there are key differences when designing shifts. Overall, GOOD contains 8 datasets with 14 domain selections. When combined with covariate, concept, and no shifts, we obtain 42 different splits. We provide performance results on 7 commonly used baseline methods with 10 random runs. This results in 294 dataset-model combinations in total. Our results show significant performance gaps between in-distribution and OOD settings. Our results also shed light on different performance trends between covariate and concept shifts by different methods. Our GOOD benchmark is a growing project and expects to expand in both quantity and variety of resources as the area develops. The GOOD benchmark can be accessed via https://github.com/divelab/GOOD/.

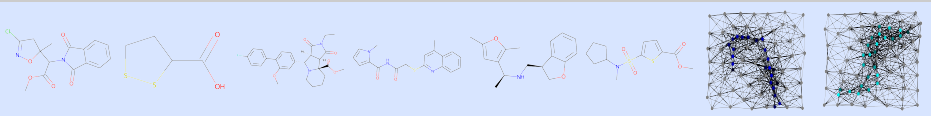
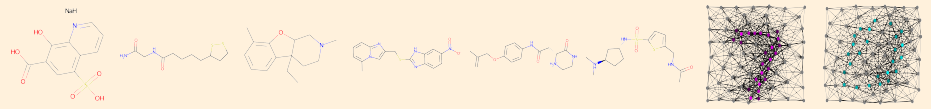
1 Introduction

In machine learning, training and test data are commonly assumed to be i.i.d.. Models designed under this assumption may not perform well when the i.i.d. assumption does not hold. The area of out-of-distribution (OOD) learning deals with scenarios in which training and test data follow different distributions. Two commonly studied OOD settings are covariate shift and concept shift. In covariate shift, the input covariate distributions differ in training and test. In contrast, the conditional relations between inputs and outputs differ in concept shift. Over the years, multiple OOD methods have been proposed [10, 39, 2, 36, 23]. To facilitate evaluations, several benchmarks have also been curated, including DomainBed [15], OoD-Bench [47], and WILDS [22]. Although both general OOD problems and graph analysis [21, 11, 40, 46, 28] have been intensively studied, graph OOD is only an emerging area of research [44, 43, 53, 5]. Some initial attempts have also been made to curate graph OOD benchmarks [19, 7]. However, existing benchmarks lack in several aspects, as detailed in Section 2.

In this work, we develop a systematic graph OOD benchmark, known as GOOD. As design principles for curating GOOD, we strive to (1) create non-trivial performance gaps between training and test

*Equal contributions

Preprint. Under review.

Dataset	**GOOD-HIV**		**GOOD-PCBA**		**GOOD-ZINC**		**GOOD-CMNIST**	
Graph/Node	Graph		Graph		Graph		Graph	
Input(X)	Molecule		Molecule		Molecule		Image-conveted graphs	
Prediction(Y)	HIV replications		Bioassays		Constrained Solubility		Digit numbers	
Domain selection	Scaffold/Size		Scaffold/Size		Scaffold/Size		Color	
# domains	19,089/151		113,760/192		129,959/33		7 (10)	
# examples	41,127		437,929		249,455		70,000	
Source	Real-world		Real-world		Real-world		Semi-artificial	
Task	Binary-classification		Multi-task binary classification		Regression		Multi-class classification	
Metric	ROC-AUC		Average Precision		MAE		Accuracy	
Example domain	Scaffold		Size		Scaffold		Color	
Shift type	Covariate	Concept	Convariate	Concept	Covariate	Concept	Covariate	Concept
# environments	10/1/1	3/1/1	10/1/1	3/1/1	10/1/1	3/1/1	5/1/1	5/1/1
Train example								
Test example								

Dataset	**GOOD-Motif**		**GOOD-Cora**		**GOOD-Arxiv**		**GOOD-CBAS**	
Graph/Node	Graph		Node		Node		Node	
Input(X)	Motif-base graphs		Scientific publications		Arrive papers		A BA-house graph	
Prediction(Y)	Motifs		Publication classes		Subject areas		Node roles	
Domain selection	Base/Size		Word/Degree		Time/Degree		Color	
# domains	5 (3)/5 (3)		218/102		35/547		7 (4)	
# examples	30,000		19,793		169,343		700	
Source	Synthesis		Real-world		Real-world		Synthetic	
Task	Multi-class classification		Multi-class classification		Multi-class classification		Multi-class classification	
Metric	Accuracy		Accuracy		Accuracy		Accuracy	
Example domain	Base		Degree		Time		Color	
Shift type	Covariate	Concept	Convariate	Concept	Covariate	Concept	Covariate	Concept
# environments	3/1/1	3/1/1	10/1/1	3/1/1	10/1/1	3/1/1	5/1/1	5/1/1
Train example								
Test example								

Figure 1: A summary of datasets included in the proposed benchmark. Most datasets have two domain selections. We shown the number of domains for each domain selection. The numbers in parentheses denote the domain numbers in concept shifts. # environments denotes the numbers of training/validation/test environments, respectively. Given an example domain selection, we provide examples to show the covariate and the concept shifts. For a covariate shift, the training and test examples are from different domains. For a concept shift, examples are chosen from the same domain with different labels to show the different domain-output correlations.

data; and (2) provide carefully designed data environments to ensure that the induced distribution shifts are potentially solvable for models. Specifically, GOOD contains 5 graph prediction datasets and 3 node prediction datasets as shown in Fig. 1. Among them, 5 datasets are real-world, 1 dataset is semi-artificial, and 2 datasets are synthetic. For each dataset, we select one or two types of domain. Given a domain in a dataset, we generate no-shift, covariate shift, and concept shift splits for comprehensive comparisons between 7 baselines. Experiment results show significant performance gaps between in-distribution and OOD settings, and performance differences in algorithms for various shift splits.

2 Related Work

Out-of-distribution (OOD) or distribution shift is a longstanding problem in machine learning and artificial intelligence [16, 35, 33, 37]. Several benchmarks have been curated [15, 47, 17, 22, 51] to evaluate different algorithms [10, 39, 34, 2, 23, 36, 50]. Domainbed [15] is an early OOD benchmark for computer vision algorithms. Following DomainBed, OoD-Bench [47] collects datasets and categorizes them into diversity and correlation shifts, which correspond to the covariate and concept shifts in our work. WILDS [22] collects real-world data from wilds and studies domain generalization and subpopulation shift, which are two cases of our covariate shift. Specifically, domain generalization focuses on disjoint training and test domains, while subpopulation shift considers shifts between majority and minority groups, which leads to insufficient training for minority data. With the success of graph neural networks [21, 46, 40, 11, 27, 12, 29, 30], graph OOD problems are gaining growing attention [44, 43, 5, 53, 8, 25, 26]. Ding et al. [7] collect several datasets to compare the performance of well-known baselines and data augmentation methods. DrugOOD [19] is a recent benchmark specifically designed for molecular graph OOD problems. It is curated based on a large-scale bioassay dataset ChEMBL [31] and includes an automated pipeline for obtaining more datasets.

Differences with existing OOD benchmarks. Generalization abilities of OOD algorithms for covariate shift [38] and concept shift [42, 16, 1, 23] may differ significantly. However, existing benchmarks either ignore one of the shifts or fail to compare the two shifts on the same dataset. Firstly, most existing OOD benchmarks include only one type of shift. For example, DrugOOD [19] focuses on domain generalization for molecular datasets, exclusively considering covariate shift. WILDS [22] includes domain generalization and subpopulation shift, which are two cases of covariate shift, while still ignoring concept shift. Secondly, some benchmarks involve both shifts, but each dataset has only one of these two shifts. In contrast, our benchmark generates both shifts for the same datasets to enable comparison between shifts. For example, GDS [7] collects eight datasets but makes no distinctions between the two shifts; among their datasets, we can categorize ColoredMNIST as concept shift and others as covariate shift. OoD-Bench [47] categorizes each collected dataset into diversity shift or correlation shift, which correspond to the covariate and concept shifts in our work, respectively. In addition, we curate various graph datasets with diverse tasks, including single/multi-task graph classification, graph regression, and node classification, while no existing graph OOD benchmark includes graph regression or node classification tasks.

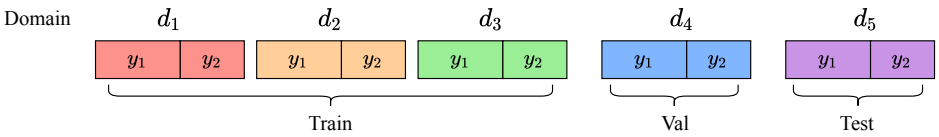
3 The GOOD Benchmark Design

In supervised learning, a model is trained to predict an output Y given an input vector $X \in \mathbb{R}^p$, also known as a covariate vector. The output Y is categorical in classification and continuous in regression problems. In multi-task learning, the output Y becomes a vector, and we consider each task separately. Since the joint distribution $P(Y, X)$ can be written as $P(Y|X)P(X)$, two types of OOD problems are commonly considered; namely covariate and concept shifts. In covariate shift, the input distributions have been shifted between training and test data. Formally, $P^{\text{train}}(X) \neq P^{\text{test}}(X)$ and $P^{\text{train}}(Y|X) = P^{\text{test}}(Y|X)$, where $P^{\text{train}}(\cdot)$ and $P^{\text{test}}(\cdot)$ denote training and test distributions, respectively. In contrast, in concept shift the conditional distribution $P(Y|X)$ has been shifted as $P^{\text{train}}(Y|X) \neq P^{\text{test}}(Y|X)$ and $P^{\text{train}}(X) = P^{\text{test}}(X)$. In this work, we explicitly make distinctions and consider both covariate and concept shifts.

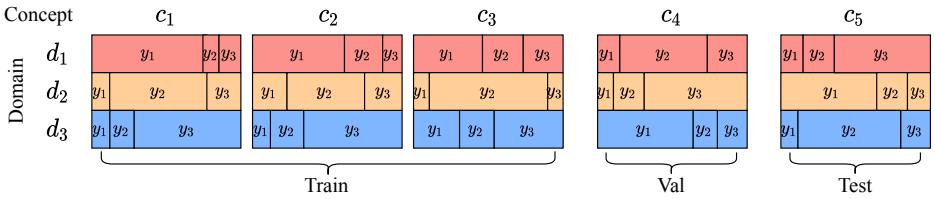
When training and test samples are assumed to be i.i.d., random split is commonly used to split datasets into training and test sets. In contrast, splits in OOD problems should be carefully designed in order to accurately assess the generalization ability of algorithms. In GOOD, we consider both covariate and concept shifts and design data splits to ensure these shifts are reflected in our data splits.

3.1 Covariate shifts

Domain generalization methods follow the covariate shift assumption [4] and assume that the covariate distribution $P(X)$ shifts across splits, while the concept distribution $P(Y|X)$ remains the same. This implies that a shift of $P(X)$ should not cause corresponding shift in $P(Y|X)$. That is, covariate shifts can only happen on input features that are not causally related to Y. Therefore, with prior knowledge, we can manually select and shift one or several of these causally irrelevant features, denoted as X_{var}, to build covariate splits. Different X_{var} feature values indicates different domains, and each domain



(a) Covariate shift split



(b) Concept shift split

Figure 2: (a) Illustration of covariate shift split. Five domains are denoted as different colors, where each domain includes outputs of the same distribution. We sort the dataset according to the domain d_i, then group them into train/validation/test sets. (b) Illustration of concept shift split. Each concept includes all three domains, and each domain has spurious correlations with a specific output in a concept. For example, in concept c_1, the domain colored in red is highly associated with y_1, but this domain corresponds to y_2 in concept c_4. Note that the distribution of concepts in training are similar.

can be viewed as a split. For instance, in the graph ColoredMNIST dataset in which we distinguish hand-written digits with colors, the color is irrelevant with labels. Thus, in our covariate splits, digits with different colors belong to corresponding color domains, and each domain becomes a split.

Formally, given p input features $X \in \mathbb{R}^p$, the invariant features X_{inv} fully determine Y; *i.e.*, $P(Y|X) = P(Y|X_{\text{inv}})$. Here, inv $\subset \{1, \cdots, p\}$ is the index set of invariant features, and X_{inv} denotes a subvector containing elements of X indexed by inv. In contrast, the variant features X_{var} are causally irrelevant with Y, where var $= \{1, \cdots, p\} \setminus$ inv. Note that in practice, it might be hard to strictly separate X_{inv} and X_{var}. Thus, we may only select and shift part of X_{var}, though we use X_{var} throughout this paper for simplicity. For each dataset, possible values of X_{var} are discrete and finite. Therefore, we define each domain by its unique X_{var} values, thereby forming its unique input distribution $P(X)$. Then, a dataset can be viewed as a mixture of D domains as $\mathcal{D} = \{d_1, \ldots, d_D\}$. For a domain d_i, we represent its input distribution as $P_{d_i}(X)$. Specifically, $P_{d_i}(X_{\text{inv}})$ is fixed while $P_{d_i}(X_{\text{var}} = \mathbf{x}_i) = 1$, where $\mathbf{x}_i$ is the value of X_{var} in domain d_i. Since $P_{d_i}(Y|X) = P(Y|X)$, the dataset distribution is

$$P(Y, X) = \sum_{i=1}^{|\mathcal{D}|} w_i P_{d_i}(Y, X) = \sum_{i=1}^{|\mathcal{D}|} w_i P_{d_i}(X) P(Y|X), \tag{1}$$

where w_i is the mixture weight for domain d_i.

