Understanding Risk Extrapolation (REx) and when it finds Invariant Relationships

Jeroen Hofland
Supervisor(s): Jesse Krijthe, Rickard Karlsson, Stephan Bongers
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Generalizing models for new unknown datasets is a common problem in machine learning. Algorithms that perform well for test instances with the same distribution as their training dataset often perform severely on new datasets with a different distribution. This problem is caused by distributional shifts between the training of the model and applying that model to a test domain. This paper addresses whether and in what situations *Risk Extrapolation (REx)* can tackle this problem of *Out-Of-Distribution generalization* by exploiting *invariant relationships.* These relationships are based on features that are invariant across all domains. By learning these relationships, REx aims to learn the concept of the problem we are trying to solve. We show in what situations REx can learn these invariant relationships and when it does not. We translate the definition of an invariant relationship into a homoscedastic synthetic dataset with either covariate, confounded, anti-causal, or hybrid shift. We expose REx to experiments in sample complexity, the number of training domains, and the training domain distance. We show that REx performs better for invariant prediction in situations with larger sample sizes and training domain distance and that if these criteria are met, REx performs equivalently in all four distributional shifts. We also compare REx to Invariant- and Empirical Risk Minimization and show that; REx is less sensitive and thus robust to the shifting of the average distributional variance in the training domains; REx asymptotically out-performs the methods in the more complex distributional shifts.

1 Introduction

Machine learning algorithms have seen remarkable success in the past years. They aim to learn a model of a training dataset in order for it to perform well on new (test) instances and do this very well. But when applied to a new domain (i.e. dataset) this performance is not always guaranteed. Beery, van Horn, and Perona (2018) showed that the performance of these algorithms, when applied in a real world problem such as recognizing cows, often plummets in new unseen domains. *Domain generalization* is a principle which aims tackle this. It tries to learn the concept of a problem in a domain in order for it to generalize well for instances of an new unseen domain (Wang et al., 2021).This paper addresses the problem called *Out-Of-Domain generalization (OOD)*, also known as out-of-distribution generalization. The OOD generalization problem refers to a setting of domain generalization where the unknown test distribution shifts from the training distribution. This shift is called *distributional shift.* it can be problematic for machine learning algorithms focusing exclusively on test accuracy in the training domain as a performance metric (Shen et al., 2021).

But how does an algorithm know what features of an instance are important to give a high weight to succeed on a new unseen domain? In other words, what features does it need to learn and what features does it need to ignore? Features that do not change through domains and directly correlate with the correct classification of an instance are called *Invariant Relationships.* They are properties that belong to the core concept of what the method is trying to learn. Properties that help classify an instance in one domain but have no correlation to the correct label in another domain should not be learned. These properties are called *Spurious Correlations* since they seem to correlate to a label in a domain but do not correlate with the actual concept we are trying to learn.

This can be illustrated using the example of Beery et al. (2018). The goal is to classify cows in a picture such as a figure 1. The algorithm trained on pictures of cows in a grass field may correlate grass with the label of a cow. When presented with pictures of cows in a different domain, for example, a cow in the desert, the algorithm may not be able to correctly classify the cow since it may rely on the spurious correlations (scenery) and not on the invariant relationships (properties of a cow) of the instance.

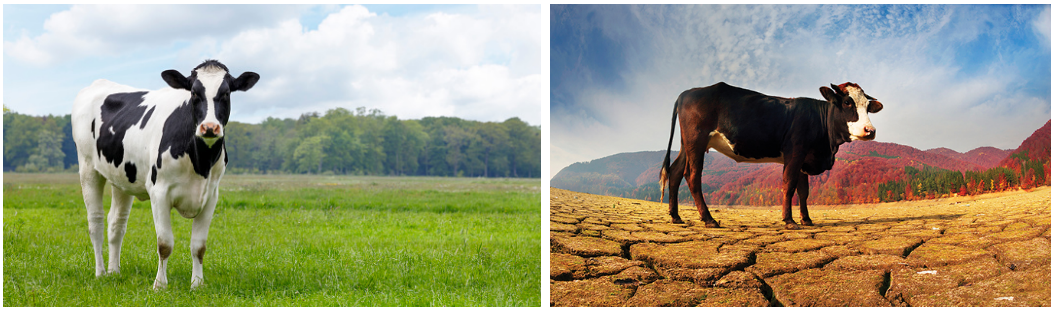


Figure 1: **Left:** example of instances in the training domain. **Right:** example of instances in the testing domain. Incorrect classification of instances in the testing domain could occur due to the spurious correlation of the scenery with the class label cow.

Multiple methods to tackle this problem such as Invariant Risk Minimization (IRM) (Arjovsky, Bottou, Gulrajani, & Lopez-Paz, 2019), Distributionally Robust neural networks (Sagawa, Wei Koh, Hashimoto, & Liang, 2019) and Risk Extrapolation (REx) (Krueger et al., 2020) have been proposed. The last one is the method for this research project. It selects a method that equalizes the training loss over multiple domains to encourage generalization. This paper will be about understanding how the method behaves in different situations, what its limits are, and how these affect the implementation in these situations.