We perform covariate shift splits on synthetic, semi-artificial, and real-world datasets. For a synthetic dataset, given the X_{var} feature values of each domain, we generate graphs according to its domain distributions, respectively. For a semi-artificial dataset, given a graph, we generate extra variant features to produce modified graphs that follow the domain distribution. Since we cannot create or modify graphs for real-world datasets, we propose to use carefully designed data splits. As shown in Fig. 2, we sort the graphs by their domain $d_i \in \mathcal{D}$ and then divide the dataset into five domain splits with a specific split ratio, *e.g.*, 20%, for each domain. Finally, the training, validation, and test sets are obtained based on domains without intersections.

3.2 Concept shifts

In contrast to covariate shift, concept shift considers the scenario in which the concept distribution $P(Y|X)$ is shifted across splits, while the covariate distribution $P(X)$ remains the same. Since X_{inv} can fully determine Y, $P(Y|X_{\text{inv}})$ is invariant as mentioned in Sec. 3.1. Thus, shifts of $P(Y|X)$ can only happen with shifts of $P(Y|X_{\text{var}})$. Here, because X_{var} is causally irrelevant with Y, the variant correlation $P(Y|X_{\text{var}})$ between Y and X_{var} in each domain is spurious correlation. Therefore, given

the selected domain variant features X_{var}, we can build concept shift splits by manually creating such spurious correlations of certain rates. For example, the spurious correlation rate between the domain d_i and the output value y_j can be set as $P(Y = y_j|X_{\text{var}} = \mathbf{x}_i) = 90\%$. Specifically, different spurious correlation rates define different concepts, and each concept can be viewed as a split. We use the graph ColoredMNIST dataset as an example as shown in Fig. 2. In every concept, each color domain is highly correlated with a label. Therefore, in our concept splits, different spurious color-label correlation rates determine different concepts, and each concept becomes a split.

Formally, a dataset can be viewed as a mixture of C concepts $\mathcal{C} = \{c_1, \ldots, c_C\}$ in our concept shift split. We use $P_{y_j,d_i}(Y)$ to represent a certain output distribution on value y_j given domain d_i, defined as $P(Y = y_j|X_{\text{var}} = \mathbf{x_i}) = 1$. We first consider the classification case in which Y is categorical. Given a concept c_k, we formulate its 2-D conditional distribution $P_{c_k}(Y|X)$ by describing multiple 1-D distributions; that is, for each domain d_i,

$$P_{c_k}(Y|X_{\text{var}} = \mathbf{x_i}) = \sum_{j=1}^{|\mathcal{Y}|} q_{i,j}^k P_{y_j,d_i}(Y), \tag{2}$$

where $q_{i,j}^k$ is the rate of the spurious correlation in concept c_k between domain d_i and output y_j. In the regression case where Y is continuous, the sum becomes integral. In the multi-task case, Y and y_j become vectors. Since the input distribution is fixed, *i.e.*, $P_{c_k}(X) = P(X)$, the overall dataset distribution can be written as

$$P(Y, X) = \sum_{k=1}^{|\mathcal{C}|} w_k P_{c_k}(Y, X) = \sum_{k=1}^{|\mathcal{C}|} w_k P(X) P_{c_k}(Y|X), \tag{3}$$

where w_k is the mixture weight for concept c_k.

In practice, we create significant concept shifts between training, validation, and test sets, in which the domain-output correlations are completely different. Note that mixing different domain-output correlations can weaken the spuriousness of these correlations. Thus, concepts within the training set are designed to have similar domain-output correlations to guarantee the concept shift between training and test. Concretely, we perform concept shift splits on synthetic, semi-artificial, and real-world datasets. For synthetic datasets, we generate graphs where the domain feature is highly correlated with a specific output according to the preset correlation in the concept. For semi-artificial datasets, given a graph and a concept, we generate extra features as domains to build spurious domain-output correlations. For real-world datasets, we cannot create or modify data. Thus, we propose a screening approach to scan and select graphs in the dataset. Each graph has a probability to be included in a concept c_k according to the value of $q_{i,j}^k$. Specially, due to the domain imbalance and the output imbalance in real-world datasets, we commonly divide the labels into high/low labels and build the domain-output correlations under this setting. Note that since data is limited in real-world datasets, creating a non-trivial concept shift may unavoidably lead to a slight covariate shift.

3.3 Environments

Many current OOD learning algorithms follow the framework of invariant causal predictor (ICP) [34] and invariant risk minimization (IRM) [2], and assume that the training data form groups, known as environments, according to their distributions. This framework assumes the data are similar within an environment and dissimilar across different environments. Specifically, the shift between training and test data, though more significant, should be similarly reflected among different training environments, so that OOD models can potentially grasp the shift between training and test data by learning the shifts among different training environments. Following this strategy, to enhance the OOD generalization ability of models, we use the distribution shift information provided by the difference of training environments to convey the types of shifts expected between training and test data. In covariate shift, environments take the form of domains. During training, models can learn from $P^{\text{train}}(X_{\text{var}})$, which varies across domains, that X_{var} is not causally related to labels, thereby preventing the unknown $P^{\text{test}}(X_{\text{var}})$ from misleading predictions during test. In concept shifts, environments take the form of concepts. By learning from different spurious correlations across training concepts, models can learn that the domain-output correlations $P^{\text{train}}(Y|X_{\text{var}})$ are spurious, thereby avoiding being misled by the new spurious correlation $P^{\text{test}}(Y|X_{\text{var}})$ during test.

Formally, we consider a dataset with a set of E environments $\mathcal{E} = \{e_1, \ldots, e_E\}$, each with distribution $P_e(Y, X)$ for $e \in \mathcal{E}$. In this case, the dataset distribution $P(Y, X) = \sum_e P_e(Y, X)$. Specifically, for both covariate and concept shifts, the distributions P^{train} and P^{test} are weighted combinations of environment distributions $P_e(Y, X)$. With the training and test environments $\mathcal{E}^{\text{train}}, \mathcal{E}^{\text{test}} \subset \mathcal{E}$, we express $P^{\text{train}} = \sum_{e \in \mathcal{E}^{\text{train}}} w_e^{\text{train}} P_e(Y, X)$ and $P^{\text{test}} = \sum_{e \in \mathcal{E}^{\text{test}}} w_e^{\text{test}} P_e(Y, X)$, where w_e^{train} and w_e^{test} are the weights for each training and test environment, respectively.

4 The GOOD Datasets

In this section, we introduce the datasets in GOOD. The benchmark contains 8 datasets, covering multiple tasks and data sources. The tasks include graph single/multi-task binary classification, multi-class classification, regression, and node multi-class classification. Meanwhile, the datasets in GOOD consist of 5 real-world datasets, 1 semi-artificial dataset, and 2 synthetic datasets.

For each dataset, we select one or two domain features. Then we apply covariate and concept shift splits per domain to create diverse distribution shifts between training, OOD validation, and OOD test sets. Finally, we shuffle the training set and divide it into final training set, in-domain (ID) validation set, and in-domain (ID) test set. Summary statistics of datasets are given in Fig. 1. Other specific statistics and data processing details are included in Appendix A.

4.1 Graph prediction tasks

GOOD-HIV is a small-scale real-world molecular dataset adapted from MoleculeNet [45]. The inputs are molecular graphs in which nodes are atoms, and edges are chemical bonds. The task is to predict whether the molecule can inhibit HIV replication. We design splits based on two domain selections, namely, scaffold and size. The first one is Bemis-Murcko scaffold [3] which is the two-dimensional structural base of a molecule. The second one is the number of nodes in a molecular graph, an inevitable structural feature of a graph. Both features cannot determine the label, therefore, both can become major sources of distribution shifts. For each domain selection, the value space for the feature is very large, therefore we cluster graphs with similar domain values into one environment, improving the OOD learning procedure and reducing the training time complexity.

GOOD-PCBA is a real-world molecular dataset from Wu et al. [45]. It includes 128 bioassays, forming 128 binary classification tasks. Due to the extremely unbalanced classes (only 1.4% positive labels), we use the Average Precision (AP) averaged over the tasks as the evaluation metric. GOOD-PCBA uses the same domain selections as GOOD-HIV.

GOOD-ZINC is a real-world molecular property regression dataset from ZINC database [14]. The inputs are molecular graphs with up to 38 heavy atoms, and the task is to predict the constrained solubility [20, 24] of molecules. GOOD-ZINC uses the same domain selections as GOOD-HIV.

GOOD-CMNIST is a semi-artificial dataset. It contains graphs of hand-written digits transformed from MNIST database using superpixel techniques [32]. Following Arjovsky et al. [2], we color digits according to their domains and concepts. Specifically, in covariate shift split, we color digits with 7 different colors, and digits with the first 5 colors, the 6th color, and the 7th color are categorized into training, validation, and test sets. In concept shift split, we color digits with 10 colors. Each color is highly correlated with one digit label in the training set, while colors have weak correlations and no correlation with labels in validation and test sets, respectively.

GOOD-Motif is a synthetic base-motif dataset motivated by Spurious-Motif [44]. Each graph in the dataset is generated by connecting a base graph and a motif, and the label is determined by the motif solely. Instead of combining the base-label spurious correlations and size covariate shift together as in Wu et al. [44], we study covariate and concept shifts separately. Specifically, we generate graphs using five label irrelevant base graphs (wheel, tree, ladder, star, and path) and three label determining motifs (house, cycle, and crane). To create covariate and concept splits, we select the base graph type and the size as domain features. Concrete generation processes are described in Appendix A.

4.2 Node prediction tasks

GOOD-Cora is a citation network adopted from the full Cora dataset [6]. The input is a small-scale citation network graph, in which nodes represent scientific publications and edges represent citation

Table 1: In-distribution and out-of-distribution performance gaps learned with ERM across 42 splits. The metric and domain selections for each dataset are in Fig. 1. ↑ indicates higher values correspond to better performance while ↓ indicates lower values for better performance. ID test results with ID validations are denoted as ID_{ID}, while OOD test results with ID validations and OOD validations are written as OOD_{ID} and OOD_{OOD}, respectively. Note that the no-shift random split only has the ID setting. We report the average values over 10 runs. The standard deviations are listed in Appendix D.

	domain selection 1							domain selection 2						
	covariate			concept			no-shift	covariate			concept			no-shift
	ID_{ID}	OOD_{ID}	OOD_{OOD}	ID_{ID}	OOD_{ID}	OOD_{OOD}	ID_{ID}	ID_{ID}	OOD_{ID}	OOD_{OOD}	ID_{ID}	OOD_{ID}	OOD_{OOD}	ID_{ID}
GOOD-HIV↑	82.79	68.86	69.58	84.22	65.31	72.33	80.86	83.72	58.41	59.94	88.05	44.75	63.26	80.86
GOOD-PCBA↑	33.45	16.87	16.89	25.95	21.34	21.63	33.77	34.31	17.81	17.86	32.54	14.83	15.36	33.77
GOOD-ZINC↓	0.1224	0.1825	0.1995	0.1222	0.1328	0.1306	0.1233	0.1199	0.2569	0.2427	0.1315	0.1418	0.1403	0.1233
GOOD-CMNIST↑	77.96	26.90	28.60	90.00	40.80	42.87	77.30	–	–	–	–	–	–	–
GOOD-Motif↑	92.60	69.97	68.66	92.02	80.87	81.44	92.09	92.28	51.28	51.74	91.73	69.41	70.75	92.09
GOOD-Cora↑	70.43	64.44	64.86	66.05	64.20	64.60	69.41	72.27	55.76	56.30	68.71	60.38	60.54	69.42
GOOD-Arxiv↑	72.69	70.64	71.08	74.76	65.70	67.32	73.02	77.47	58.53	58.91	75.27	61.77	62.99	72.99
GOOD-CBAS↑	89.29	77.57	76.00	89.79	82.22	82.36	99.43	–	–	–	–	–	–	–

links. The task is to predict the publication type, forming a 70-class classification problem. We generate splits based on two domain selections, namely, word and degree. The first one is the word diversity defined by the selected-word-count of a publication, purely irrelevant with the label. The second one is the degree of a node in the graph, which is an inevitable structural feature of nodes and cannot determine the label. Therefore, both can become major sources of distribution shifts.

GOOD-Arxiv is a citation dataset adapted from OGB [18]. The input is a directed graph representing the citation network among the computer science (CS) arXiv papers. Nodes in the graph represent arXiv papers, and directed edges represent citations. The task is predicting the subject area of arXiv CS papers, forming a 40-class classification problem. We generate splits based on two domain selections; *i.e.*, time (publication year) and node degree.

GOOD-CBAS is a synthetic dataset modified from BA-Shapes [48]. The input is a graph created by attaching 80 house-like motifs to a 300-node Barabási–Albert base graph, and the task is to predict the role of nodes, including the top/middle/bottom node of a house-like motif or the node from the base graph, forming a 4-class classification task. Instead of using constant node features, we generate colored features as in GOOD-CMNIST so that OOD algorithms need to tackle node color differences in covariate splits and color-label correlations in concept splits.

5 Experimental Studies

We conduct experiments on 8 datasets with 7 baseline methods. For each dataset, we use the same GNN backbone for all baseline methods for fair comparisons. Specifically, we use GIN-Virtual [46, 13] and GCN [21, 49] as GNN backbones for graph and node prediction tasks, respectively. Note that for GOOD-Motif, we adopt GIN [46] as the GNN backbone since adding virtual nodes does not improve the performance. For all experiments, we select the best checkpoints for ID and OOD tests according to results on ID and OOD validation sets, respectively. Experimental details and hyper-parameter selections are provided in Appendix B. All the datasets, implementation codes, and best checkpoints to reproduce the results in this paper are available at https://github.com/divelab/GOOD/.

5.1 In-distribution versus out-of-distribution performance gap

As introduced in Sec. 1, one principle for designing GOOD is to create non-trivial distribution shifts and performance gaps between training and test data. Equivalently, we expect distinct performance gaps between in-distribution (ID) and out-of-distribution (OOD) settings. To verify performance gaps, we run experiments using empirical risk minimization (ERM) and summarize the results in Table 1. The differences between ID_{ID} and OOD_{ID} or $\text{OOD}_{\text{OOD}}$for each domain selection and distribution shift show the substantial and consistent performance gap between the ID and OOD settings. In addition, for most splits, OOD_{OOD} is better than OOD_{ID}. This implies that OOD validation sets outperform ID validation sets in selecting models with better generalization ability.

5.2 Performance of baseline algorithms

In our benchmark, we conduct experiments with 7 baseline methods. Based on the comparison results, we provide an analysis of the learning strategy of OOD methods.

5.2.1 Baseline methods

We consider empirical risk minimization (ERM) and 6 OOD algorithms as baselines. Firstly, we choose two domain adaptation algorithms which target minimizing feature discrepancies. DANN [10] adversarially trains the regular classifier and a domain classifier to make features indistinguishable. Deep Coral [39] encourages features in different domains to be similar by minimizing the deviation of covariant matrices from different domains. Furthermore, we adopt two invariant learning baselines following the invariant prediction assumption [34]. IRM [2] searches for data representations that perform well across all environments by penalizing feature distributions that have different optimal linear classifiers for each environment. VREx [23] targets both covariate robustness and the invariant prediction. It specifically reduces the variance of risks in test environments by minimizing the risk variances of training environments. By applying fair optimization, GroupDRO [36] tackles the problem that distribution minority lacks sufficient training. This method, known as risk interpolation [23], is achieved by explicitly minimizing the loss in the worst training environment. In addition, we incorporate a data augmentation method Mixup [50] to improve model generalization abilities. Precisely, we follow the implementation of Wang et al. [41], since it is designed for graph data.

5.2.2 Quantitative comparison and analysis

Table 2 shows the OOD_{OOD} and ID_{ID} results of 7 baselines for all splits. Most OOD algorithms have comparable performances with ERM, while many OOD algorithms outperform ERM with certain patterns. Specifically, we observe that the risk interpolation (GroupDRO) and extrapolation (VREx) perform favorably against other methods on multiple datasets and shift splits. VREx outperforms 9 out of 28 OOD splits, evidencing its learning invariance and robustness, especially for covariate shifts in graph prediction tasks. GroupDRO outperforms 6 out of 28 OOD results, showing its advantage in fair optimization. The two feature discrepancy minimization methods, DANN and Deep Coral, do not perform well enough. DANN outperforms 4 out of 28 OOD results, but it is especially suitable for graph concept shift splits. Deep Coral outperforms 2 of them and usually has advantages on ID tests. Mixup exclusively excels at node prediction tasks, which can attribute to its node-specific design [41]. However, it fails at graph prediction tasks due to the simple graph representation mixup strategy. Finally, IRM performs similarly to ERM, showing the difficulty of achieving invariant prediction in non-linear settings. From another perspective, OOD algorithms achieve good performance on certain splits, but they usually cannot perform equally well in corresponding ID settings. This phenomenon reveals the OOD-specific generalization ability of these algorithms. In contrast, Mixup, the data augmentation method, performs equally well in both OOD and ID settings. This indicates its data augmentation nature that benefits the model's generalization ability by making overall progress in learning. We list additional results in Appendix D for references. More empirical results and analysis are in Appendix C.

6 Discussions

GOOD aims to facilitate the development of graph OOD and general OOD algorithms. Our results and comparisons show that current OOD algorithms can improve generalization abilities, but not significantly. In addition, an algorithm might improve performance on one type of shift, but not both. With these observations, future OOD methods can focus on solving one of covariate and concept shifts to improve the specific generalization ability. The improvement might be achieved by managing well-designed model architectures, optimization schemes, or data augmentation strategies. Moreover, models cannot be expected to solve unknown distribution shifts. Thus, we believe using the given environment information to convey the types of shifts expected during testing is a promising direction.

Our GOOD benchmark is a growing project and expects to include more datasets, splits, and methods as the OOD field develops. Currently, we only include one baseline method, *i.e.*, Mixup, specifically designed for graphs, and all other baselines are general OOD algorithms. Even though there are a few recent graph OOD methods [44, 43, 5, 53, 8, 25], we only select from OOD methods that are not too recent. We expect to include more methods in future work, especially graph-related ones. We will

Table 2: ID_{ID} and OOD_{OOD} performances of 7 baselines on 8 datasets. All numerical results are averages across 10 random runs. Numbers in **bold** represent the best results.

covariate	GOOD-HIV↑				GOOD-PCBA↑				GOOD-ZINC↓				GOOD-CMNIST↑	
	scaffold		size		scaffold		size		scaffold		size		color	
	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}
ERM	**82.79**	69.58	83.72	59.94	33.45	16.89	34.31	17.86	0.1224	0.1995	0.1199	0.2427	77.96	28.60
IRM	81.35	67.97	81.33	59.00	33.56	16.90	34.28	**18.05**	0.1213	0.2025	0.1222	0.2403	77.92	27.83
VREx	82.11	**70.77**	83.47	58.53	**33.88**	**16.98**	34.09	17.79	0.1211	0.2094	0.1234	**0.2384**	77.98	28.48
GroupDro	82.60	70.64	83.79	58.98	33.81	16.98	33.95	17.59	**0.1168**	**0.1934**	0.1180	0.2423	77.98	29.07
DANN	81.18	70.63	83.90	58.68	33.63	16.90	34.17	17.86	0.1186	0.2004	0.1188	0.2439	78.00	**29.14**
Deep Coral	82.53	68.61	**84.70**	**60.11**	33.47	16.93	**34.49**	17.94	0.1185	0.2036	**0.1134**	0.2505	**78.64**	29.05
Mixup	82.29	68.88	83.16	59.03	30.22	16.59	30.63	17.06	0.1279	0.2240	0.1255	0.2748	77.40	26.47

covariate	GOOD-Motif↑				GOOD-Cora↑				GOOD-Arxiv↑				GOOD-CBAS↑	
	base		size		word		degree		time		degree		color	
	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}
ERM	92.60	68.66	92.28	51.74	70.43	64.86	72.27	56.30	72.69	71.08	77.47	58.91	89.29	76.00
IRM	92.60	70.65	92.18	51.41	70.27	64.77	72.64	56.28	72.66	71.04	77.50	58.98	91.00	76.00
VREx	92.60	**71.47**	92.25	**52.67**	70.47	64.80	72.25	56.30	72.66	71.12	77.49	58.99	**91.14**	77.14
GroupDro	92.61	68.24	**92.29**	51.95	70.41	64.72	72.18	56.29	72.68	71.15	77.46	**59.08**	90.86	76.14
DANN	92.60	65.47	92.23	51.46	70.66	64.77	72.47	56.10	**72.74**	71.05	77.51	59.00	90.14	**77.57**
Deep Coral	92.61	68.88	92.22	50.97	70.47	64.72	72.16	56.35	72.66	71.07	77.48	58.97	**91.14**	75.86
Mixup	**92.68**	70.08	92.02	51.48	**71.54**	**65.23**	**74.57**	**58.20**	72.49	**71.34**	**77.61**	57.60	73.57	70.57

concept	GOOD-HIV↑				GOOD-PCBA↑				GOOD-ZINC↓				GOOD-CMNIST↑	
	scaffold		size		scaffold		size		scaffold		size		color	
	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}
ERM	84.22	72.33	88.05	63.26	25.95	21.63	32.54	15.36	0.1222	0.1306	0.1315	0.1403	90.00	42.87
IRM	82.89	72.59	**88.62**	59.90	25.89	21.22	32.99	16.07	0.1225	0.1314	0.1278	0.1368	90.02	42.80
VREx	83.84	72.60	88.28	60.23	**26.62**	**22.02**	32.49	15.59	0.1186	0.1270	0.1309	0.1419	89.99	43.31
GroupDro	83.40	**73.64**	88.28	61.37	26.32	21.83	**33.03**	15.99	0.1207	0.1281	**0.1251**	0.1369	**90.02**	**43.32**
DANN	83.87	71.92	87.28	**65.27**	26.07	21.64	32.74	15.78	**0.1172**	**0.1256**	0.1253	**0.1339**	89.94	43.11
Deep Coral	**84.65**	72.97	87.88	62.28	26.38	21.95	32.67	**16.20**	0.1187	0.1279	0.1287	0.1370	89.94	43.16
Mixup	82.36	72.03	87.64	64.87	23.73	19.78	30.23	13.36	0.1353	0.1475	0.1423	0.1522	89.95	40.96

concept	GOOD-Motif↑				GOOD-Cora↑				GOOD-Arxiv↑				GOOD-CBAS↑	
	base		size		word		degree		time		degree		color	
	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}	ID_{ID}	OOD_{OOD}
ERM	92.02	81.44	91.73	**70.75**	66.05	64.60	68.71	60.54	74.76	67.32	**75.27**	62.99	89.79	82.36
IRM	92.00	80.71	91.68	69.77	66.09	64.60	68.58	61.23	74.67	67.41	75.23	62.97	90.71	**83.21**
VREx	**92.05**	**81.56**	91.67	70.24	66.00	64.57	68.45	60.58	74.80	67.37	75.19	**63.00**	89.50	82.86
GroupDro	92.01	81.43	91.67	69.98	66.17	**64.62**	68.37	60.65	74.73	**67.45**	75.19	62.88	90.36	82.00
DANN	92.02	81.33	**91.81**	70.72	66.16	64.51	68.08	60.78	74.73	67.28	75.25	62.91	89.93	82.50
Deep Coral	92.01	81.37	91.68	70.49	66.13	64.58	68.38	60.58	74.77	67.42	75.16	62.85	89.36	82.64
Mixup	91.89	77.63	91.45	67.81	**69.66**	64.44	**70.32**	**63.65**	**74.92**	64.84	72.75	61.28	**93.64**	64.57

also include datasets and domain selections of a larger quantity and variety. In addition, the current benchmark does not consider link prediction tasks [52], which will be added as the project develops.

Acknowledgments and Disclosure of Funding

We thank Jundong Li and Jing Ma for insightful discussions. This work was supported in part by National Science Foundation grants IIS-1955189, IIS-1908198, and IIS-1908220.

References

[1] Rocío Alaiz-Rodríguez and Nathalie Japkowicz. Assessing the impact of changing environments on classifier performance. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 13–24. Springer, 2008.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

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

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1): 151–175, 2010.

[5] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, pages 837–851. PMLR, 2021.

[6] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

[7] Mucong Ding, Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Micah Goldblum, David Wipf, Furong Huang, and Tom Goldstein. A closer look at distribution shifts and out-of-distribution generalization on graphs. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. URL https://openreview.net/forum?id=XvgPGWazqRH.

[8] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. Generalizing graph neural networks on out-of-distribution graphs. *arXiv preprint arXiv:2111.10657*, 2021.

[9] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[11] Hongyang Gao and Shuiwang Ji. Graph U-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019.

[12] Hongyang Gao, Yi Liu, and Shuiwang Ji. Topology-aware graph pooling networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4512–4518, 2021.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[14] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[15] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[16] David J Hand. Classifier technology and the illusion of progress. *Statistical science*, 21(1): 1–14, 2006.

[17] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021.

[18] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[19] Yuanfeng Ji, Lu Zhang, Jiaxiang Wu, Bingzhe Wu, Long-Kai Huang, Tingyang Xu, Yu Rong, Lanqing Li, Jie Ren, Ding 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.

[20] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

[21] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[23] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (REx). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[24] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International conference on machine learning*, pages 1945–1954. PMLR, 2017.

[25] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. OOD-GNN: Out-of-distribution generalized graph neural network. *arXiv preprint arXiv:2112.03806*, 2021.

[26] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987*, 2022.

[27] Meng Liu, Youzhi Luo, Limei Wang, Yaochen Xie, Hao Yuan, Shurui Gui, Haiyang Yu, Zhao Xu, Jingtun Zhang, Yi Liu, et al. DIG: a turnkey library for diving into graph deep learning research. *Journal of Machine Learning Research*, 22(240):1–9, 2021.

[28] Meng Liu, Haiyang Yu, and Shuiwang Ji. Your neighbors are communicating: Towards powerful and scalable graph neural networks. *https://arxiv.org/abs/2206.02059*, 2022.

[29] Yi Liu, Hao Yuan, Lei Cai, and Shuiwang Ji. Deep learning of high-order interactions for protein interface prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 679–687, 2020.

[30] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=givsRXsOt9r.

[31] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic acids research*, 47(D1):D930–D940, 2019.

[32] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

[33] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2011.06.019. URL https://www.sciencedirect.com/science/article/pii/S0031320311002901.

[34] Jonas Peters, Peter Bühlmann, and Nicolai 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.

[35] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[36] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[37] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

[38] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

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

[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ. accepted as poster.

[41] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*, pages 3663–3674, 2021.

[42] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

[43] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*, 2022.

[44] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat seng Chua. Discovering invariant rationales for graph neural networks. In *ICLR*, 2022.

[45] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[46] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

[47] Nanyang Ye, Kaican Li, Lanqing Hong, Haoyue Bai, Yiting Chen, Fengwei Zhou, and Zhenguo Li. OoD-Bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms. *arXiv preprint arXiv:2106.03721*, 2021.

[48] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[49] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. GraphSAINT: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJe8pkHFwS.

[50] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[51] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. NICO++: Towards better benchmarking for domain generalization. *arXiv preprint arXiv:2204.08040*, 2022.

[52] Yangze Zhou, Gitta Kutyniok, and Bruno Ribeiro. OOD link prediction generalization capabilities of message-passing GNNs in larger test graphs. *arXiv preprint arXiv:2205.15117*, 2022.

[53] Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Shift-robust GNNs: Overcoming the limitations of localized graph training data. *Advances in Neural Information Processing Systems*, 34, 2021.

Appendix A: GOOD Dataset Details

GOOD provides 8 datasets with 14 domain selections. For each domain selection, we provide two shift splits and a no shift split, leading to 42 splits. For each covariate/concept shift split, we split data into 5 subsets, namely, training set, in-distribution (ID) validation set, in-distribution (ID) test set, out-of-distribution (OOD) validation set, and out-of-distribution (OOD) test set. For no shift splits, we split data into training, ID validation, and ID test sets. The statistics of splits are listed in Table 3. Meanwhile, the datasets in GOOD consist of 5 real-world datasets, 1 semi-artificial dataset, and 2 synthetic datasets, and we specify the details of splits for each category in the following three paragraphs. The 5 real-world datasets are public datasets [18, 45, 14, 6] and we closely follow the license rules, which are specified in the papers.

Real-world datasets. For covariate shift splits, given a domain selection, we sort the graphs/nodes by their domains and divide the data into a certain number of domains by specifying the split ratio. Then training, validation, and test sets consist of one or several domains. For concept shift splits, we adopt a screening process to build the splits. We first explain this screening process for graph prediction datasets. Each concept has specific domain-label correlations, which come in the form of a set of domain-label probabilities. Consequently, to build a specific concept, each graph has a domain-label probability to be included in this concept. Therefore, we build each concept by scanning the whole dataset and selecting graphs to be included according to their probabilities. The selected graphs form the current concept and are excluded from the dataset scanning. We repeat this procedure to form each of the concepts sequentially, and the last concept includes all the remaining graphs. Similarly, in node classification tasks, we apply the screening process to nodes instead of graphs. That is, we build node selection masks instead of collecting graphs out of datasets. Note that the selection probabilities are relatively similar for those concepts within the training set, while largely dissimilar between the training, validation, and test sets. Also, it is difficult to specify all domain-label probabilities for tasks like 70-classes classification in GOOD-Cora, and impossible for regression tasks. Therefore, we design to group labels as only two categories, namely high/low labels. Then we can build concept splits in a clear sense. For example, we assign a high probability for domain d_i with label 0 in training concepts, while a high probability for domain d_i with label 1 in test concepts.

Semi-artificial datasets. For semi-artificial datasets, we modify graph attributes to create domains instead of directly creating splits with graphs unchanged. Due to the difficulty in modifying graph structures without breaking the semantics of the original graphs, we choose to modify or append the features of nodes in graphs. GOOD currently corporate one semi-artificial dataset GOOD-CMNIST. For GOOD-CMNIST, since the original colors of graphs are in gray-scale, we color graphs by setting node features as 3-channel RGB colors such as red, blue, and cyan. For covariate shift, the training graphs contain 5 color domains, thus forming 5 environments. Other than these 5 colors, the validation and test set graphs are in different colors, respectively. Therefore, in total, we produce 7 different node color features. For concept shift, each digit label is associated with one color, *e.g.*, 0 with red, 1 with green, 2 with blue, etc. Hence we generate 10 different node color features to match the 10-class labels.

Synthetic datasets. For GOOD-Motif, we generate graphs using five label-independent base graphs (wheel, tree, ladder, star, and path) and three label-dependent motifs (house, cycle, and crane). We select the base graph type and the size as domain features to create covariate and concept splits. In the covariate shift splits with base domain, the training set includes graphs with the first three bases, while the validation and the test sets include graphs with base star and path, respectively. In the concept splits with base domain, for 3 different concepts in the training set, each motif is highly correlated to a specific base graph with different correlation rates; *i.e.*, the house-wheel, cycle-tree, and crane-ladder correlations in the training concepts have high probabilities of 99%, 97%, and 95%. In contrast, in the validation and the test sets, these correlations are weak and nonexistent, respectively. Note that only three base graphs are used in this concept shift. In both shift splits with size domain, the base graphs match motifs randomly, while the sizes of base graphs differ. Given five size ranges, in covariate splits, the training set contains three small sizes, while the validation and the test sets include the middle and the largest size ranges, respectively. In concept splits, there are three size ranges which have high, weak, and no correlations with labels for the training, validation, and test sets, respectively. GOOD-CBAS is a color domain dataset with a similar color strategy as GOOD-CMNIST. The main difference in the coloring process is that GOOD-CBAS adopts 4-channel RGBA colors instead of 3-channel colors.

Table 3: Numbers of graphs/nodes in training, ID validation, ID test, OOD validation, and OOD test sets for the 8 datasets.

Dataset	Shift	Train	ID validation	ID test	OOD validation	OOD test	Train	OOD validation	ID validation	ID test	OOD test
		Scaffold					Size				
GOOD-HIV	covariate	24682	4112	4112	4113	4108	26169	4112	4112	2773	3961
	concept	15209	3258	3258	9365	10037	14454	3096	3096	9956	10525
	no shift	24676	8225	8226	-	-	24676	8225	8226	-	-
		Scaffold					Size				
GOOD-PCBA	covariate	262764	43792	43792	44019	43562	269990	43792	43792	48430	31925
	concept	159158	34105	34105	90740	119821	150121	32168	32168	108267	115205
	no shift	262757	87586	87586	-	-	262757	87586	87586	-	-
		Scaffold					Size				
GOOD-ZINC	covariate	149674	24945	24945	24945	24946	161893	24945	24945	20270	17402
	concept	101867	21828	21828	43539	60393	89418	19161	19161	51409	70306
	no shift	149673	49891	49891	-	-	149673	49891	49891	-	-
		Color									
GOOD-CMNIST	covariate	42000	7000	7000	7000	7000					
	concept	29400	6300	6300	14000	14000					
	no shift	42000	14000	14000	-	-					
		Base					Size				
GOOD-Motif	covariate	18000	3000	3000	3000	3000	18000	3000	3000	3000	3000
	concept	12600	2700	2700	6000	6000	12600	2700	2700	6000	6000
	no shift	18000	6000	6000	-	-	18000	6000	6000	-	-
		Word					Degree				
GOOD-Cora	covariate	9378	1979	1979	3003	3454	8213	1979	1979	3841	3781
	concept	7273	1558	1558	3807	5597	7281	1560	1560	3706	5686
	no shift	11875	3959	3959	-	-	11875	3959	3959	-	-
		Time					Degree				
GOOD-Arxiv	covariate	57073	16934	16934	29799	48603	68607	16934	16934	46264	20604
	concept	62083	13303	13303	32560	48094	58619	12561	12561	34222	51380
	no shift	101605	33869	33869	-	-	101605	33869	33869	-	-
		Color									
GOOD-CBAS	covariate	420	70	70	70	70					
	concept	140	140	140	140	140					
	no shift	420	140	140	-	-					

Table 4: General model and hyperparameters for 8 datasets.

Dataset	model	# model layers	batch size	# max epochs	# iterations per epoch	initial learning rate
GOOD-HIV	GIN-Virtual	3	32	200	–	1e-3
GOOD-PCBA	GIN-Virtual	5	32	200	–	1e-3
GOOD-ZINC	GIN-Virtual	3	32	200	–	1e-3
GOOD-CMNIST	GIN-Virtual	5	128	500	–	1e-3
GOOD-Motif	GIN	3	32	200	–	1e-3
GOOD-Cora	GCN	3	4096	100	10	1e-3
GOOD-Arxiv	GCN	3	4096	100	100	1e-3
GOOD-CBAS	GCN	3	1000	200	10	3e-3

Appendix B: Experimental Details

We conduct experiments on 8 datasets, 42 shift splits, with 7 baseline methods. For graph prediction and node prediction tasks, we respectively select strong and commonly acknowledged GNN backbones. For each dataset, we use the same GNN backbone for all baseline methods for fair comparison. For graph prediction tasks, we use GIN-Virtual Node [46, 13] as the GNN backbone. As an exception, for GOOD-Motif we adopt GIN [46] as the GNN backbone, since we observe from experiments that the global information provided by virtual nodes would interrupt the training process here. For node prediction tasks, we adopt GraphSAINT [49] and use GCN [21] as the GNN backbone. Note that the GNN backbone for Mixup is a modified GCN according to the implementation of Wang et al. [41].

Our code is implemented based on PyTorch Geometric [9]. For all the experiments, we use the Adam optimizer, with a weight decay of 0 and a dropout rate of 0.5. The GNN model and the number of convolutional layers for each dataset are specified in Table 4. We use mean global pooling and the RELU activation function, and the dimension of the hidden layer is 300. The batch size, the maximum number of epochs, (the number of iterations per epoch for node prediction tasks,) and initial learning rate are also specified in Table 4. In the training process, all models are trained to converge. For computation, we use one NVIDIA GeForce RTX 2080 Ti for each single experiment.

Hyperparameters for OOD algorithms. For each OOD algorithm, we choose one algorithm-specific hyperparameter to tune. For IRM and Deep Coral, we tune the weight for penalty loss. For VREx, we tune the weight for VREx's loss variance penalty. For GroupDRO, we tune the step size. For DANN, we tune the weight for domain classification penalty loss. For Mixup, we tune the alpha value of its Beta function. The Beta function is used to randomize the lamda weight, which is the weight for mixing two instances up. For each split of a dataset and each OOD algorithm, we search from a hyperparameter set of 3 values and select the optimal one based on validation metric scores. The hyperparameter sets and the optimal hyperparameters are listed in Appendix E.

Reproducibility. For all experiments, we select the best checkpoints for ID and OOD tests according to ID and OOD validation sets, and report the results. All the datasets, codes, and best checkpoints to reproduce the results in this paper are available at https://github.com/divelab/GOOD/. Simple usage guideline and examples are as Appendix F. For coding details and instructions, please refer to the GOOD package documents https://good.readthedocs.io.

Appendix C: Empirical Results and Analysis

We analyze empirical results based on the numerical results in Appendix D. Notations are the same as in the main paper. By comparing ID_{ID} with OOD_{ID}, and ID_{OOD} with OOD_{OOD} results, we can observe substantial and consistent gaps between both pairs of ID/OOD performances. In all cases, the OOD performance is significantly worse than the corresponding ID performance, demonstrating that all our splits meaningfully produce distribution shifts. For most splits, the OOD_{ID} performance is worse than the OOD_{OOD} performance. This implies that OOD validation sets outperform ID validation sets in selecting models with better generalization ability, since the OOD validation set contains similar distribution shifts as the OOD test set. However, this is not always the case, since models do not possess sufficient generalization ability, and cannot always deal with distribution shifts during test even these shifts are similar to that during validation. In addition, for the no shift random split, where only ID setting exists, performances are comparable with covariate/concept ID_{ID} settings but constantly a bit worse; this is explainable in the sense that no shift splits include more unfiltered OOD data, and the greater diversity of data adds to training difficulty.

In most cases, algorithms have comparable performances on the same split. Many OOD algorithms outperform ERM with certain patterns, and the number of outperforming cases reveals essential information about the generalization ability of an algorithm. As mentioned in Section 5.2 of the main paper, the risk interpolation (GroupDRO) and extrapolation (VREx) perform favorably against other methods on multiple datasets and shift splits. VREx outperforms 12 out of 56 OOD splits and 7 out of 70 ID splits, evidencing its learning invariance and robustness, especially for covariate shifts in graph prediction tasks. GroupDRO outperforms 8 out of 56 OOD splits and 10 out of 70 ID splits, showing its advantage in fair optimization. As for the feature discrepancy minimization methods, DANN outperforms 11 out of 56 OOD splits and 7 out of 70 ID splits. It is especially suitable for graph concept shift splits, showing the benefits of explicitly using environment information to train a domain classifier. Deep Coral outperforms 5 out of 56 OOD splits and 18 out of 70 ID splits, having advantages on ID tests. Mixup outperforms 8 out of 56 OOD splits and 20 out of 70 ID splits. Mixup exclusively excels at node prediction tasks, which can attribute to its node-specific network design [41]. However, when node features are simple, the linear interpolation in Mixup might not work, as the case for GOOD-CBAS. Moreover, it fails at graph prediction tasks due to the direct and simple graph representation mixup strategy [41]. Finally, IRM performs similarly to ERM and outperforms 7 out of 56 OOD splits and 4 out of 70 ID splits, showing the difficulty of achieving invariant prediction in non-linear settings.

When OOD algorithms achieve good performance on certain splits, they usually cannot perform equally well in the corresponding ID settings. This phenomenon reveals the OOD-specific generalization ability of these algorithms. In contrast, Mixup, the data augmentation method, performs equally well in both OOD and ID settings. This indicates its data augmentation nature that benefits the model's generalization ability by making overall progress in learning. Also, the deviation minimization of feature covariant matrices benefits Deep Coral's performances in ID settings.

Insights on future OOD method development. Our results and comparisons show that current OOD algorithms can improve generalization abilities, but not significantly, underscoring the need for OOD methods that are more robust and better-performing in practice. Additionally, in practice models

cannot be expected to solve unknown distribution shifts. Thus, we believe using the given environment information in training to convey the types of shifts expected during testing is a promising direction. Similarly, we suggest using OOD validation containing possible distribution shift types of OOD test set to select models that are potentially better at our target generalization abilities. Moreover, we observe distinct performance difference on covariate and concept shifts for OOD algorithms, demonstrating that OOD algorithms might need shift-specific design to maximize generalization ability for one type of shift. In this case, future OOD methods can focus on solving one of covariate or concept shift. Inspired by a recent work [51], we expect to evaluate covariate and concept shifts using shift-specific metrics. Therefore, covariate and concept shifts can be viewed, solved and evaluated separately. On top of that, OOD generalization abilities can be improved by managing well-designed model architectures, optimization schemes, or data augmentation strategies. To solve graph OOD problems, it is critical that methods should be specifically designed for graphs. For example, Mixup's specifically designed node prediction network [41] is quite well-performing while the graph prediction network [41] adopted directly from image field [50] shows no advantage. One possible reason is that functional data augmentation for graphs should consider the complex structure of graphs, so simple strategies like direct graph representation Mixup can cause topological mismatch.

Appendix D: Complete Dataset Results

D.1 Complete numerical results

We report the complete results of ID/OOD test performances from ID/OOD validation for 7 baselines on 8 datasets in a series of tables, as shown in Table 5-18.

Table 5: ID/OOD test performances from ID/OOD validation on GOOD-HIV with scaffold domain. Numerical results are average ± standard deviation across 10 random runs. Numbers in **bold** represent the best results. The metric and domain selections for each dataset are listed in each table. Note that the no shift random split only has the ID setting.

GOOD-HIV	scaffold								
	covariate				concept				no shift
ROC-AUC	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	**82.79**±1.10	68.86±2.10	80.84±0.57	69.58±1.99	84.22±0.85	65.31±3.49	82.64±1.58	72.33±1.04	80.86±0.63
IRM	81.35±0.83	67.31±1.94	80.74±0.87	67.97±2.46	82.89±1.27	66.06±3.06	81.93±1.11	72.59±0.45	**81.06**±0.61
VREx	82.11±1.48	69.25±1.84	81.09±1.56	**70.77**±1.35	83.84±1.09	66.48±2.16	82.55±1.09	72.60±0.82	80.57±0.65
GroupDRO	82.60±1.25	69.24±2.20	81.60±1.40	70.64±1.72	83.40±0.67	65.89±2.78	82.01±1.28	**73.64**±0.86	80.27±0.90
DANN	81.18±1.37	70.05±1.02	80.85±1.42	70.63±1.82	83.87±0.99	**66.57**±2.30	82.58±1.14	71.92±1.23	80.82±0.64
Deep Coral	82.53±1.01	68.00±2.62	**82.02**±0.69	68.61±1.70	**84.65**±1.73	65.74±3.49	**82.99**±2.09	72.97±1.04	80.73±0.83
Mixup	82.29±1.34	**70.66**±3.56	81.27±1.83	68.88±2.40	82.36±1.94	65.94±2.96	80.81±2.26	72.03±0.53	80.28±1.27

Table 6: Performance on GOOD-HIV with size domain.

GOOD-HIV	size								
	covariate				concept				no shift
ROC-AUC	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	83.72±1.06	58.41±2.53	82.94±1.65	59.94±2.86	88.05±0.67	44.75±2.92	82.97±2.73	63.26±2.47	80.86±0.63
IRM	81.33±1.13	58.41±1.79	79.93±1.00	59.00±2.74	**88.62**±0.86	44.17±4.58	**85.67**±1.20	59.90±3.15	**81.06**±0.61
VREx	83.47±1.11	**60.24**±2.54	83.20±1.35	58.53±2.22	88.28±0.88	44.43±3.77	84.93±1.32	60.23±1.70	80.57±0.65
GroupDRO	83.79±0.68	59.50±2.21	82.03±1.45	58.98±1.84	88.28±0.84	45.42±3.34	84.41±1.72	61.37±2.79	80.27±0.90
DANN	83.90±0.68	58.68±3.02	82.17±2.49	58.68±1.83	87.28±1.12	43.26±3.68	81.83±2.56	**65.27**±3.75	80.82±0.64
Deep Coral	**84.70**±1.17	59.72±3.66	**83.89**±0.83	**60.11**±3.53	87.88±0.57	**47.56**±3.55	84.80±1.17	62.28±1.42	80.73±0.83
Mixup	83.16±1.12	60.13±2.06	82.03±1.72	59.03±3.07	87.64±0.81	46.19±4.40	81.20±1.97	64.87±1.77	80.28±1.27

Table 7: Performance on GOOD-PCBA with scaffold domain.

GOOD-PCBA	scaffold								
	covariate				concept				no shift
AP	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	33.45±0.42	16.87±0.49	32.62±1.02	16.89±0.55	25.95±0.94	21.34±0.89	25.95±1.06	21.63±0.97	33.77±0.31
IRM	33.56±0.57	16.94±0.35	32.86±0.65	16.90±0.42	25.89±0.42	21.05±0.39	25.78±0.62	21.22±0.39	33.36±0.31
VREx	**33.88**±0.74	17.01±0.27	**33.27**±1.18	**16.98**±0.29	**26.62**±0.64	**21.98**±0.86	26.45±0.73	**22.02**±0.88	33.61±0.49
GroupDRO	33.81±0.55	**17.06**±0.28	32.32±0.88	16.98±0.26	26.32±0.41	21.61±0.53	26.03±0.75	21.83±0.61	33.35±0.53
DANN	33.63±0.46	16.86±0.46	32.62±0.90	16.90±0.33	26.07±0.29	21.23±0.44	25.99±0.46	21.64±0.37	33.47±0.32
Deep Coral	33.47±0.57	16.84±0.55	32.50±1.49	16.93±0.59	26.38±0.82	21.70±0.66	**26.46**±0.83	21.95±0.76	**33.77**±0.48
Mixup	30.22±0.33	16.68±0.37	29.92±0.46	16.59±0.42	23.73±0.53	19.58±0.56	23.25±0.79	19.78±0.44	30.35±0.26

Table 8: Performance on GOOD-PCBA with size domain.

GOOD-PCBA	size								
	covariate				concept				no shift
AP	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	34.31±0.57	17.81±0.43	34.29±0.56	17.86±0.38	32.54±0.83	14.83±0.61	31.96±0.93	15.36±0.54	33.77±0.31
IRM	34.28±0.46	**17.94**±0.30	34.29±0.54	**18.05**±0.29	32.99±0.89	**15.76**±0.54	32.55±0.89	16.07±0.52	33.36±0.31
VREx	34.09±0.29	17.76±0.43	34.07±0.28	17.79±0.41	32.49±0.76	15.22±0.53	32.06±0.74	15.59±0.57	33.61±0.49
GroupDRO	33.95±0.51	17.49±0.46	33.92±0.45	17.59±0.46	**33.03**±0.32	15.62±0.53	**32.58**±0.45	15.99±0.43	33.35±0.53
DANN	34.17±0.34	17.86±0.47	34.09±0.34	17.86±0.48	32.74±0.50	15.40±0.46	32.25±0.77	15.78±0.39	33.47±0.32
Deep Coral	**34.49**±0.43	17.76±0.39	**34.41**±0.43	17.94±0.38	32.67±1.01	15.63±0.77	32.14±1.21	**16.20**±0.72	**33.77**±0.48
Mixup	30.63±0.65	17.09±0.58	30.55±0.72	17.06±0.54	30.23±1.02	13.00±0.81	29.97±1.13	13.36±0.66	30.35±0.26

Table 9: Performance on GOOD-ZINC with scaffold domain.

GOOD-ZINC	scaffold								
	covariate				concept				no shift
MAE	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	0.1224±0.0029	0.1825±0.0129	0.1384±0.0075	0.1995±0.0114	0.1222±0.0052	0.1328±0.0060	0.1225±0.0055	0.1306±0.0038	0.1233±0.0045
IRM	0.1213±0.0044	0.1787±0.0094	0.1463±0.0128	0.2025±0.0145	0.1225±0.0036	0.1319±0.0039	0.1223±0.0035	0.1314±0.0042	0.1200±0.0049
VREx	0.1211±0.0025	0.1771±0.0099	0.1512±0.0130	0.2094±0.0118	0.1186±0.0035	0.1273±0.0044	0.1186±0.0036	0.1270±0.0040	0.1247±0.0021
GroupDRO	**0.1168**±0.0045	0.1784±0.0083	**0.1373**±0.0079	**0.1934**±0.0114	0.1207±0.0037	0.1284±0.0042	0.1210±0.0038	0.1281±0.0041	0.1222±0.0059
DANN	0.1186±0.0030	0.1762±0.0108	0.1404±0.0133	0.2004±0.0113	**0.1172**±0.0044	**0.1262**±0.0051	**0.1171**±0.0040	**0.1256**±0.0048	0.1217±0.0057
Deep Coral	0.1185±0.0045	**0.1752**±0.0080	0.1438±0.0097	0.2036±0.0158	0.1187±0.0066	0.1287±0.0077	0.1191±0.0070	0.1279±0.0067	**0.1156**±0.0055
Mixup	0.1279±0.0056	0.1951±0.0124	0.1575±0.0191	0.2240±0.0258	0.1353±0.0068	0.1479±0.0056	0.1357±0.0067	0.1475±0.0059	0.1418±0.0064

Table 10: Performance on GOOD-ZINC with size domain.

GOOD-ZINC	size								
	covariate				concept				no shift
MAE	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	0.1199±0.0060	0.2569±0.0138	0.1323±0.0092	0.2427±0.0068	0.1315±0.0073	0.1418±0.0057	0.1346±0.0079	0.1403±0.0065	0.1233±0.0045
IRM	0.1222±0.0059	**0.2536**±0.0227	0.1317±0.0100	0.2403±0.0106	0.1278±0.0077	0.1403±0.0138	0.1302±0.0084	0.1368±0.0119	0.1200±0.0049
VREx	0.1234±0.0054	0.2560±0.0212	0.1327±0.0089	**0.2384**±0.0098	0.1309±0.0064	0.1462±0.0139	0.1352±0.0092	0.1419±0.0090	0.1247±0.0021
GroupDRO	0.1180±0.0054	0.2598±0.0213	0.1293±0.0069	0.2423±0.0097	**0.1251**±0.0066	0.1402±0.0091	**0.1273**±0.0089	0.1369±0.0076	0.1222±0.0059
DANN	0.1188±0.0048	0.2555±0.0183	0.1303±0.0057	0.2439±0.0056	0.1253±0.0034	**0.1371**±0.0084	0.1297±0.0055	**0.1339**±0.0048	0.1217±0.0057
Deep Coral	**0.1134**±0.0071	0.2545±0.0159	**0.1269**±0.0092	0.2505±0.0073	0.1287±0.0041	0.1415±0.0074	0.1310±0.0058	0.1370±0.0052	**0.1156**±0.0055
Mixup	0.1255±0.0071	0.2776±0.0215	0.1317±0.0145	0.2748±0.0167	0.1423±0.0062	0.1599±0.0115	0.1459±0.0073	0.1522±0.0064	0.1418±0.0064

Table 11: Performance on GOOD-CMNIST with color domain.

GOOD-CMNIST	color								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	77.96±0.34	**26.90**±1.91	76.26±0.56	28.60±2.01	90.00±0.17	40.80±1.60	89.43±0.33	42.87±0.72	**77.30**±0.35
IRM	77.92±0.30	25.81±2.70	75.91±2.89	27.83±1.84	90.02±0.12	41.70±0.54	89.44±0.43	42.80±0.38	77.28±0.21
VREx	77.98±0.32	26.75±2.21	76.42±0.74	28.48±2.08	89.99±0.18	41.26±1.40	89.42±0.24	43.31±0.78	77.03±0.44
GroupDRO	77.98±0.38	26.51±0.95	**76.57**±0.84	29.07±2.62	**90.02**±0.27	41.47±0.95	89.33±0.32	**43.32**±0.75	77.01±0.33
DANN	78.00±0.43	26.82±1.64	76.02±1.77	**29.14**±2.93	89.94±0.19	**41.86**±0.68	89.49±0.39	43.11±0.64	77.15±0.48
Deep Coral	**78.64**±0.48	26.16±1.59	76.11±1.60	29.05±2.19	89.94±0.17	41.28±0.86	89.42±0.28	43.16±0.56	77.12±0.32
Mixup	77.40±0.22	26.24±2.43	74.86±1.13	26.47±1.73	89.95±0.25	39.59±1.11	**89.63**±0.31	40.96±0.81	76.62±0.37

Table 12: Performance on GOOD-Motif with base domain.

GOOD-Motif	base								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	92.60±0.03	69.97±1.94	92.43±0.20	68.66±3.43	92.02±0.05	**80.87**±0.65	92.05±0.04	81.44±0.45	92.09±0.04
IRM	92.60±0.02	70.30±1.23	92.51±0.08	70.65±3.18	92.00±0.02	80.41±0.27	92.00±0.03	80.71±0.46	92.04±0.06
VREx	92.60±0.03	**72.23**±2.28	**92.52**±0.12	**71.47**±2.75	**92.05**±0.06	80.71±0.79	**92.06**±0.04	**81.56**±0.35	92.09±0.07
GroupDRO	92.61±0.03	70.29±2.02	92.48±0.13	68.24±1.94	92.01±0.04	80.32±0.57	92.02±0.05	81.43±0.70	92.09±0.08
DANN	92.60±0.03	69.04±1.90	92.38±0.16	65.47±5.35	92.02±0.04	80.57±0.59	92.04±0.03	81.33±0.52	**92.10**±0.06
Deep Coral	92.61±0.03	70.43±1.44	92.37±0.27	68.88±3.61	92.01±0.05	80.27±0.72	92.04±0.03	81.37±0.42	92.09±0.07
Mixup	**92.68**±0.05	69.30±1.00	92.48±0.17	70.08±2.06	91.89±0.03	77.57±0.56	91.89±0.01	77.63±0.57	92.04±0.06

Table 13: Performance on GOOD-Motif with size domain.

GOOD-Motif	size								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	92.28±0.10	51.28±1.94	**92.13**±0.16	51.74±2.27	91.73±0.10	69.41±0.91	91.78±0.16	**70.75**±0.56	92.09±0.04
IRM	92.18±0.09	49.65±1.31	91.99±0.12	51.41±3.30	91.68±0.13	68.55±1.79	91.70±0.12	69.77±0.88	92.04±0.06
VREx	92.25±0.08	48.87±0.99	92.09±0.14	**52.67**±2.87	91.67±0.13	68.73±1.23	91.76±0.20	70.24±0.72	92.09±0.07
GroupDRO	**92.29**±0.09	49.21±1.50	92.12±0.10	51.95±2.80	91.67±0.14	68.28±1.50	91.74±0.15	69.98±0.86	92.09±0.08
DANN	92.23±0.08	49.92±2.63	92.04±0.25	51.46±3.41	**91.81**±0.16	**69.68**±1.40	91.69±0.32	70.72±1.16	**92.10**±0.06
Deep Coral	92.22±0.13	**52.70**±3.04	92.05±0.13	50.97±1.76	91.68±0.10	68.76±0.95	**91.78**±0.09	70.49±0.84	92.09±0.07
Mixup	92.02±0.10	49.98±2.19	91.90±0.14	51.48±3.35	91.45±0.13	66.42±1.07	91.39±0.22	67.81±1.13	92.04±0.06

Table 14: Performance on GOOD-Cora with word domain.

GOOD-Cora	word								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	70.43±0.47	64.44±0.55	70.31±0.39	64.86±0.38	66.05±0.22	64.20±0.56	66.16±0.37	64.60±0.17	69.41±0.30
IRM	70.27±0.33	**64.83**±0.25	70.07±0.23	64.77±0.36	66.09±0.32	64.16±0.61	66.19±0.36	64.60±0.16	69.42±0.38
VREx	70.47±0.40	64.49±0.55	70.35±0.42	64.80±0.28	66.00±0.26	64.20±0.54	66.37±0.41	64.57±0.18	69.43±0.29
GroupDRO	70.41±0.46	64.49±0.66	70.38±0.29	64.72±0.34	66.17±0.30	**64.38**±0.34	66.36±0.44	**64.62**±0.17	69.46±0.25
DANN	70.66±0.36	64.72±0.22	70.51±0.47	64.77±0.42	66.16±0.31	64.29±0.33	66.14±0.41	64.51±0.19	69.25±0.33
Deep Coral	70.47±0.37	64.63±0.38	70.37±0.32	64.72±0.36	66.13±0.18	64.38±0.36	66.34±0.40	64.58±0.18	69.46±0.27
Mixup	**71.54**±0.63	63.07±1.52	**72.14**±0.70	**65.23**±0.56	**69.66**±0.45	64.22±0.33	**69.56**±0.45	64.44±0.10	**70.56**±0.35

Table 15: Performance on GOOD-Cora with degree domain.

GOOD-Cora	degree								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	72.27±0.57	55.76±0.82	72.51±0.57	56.30±0.49	68.71±0.56	60.38±0.33	68.43±0.28	60.54±0.44	69.42±0.30
IRM	72.64±0.45	55.77±0.46	72.75±0.36	56.28±0.63	68.58±0.40	61.00±0.34	68.53±0.38	61.23±0.32	69.40±0.38
VREx	72.25±0.65	55.46±0.87	72.49±0.59	56.30±0.50	68.45±0.44	60.05±0.72	68.37±0.33	60.58±0.42	69.42±0.29
GroupDRO	72.18±0.58	55.44±0.91	72.66±0.41	56.29±0.43	68.37±0.79	60.03±0.88	68.34±0.25	60.65±0.31	69.40±0.30
DANN	72.47±0.37	55.50±0.60	72.51±0.42	56.10±0.59	68.08±1.05	59.65±0.94	68.51±0.36	60.78±0.38	69.24±0.34
Deep Coral	72.16±0.53	55.52±0.93	72.57±0.37	56.35±0.38	68.38±0.76	60.22±0.55	68.30±0.30	60.58±0.40	69.43±0.30
Mixup	**74.57**±0.54	**57.21**±1.12	**74.34**±0.56	**58.20**±0.67	**70.32**±0.59	**63.49**±0.23	**70.44**±0.53	**63.65**±0.39	**70.87**±0.47

Table 16: Performance on GOOD-Arxiv with time domain.

GOOD-Arxiv	time								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	72.69±0.19	70.64±0.47	72.66±0.17	71.08±0.23	74.76±0.18	**65.70**±0.42	73.68±0.49	67.32±0.24	73.02±0.14
IRM	72.66±0.15	70.55±0.33	72.58±0.20	71.04±0.16	74.67±0.15	65.69±0.55	73.53±0.46	67.41±0.16	72.90±0.14
VREx	72.66±0.18	70.54±0.33	72.58±0.21	71.12±0.24	74.80±0.14	65.40±0.54	73.72±0.43	67.37±0.27	72.84±0.09
GroupDRO	72.68±0.17	70.67±0.31	72.46±0.26	71.15±0.20	74.73±0.18	65.57±0.66	73.55±0.34	**67.45**±0.15	72.91±0.12
DANN	**72.74**±0.11	70.57±0.40	**72.67**±0.20	71.05±0.29	74.73±0.15	65.42±0.53	73.99±0.35	67.28±0.16	73.00±0.12
Deep Coral	72.66±0.18	70.59±0.29	72.54±0.09	71.07±0.21	74.77±0.16	65.53±0.63	73.40±0.32	67.42±0.22	72.95±0.09
Mixup	72.49±0.26	**71.05**±0.31	72.55±0.23	**71.34**±0.14	**74.92**±0.32	64.01±0.50	**74.55**±0.18	64.84±0.59	**73.19**±0.16

Table 17: Performance on GOOD-Arxiv with degree domain.

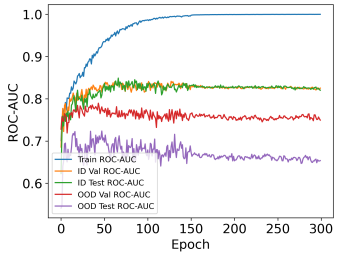
GOOD-Arxiv	degree								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	77.47±0.12	58.53±0.16	77.18±0.23	58.91±0.23	**75.27**±0.16	**61.77**±0.29	74.74±0.19	62.99±0.20	72.99±0.12
IRM	77.50±0.11	**58.70**±0.12	77.15±0.29	58.98±0.28	75.23±0.11	61.49±0.36	74.64±0.42	62.97±0.27	72.92±0.07
VREx	77.49±0.11	58.59±0.21	77.33±0.17	58.99±0.16	75.19±0.14	61.61±0.32	74.64±0.22	**63.00**±0.33	72.88±0.09
GroupDRO	77.46±0.18	58.46±0.21	77.16±0.20	**59.08**±0.16	75.19±0.14	61.59±0.56	**74.92**±0.20	62.88±0.24	72.98±0.10
DANN	77.51±0.08	58.56±0.16	77.19±0.29	59.00±0.18	75.25±0.08	61.43±0.40	74.76±0.25	62.91±0.22	72.97±0.10
Deep Coral	77.48±0.13	58.63±0.21	77.16±0.26	58.97±0.20	75.16±0.15	61.77±0.37	74.89±0.12	62.85±0.29	72.91±0.12
Mixup	**77.61**±0.15	57.43±0.27	**77.47**±0.29	57.60±0.31	72.75±0.38	60.60±1.01	72.31±0.84	61.28±0.87	**73.03**±0.14

Table 18: Performance on GOOD-CBAS with color domain.

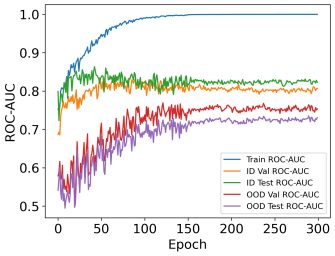
GOOD-CBAS	color								
	covariate				concept				no shift
Accuracy	ID validation		OOD validation		ID validation		OOD validation		ID validation
	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test	OOD test	ID test
ERM	89.29±3.16	77.57±2.96	**89.72**±3.20	76.00±3.00	89.79±1.18	82.22±1.81	90.14±1.10	82.36±0.97	99.43±0.45
IRM	91.00±1.28	77.00±2.21	87.43±4.05	76.00±3.39	90.71±0.87	81.50±1.46	90.21±0.91	**83.21**±0.54	99.64±0.46
VREx	**91.14**±2.72	77.71±2.03	88.43±1.81	77.14±1.43	89.50±1.13	**82.50**±1.47	90.21±0.96	82.86±1.26	99.64±0.46
GroupDRO	90.86±2.92	77.71±2.00	89.71±2.12	76.14±1.78	90.36±0.91	81.22±1.78	91.00±1.01	82.00±1.46	99.72±0.33
DANN	90.14±3.16	**79.14**±2.40	86.71±4.78	**77.57**±2.86	89.93±1.25	80.50±1.31	89.78±1.01	82.50±0.72	99.65±0.33
Deep Coral	**91.14**±2.02	77.86±2.22	88.14±2.43	75.86±3.06	89.36±1.87	81.93±1.36	90.14±0.98	82.64±1.40	**99.79**±0.28
Mixup	73.57±8.72	73.72±6.60	73.00±9.27	70.57±7.41	**93.64**±0.57	63.57±1.43	**92.86**±1.19	64.57±1.81	98.43±1.72

D.2 Metric score curves

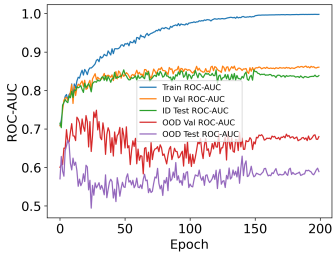
We also report the metric score curves for 8 datasets in Fig. 3-10. Note that we only include the curves for ERM with all splits, while all curve figures for other algorithms are available at our GitHub repository.



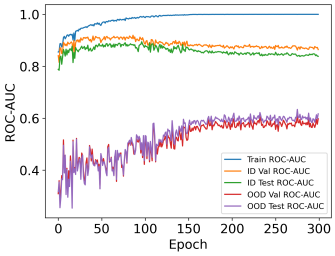
(a) covariate shift + scaffold domain



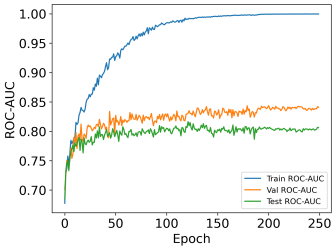
(b) concept shift + scaffold domain



(c) covariate shift + size domain

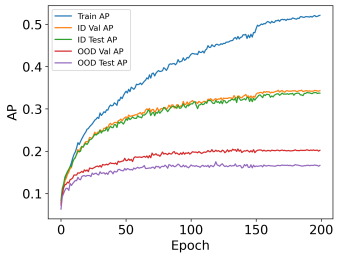


(d) concept shift + size domain

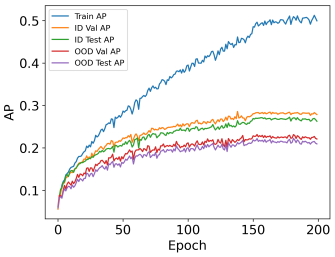


(e) no shift

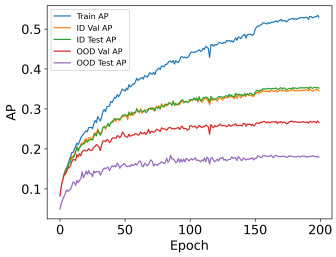
Figure 3: Metric score curves for ERM on GOOD-HIV. Note that we omit the domain selection for no shift since the two cases make no difference in results.



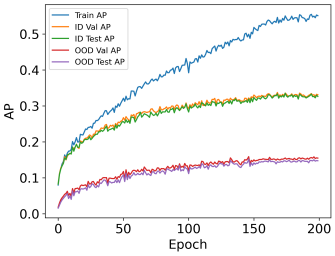
(a) scaffold domain, covariate shift



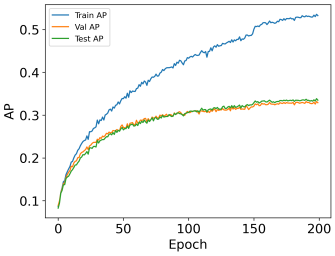
(b) concept shift + scaffold domain



(c) covariate shift + size domain



(d) concept shift + size domain

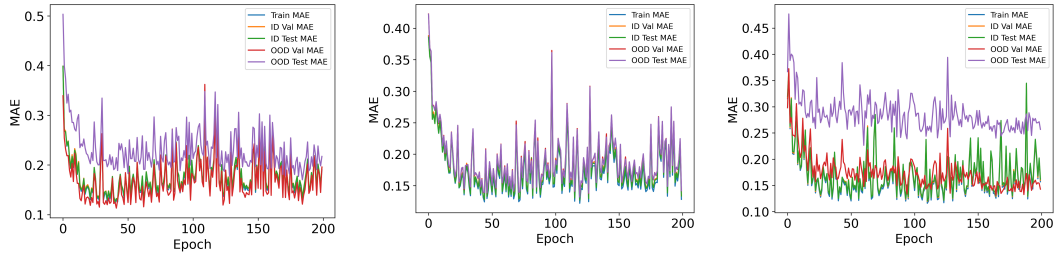


(e) no shift

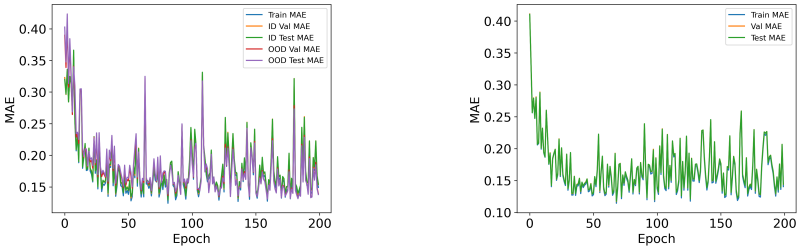
Figure 4: Metric score curves for ERM on GOOD-PCBA.

D.3 Comparison between training, validation and test scores

To directly view performance gaps between training and test data, we compare training, validation, and test scores in Table 19. These comparisons reveal the distribution shift by definition.

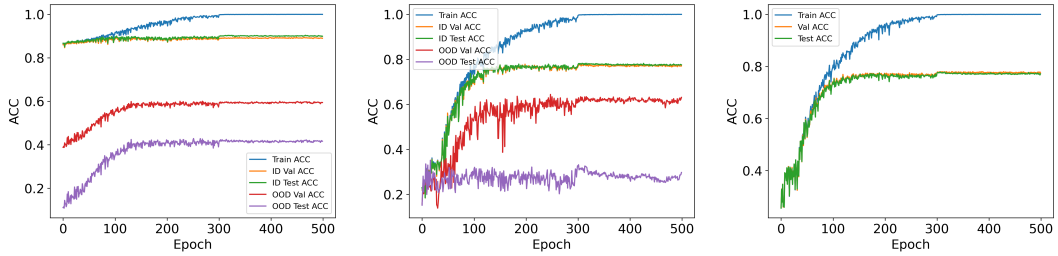


(a) covariate shift + scaffold domain (b) concept shift + scaffold domain (c) covariate shift + size domain



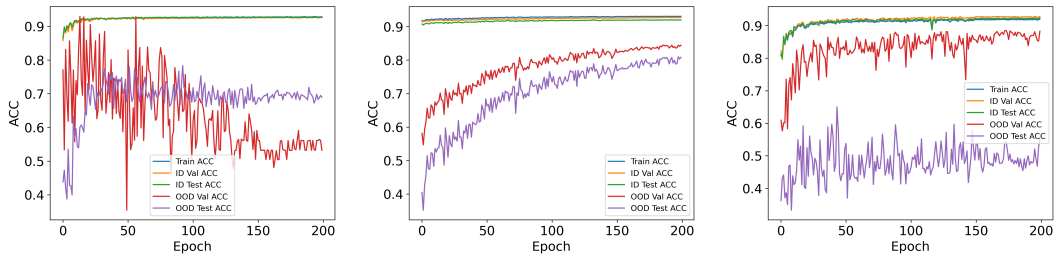
(d) concept shift + size domain (e) no shift

Figure 5: Metric score curves for ERM on GOOD-ZINC.

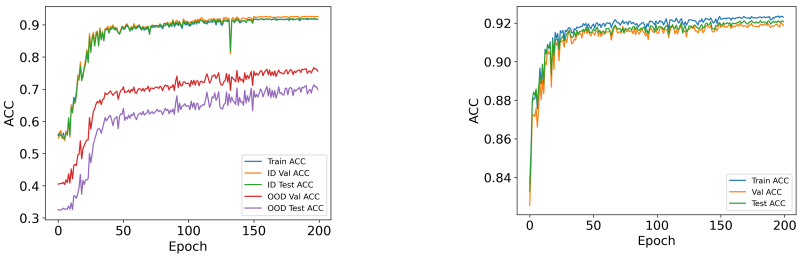


(a) concept shift + color domain (b) covariate shift + color domain (c) no shift + color domain

Figure 6: Metric score curves for ERM on GOOD-CMNIST.

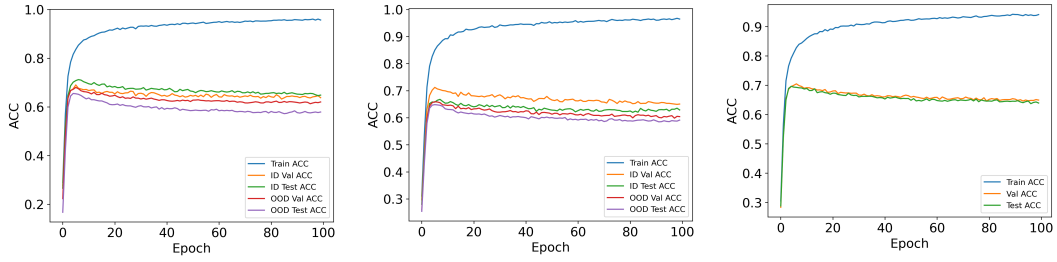


(a) covariate shift + base domain (b) concept shift + base domain (c) covariate shift + size domain

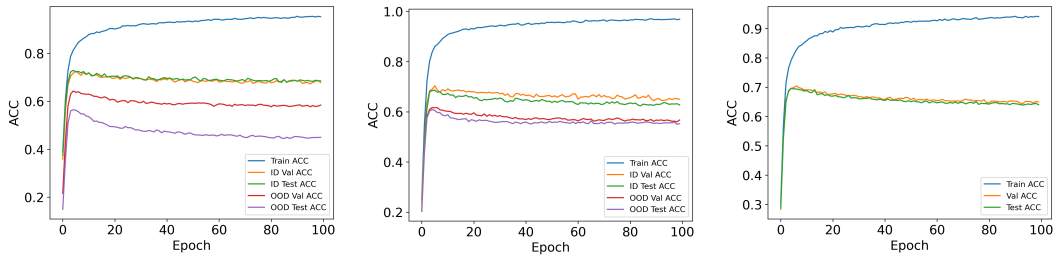


(d) concept shift + size domain (e) no shift

Figure 7: Metric score curves for ERM on GOOD-Motif.

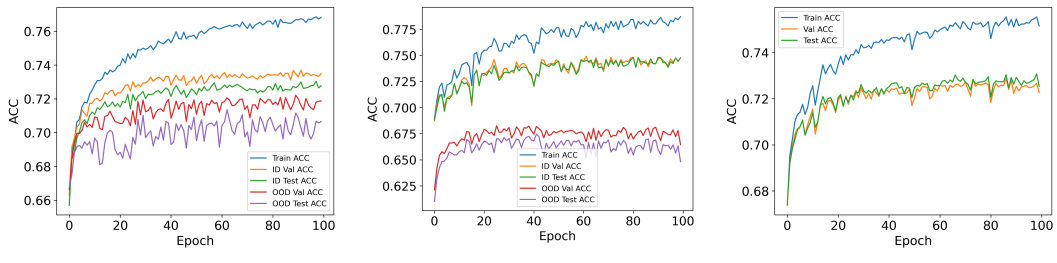


(a) covariate shift + word domain (b) concept shift + word domain (c) no shift + word domain

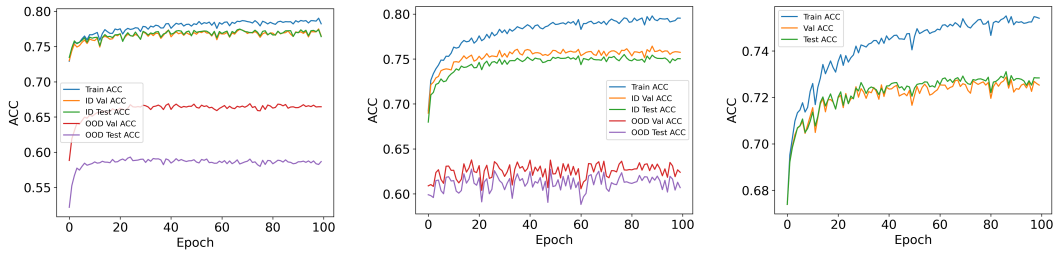


(d) covariate shift + degree domain (e) concept shift + degree domain (f) no shift + degree domain

Figure 8: Metric score curves for ERM on GOOD-Cora.

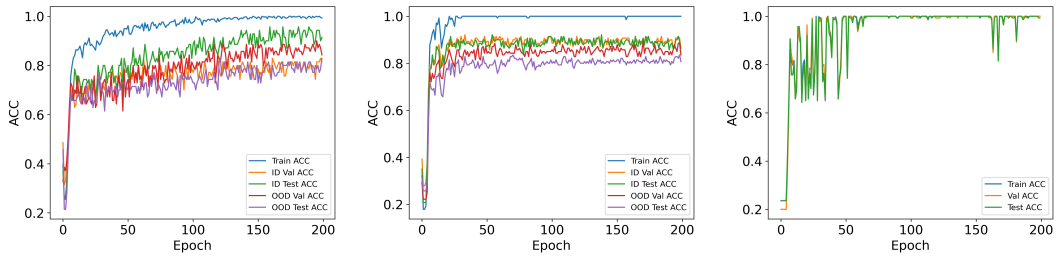


(a) covariate shift + time domain (b) concept shift + time domain (c) no shift + time domain



(d) covariate shift + degree domain (e) concept shift + degree domain (f) no shift + degree domain

Figure 9: Metric score curves for ERM on GOOD-Arxiv.



(a) covariate shift + color domain (b) concept shift + color domain (c) no shift + color domain

Figure 10: Metric score curves for ERM on GOOD-CBAS.

Table 19: Comparison between training, validation and test scores for ERM on 8 datasets. The scores are evaluated on the final model of a random run. ↑ indicates higher values correspond to better performance while ↓ indicates lower values for better performance.

Dataset	Domain	Shift	ID validation				OOD validation			
			Train	Validation	ID test	OOD test	Train	Validation	ID test	OOD test
GOOD-HIV↑	scaffold	covariate	99.40	84.11	82.62	68.65	91.63	78.94	81.49	69.57
		concept	94.04	83.56	82.63	58.28	99.56	76.92	80.55	72.43
	size	covariate	99.76	86.34	83.58	59.26	87.84	74.86	82.14	54.68
		concept	98.53	91.93	88.38	50.07	99.97	61.48	83.89	63.38
GOOD-PCBA↑	scaffold	covariate	51.57	33.74	32.75	17.01	47.95	20.75	32.76	16.49
		concept	47.86	28.42	25.86	20.40	49.57	22.00	26.08	21.24
	size	covariate	52.91	33.89	34.17	18.26	52.91	27.23	34.17	18.26
		concept	55.16	34.56	33.64	16.45	55.78	17.32	33.36	16.49
GOOD-ZINC↓	scaffold	covariate	0.1183	0.1224	0.1224	0.1895	0.1380	0.1421	0.1409	0.2159
		concept	0.1074	0.1138	0.1128	0.1243	0.1086	0.1232	0.1141	0.1239
	size	covariate	0.1167	0.1215	0.1214	0.2581	0.1210	0.1313	0.1259	0.2352
		concept	0.1117	0.1142	0.1162	0.1370	0.1244	0.1286	0.1298	0.1279
GOOD-CMNIST↑	color	covariate	93.54	77.83	77.17	26.39	97.07	64.39	76.97	29.49
		concept	92.15	89.46	90.00	36.68	99.13	60.01	89.40	42.60
GOOD-Motif↑	base	covariate	92.90	92.70	92.67	70.43	91.60	90.53	91.93	66.57
		concept	93.06	92.74	91.96	80.37	93.08	84.45	92.00	80.78
	size	covariate	92.06	92.87	92.40	55.63	91.85	88.57	92.17	54.33
		concept	91.80	92.63	91.81	69.93	91.97	76.67	91.81	71.35
GOOD-Cora↑	word	covariate	83.61	69.02	70.79	65.38	83.61	67.98	70.79	65.38
		concept	84.96	71.22	65.77	64.84	84.96	65.92	65.77	64.84
	degree	covariate	82.98	72.46	72.36	56.25	81.29	64.30	72.92	56.49
		concept	85.98	70.41	68.61	60.51	83.80	61.68	68.49	60.93
GOOD-Arxiv↑	time	covariate	76.74	73.70	73.00	70.30	76.69	72.21	72.66	71.16
		concept	78.23	74.92	74.51	65.17	76.17	68.25	73.39	67.01
	degree	covariate	79.02	77.50	77.39	58.27	78.48	66.89	76.96	59.03
		concept	79.79	76.43	75.43	61.85	77.33	63.77	74.41	62.75
GOOD-CBAS↑	color	covariate	94.05	82.86	82.86	70.00	99.76	80.00	91.43	77.14
		concept	100.00	92.14	87.86	82.86	100.00	89.29	90.71	81.43

Appendix E: Complete OOD Parameter Selections

Following Appendix B, in this section we specify the hyperparameter tune set and selection for each algorithm on each dataset in Table 20-27.

Table 20: OOD hyperparameter selections on GOOD-HIV.

GOOD-HIV	tune set			scaffold			size		
				covariate	concept	no shift	covariate	concept	no shift
ERM	–	–	–	–	–	–	–	–	–
IRM	10.0	0.1	1.0	1.0	0.1	0.1	10.0	0.1	0.1
VREx	10.0	1000.0	100.0	100.0	10.0	100.0	10.0	1000.0	100.0
GroupDRO	0.01	0.1	0.001	0.1	0.01	0.001	0.01	0.001	0.001
DANN	0.1	1.0	0.01	1.0	0.1	0.01	0.01	1.0	0.01
Deep Coral	0.01	1.0	0.1	0.1	0.01	0.01	0.1	0.01	0.01
Mixup	1.0	2.0	0.4	2.0	0.4	2.0	2.0	0.4	2.0

Appendix F: GOOD Usage Guidelines

We provide the open-source GOOD project to reproduce all reported results and extend OOD datasets and algorithms. The GOOD project enables automatic dataset downloads, easy data loading, and handy start-up code to work with any GOOD dataset or method. Meanwhile, we provide various modular utilities for OOD method development. Reproduction is available and effortless with given test scripts and automatic re-loading of our best checkpoints. Please refer to our GitHub repository for installation details, along with more documentation and usage information at https:

Table 21: OOD hyperparameter selections on GOOD-PCBA.

GOOD-PCBA	tune set			scaffold			size		
				covariate	concept	no shift	covariate	concept	no shift
ERM	–	–	–	–	–	–	–	–	–
IRM	1.0	0.1	10.0	0.1	0.1	0.1	0.1	1.0	0.1
VREx	10.0	100.0	1.0	10.0	100.0	10.0	1.0	10.0	10.0
GroupDRO	0.01	0.001	0.1	0.01	0.001	0.1	0.1	0.01	0.1
DANN	0.01	0.001	0.1	0.01	0.01	0.01	0.01	0.01	0.01
Deep Coral	0.1	0.01	1.0	0.01	0.1	1.0	0.1	0.1	1.0
Mixup	1.0	2.0	0.4	1.0	2.0	1.0	2.0	1.0	1.0

Table 22: OOD hyperparameter selections on GOOD-ZINC.

GOOD-ZINC	tune set			scaffold			size		
				covariate	concept	no shift	covariate	concept	no shift
ERM	–	–	–	–	–	–	–	–	–
IRM	1.0	0.1	0.01	0.01	0.01	0.01	0.01	0.01	0.01
VREx	100.0	10.0	1000.0	1000.0	100.0	100.0	1000.0	100.0	100.0
GroupDRO	0.01	0.1	0.001	0.1	0.001	0.1	0.001	0.001	0.1
DANN	0.01	0.001	0.1	0.001	0.001	0.1	0.01	0.1	0.1
Deep Coral	0.1	0.01	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Mixup	1.0	0.4	2.0	0.4	1.0	1.0	1.0	0.4	1.0

Table 23: OOD hyperparameter selections on GOOD-CMNIST.

GOOD-CMNIST	tune set			color		
				covariate	concept	no shift
ERM	–	–	–	–	–	–
IRM	0.1	1.0	0.01	0.1	0.1	1.0
VREx	0.01	0.1	1.0	1.0	0.01	0.1
GroupDRO	0.001	0.01	0.1	0.1	0.01	0.1
DANN	0.1	0.01	0.001	0.1	0.01	0.001
Deep Coral	0.1	0.01	0.001	0.1	0.0001	0.001
Mixup	1.0	2.0	0.4	1.0	0.4	0.4

Table 24: OOD hyperparameter selections on GOOD-Motif.

GOOD-Motif	tune set			base			size		
				covariate	concept	no shift	covariate	concept	no shift
ERM	–	–	–	–	–	–	–	–	–
IRM	1.0	10.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1
VREx	1000.0	100.0	10.0	1000.0	1000.0	1000.0	100.0	10.0	1000.0
GroupDRO	0.001	0.01	0.1	0.001	0.001	0.1	0.1	0.01	0.1
DANN	0.1	1.0	0.01	0.01	0.1	0.01	0.1	0.1	0.01
Deep Coral	1.0	0.1	0.01	1.0	0.1	0.1	1.0	0.01	0.1
Mixup	1.0	2.0	0.4	2.0	0.4	0.4	1.0	0.4	0.4

Table 25: OOD hyperparameter selections on GOOD-Cora.

GOOD-Cora	tune set			word			degree		
				covariate	concept	no shift	covariate	concept	no shift
ERM	–	–	–	–	–	–	–	–	–
IRM	0.1	1.0	10.0	10.0	0.1	0.1	1.0	10.0	0.1
VREx	100.0	10.0	1.0	100.0	10.0	1.0	10.0	10.0	1.0
GroupDRO	0.001	0.01	0.1	0.1	0.01	0.01	0.1	0.01	0.01
DANN	0.01	0.1	0.001	0.001	0.1	0.001	0.01	0.1	0.001
Deep Coral	0.01	0.1	1.0	0.01	0.01	1.0	1.0	0.1	0.01
Mixup	0.4	2.0	1.0	0.4	2.0	0.4	0.4	2.0	2.0

Table 26: OOD hyperparameter selections on GOOD-Arxiv.

GOOD-Arxiv	tune set			time			degree		
				covariate	concept	no shift	covariate	concept	no shift
ERM	–	–	–	–	–	–	–	–	–
IRM	0.1	1.0	10.0	0.1	1.0	1.0	0.1	1.0	1.0
VREx	1.0	100.0	10.0	100.0	1.0	100.0	1.0	100.0	1.0
GroupDRO	0.001	0.01	0.1	0.01	0.001	0.1	0.1	0.001	0.1
DANN	0.1	0.001	0.01	0.001	0.001	0.001	0.01	0.001	0.1
Deep Coral	0.1	0.01	1.0	0.1	1.0	1.0	0.1	1.0	0.1
Mixup	2.0	1.0	0.4	1.0	0.4	0.4	2.0	1.0	1.0

Table 27: OOD hyperparameter selections on GOOD-CBAS.

GOOD-CBAS	tune set			color		
				covariate	concept	no shift
ERM	–	–	–	–	–	–
IRM	10.0	1.0	0.1	10.0	1.0	10.0
VREx	100.0	1.0	10.0	100.0	100.0	1.0
GroupDRO	0.1	0.01	0.001	0.1	0.001	0.01
DANN	0.01	0.001	0.1	0.1	0.1	0.01
Deep Coral	0.01	0.1	0.001	0.01	0.001	0.01
Mixup	0.4	1.0	2.0	0.4	2.0	0.4

//github.com/divelab/GOOD/. GOOD uses the GPL3.0 license. Please refer to the GOOD GitHub repository for license details.

We provide simple and standardized examples for dataset loading and training/evaluation procedures.

F.1 GOOD dataset loading

Code listing 1 shows two ways to import a GOOD dataset and specify the domain selection and shift split.

F.2 GOOD taining/test pipeline

Code listing 2 provides a script to use the main function of the training/evaluation pipeline, following the three steps of loading the config, specifying the model, and executing the task.

```python
# Directly import
from GOOD.data.good_datasets.good_hiv import GOODHIV
hiv_datasets, hiv_meta_info = GOODHIV.load(
        dataset_root,
        domain='scaffold',
        shift='covariate',
        generate=False
)
# Or use register
from GOOD import register as good_reg
hiv_datasets, hiv_meta_info = good_reg.datasets['GOODHIV'].load(
        dataset_root,
        domain='scaffold',
        shift='covariate',
        generate=False
)
cmnist_datasets, cmnist_meta_info = ood_reg.datasets['GOODCMNIST'].load(
        dataset_root,
        domain='color',
        shift='concept',
        generate=False
)
```

Listing 1: **GOOD dataset loader**

```python
# Load a config
from GOOD import config_summoner
from GOOD.utils.args import args_parser
from GOOD.utils.logger import load_logger
args = args_parser()
config = config_summoner(args)
load_logger(config)

# Load a GNN, a dataloader, and an OOD algorithm
from GOOD.kernel.pipeline import initialize_model_dataset
from GOOD.ood_algorithms.ood_manager import load_ood_alg
model, loader = initialize_model_dataset(config)
ood_algorithm = load_ood_alg(config.ood.ood_alg, config)

# Start training
from GOOD.kernel.train import train
train(model, loader, ood_algorithm, config)
# Or start a test
from GOOD.kernel.evaluation import evaluate
test_stat = evaluate(model, loader, ood_algorithm, 'test', config)
```

Listing 2: **GOOD taining/test pipeline**