The goal of this project is to understand the method of Risk Extrapolation. For this method, we will try to answer the following question: When is REx able to learn an invariant relationship in a synthetic dataset, and when does it fail to do so? In order to break this into smaller steps, these four sub-questions will be answered:

1. What are invariant relationships, and what do REx and similar methods do to find them?
2. What synthetic datasets that apply to the method of REx capture the essence of how the method performs in different distributional shifts?
3. What is the influence of the training domains and the number of samples used on the invariant prediction capabilities of REx?
4. How does the performance of REx compare to that of similar methods?

This paper contributes to the field of OOD generalization and Invariant Relationships by showing how REx performs on four different domain shifts that capture a wide variety of datasets found in the real world. This paper shows that we can reduce these real-world shifts to a more elemental form using y for the regression value of the instance, x_1 for the (Invariant) feature that we are trying to learn and x_2 for the (Spurious) feature that does not correlate to y (figure 2). The idea behind this is that in real-world datasets, it is not always clear what features are invariant and which are not. More specifically, we show that REx performs better in situations where it can better capture the model of the dataset, such as when the number of samples is high, and the difference in the variance of the training distributions is larger. We also show that REx, compared to Invariant Risk Minimization (Arjovsky et al., 2019) and Empirical Risk Minimization (Vapnik, 1999), asymptotically outperforms similar methods in situations where; the average variance of the training domain distributions is shifted; more complex distributional shifts occur.

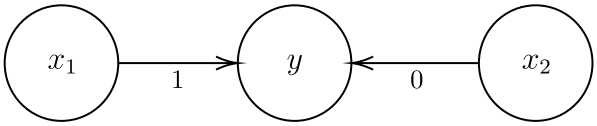


Figure 2: Model in which x_1 is the invariant relationship between the domains, x_2 is the spurious feature that differs between domains, and y is the regression value. The model represents the optimal situation where x_1 directly relates to the regression value y and x_2 does not.

In section 2, the background and relevant work will be explained in more detail. A formal problem description follows this in section 3 and the main investigation of the research in section 4. After this, the experiments and results will be discussed in section 5, followed by the research's ethics and reproducibility in section 6. In section 7, a summary of the insights will be given, and finally, a conclusion with possible future directions will be given in section 8.

2 Related Work

Out-of-domain generalization, invariant relationships and shifts have been the topic of many research papers in the past years. Nagarajan, Andreassen, and Neyshabur (2020) showed that spurious correlations in training data could influence the performance of Empirical Risk Minimization (ERM) so much that the algorithm fails. This even happens in situations that are easy to learn and where the model was expected to succeed. They also showed that when in the absence of these correlations, the algorithm performs as it should and does not fail.

Kamath, Tangella, Sutherland, and Srebro (2021) also showed that IRM, even with infinite training environments, does not always result in good OOD performance and that it sometimes even finds a worse predictor than unrestricted ERM. They also show that in some cases, IRM prefers an invariant predictor with a sub-optimal performance on OOD generalization.

Ahuja, Wang, Dhurandhar, Shanmugam, and Varshney (2020) introduced a framework for four common domain shifts: covariate, confounded, anti-causal, and hybrid shift. They then compare the performance of ERM and IRM on those shifts. This is done by testing the algorithms' performance relative to the number of samples per domain. They show that in the case of covariate shift, ERM and IRM get the same asymptotic solution (when samples become very large) and have similar performance for a finite amount of samples.

The paper about REx showed that REx outperforms IRM in the case of covariate shift. They do this using the Colored MNIST dataset (Arjovsky et al., 2019), a dataset with coloured (red/green) digits. They train the algorithm on a dataset where the distribution of the colours is different from that of the colour distribution in the test set.

This paper will try to address the gap between the papers mentioned above. More specifically, the paper will extend the work of Ahuja et al. (2020) and Kamath et al. (2021)

by comparing REx in a similar setting. The motivation behind this is that Krueger et al. (2020) compares REx to ERM and IRM. Next to these domain shifts, we will also show what the influence of the chosen training domains is on the invariant prediction capability of REx in an OOD setup.

3 Measuring invariance in domain shifts

The next subsections will formally define the problem the paper is trying to solve. In the first subsection, 3.1, a formal definition of the OOD problem will be given. In subsection 3.2, the principle of how you generate invariant relationships in a synthetic dataset will be discussed, which is followed by subsection 3.3, in which the shifts are formally defined, and the generation of the dataset is expanded with these shifts. In subsection 3.4, the metrics will be discussed.

3.1 Out-of-distribution generalization

To understand OOD generalization, we first define X as the input vector of a sample and Y as the regression value. We follow a similar structure Shen et al. (2021) used. We define a domain (or environment) as a group of samples with the same joint probability distribution $P(X, Y)$. Using this, we can now define the OOD generalization problem:

Definition 1 (OOD)**.** *Given an input vector X and regression value Y for a training and testing domain, out-of-distribution generalization is the setting of domain generalization where $P(X, Y)_{train} \neq P(X, Y)_{test}$ and where $P(X, Y)_{test}$ is unknown during the training time.*

Notice that $P(X, Y)_{train}$ can consist of multiple domains $\mathcal{D}$, each with other distributions $P(X, Y)$. In this paper, we draw the values of X and Y from a random standard distribution $\mathcal{N}(0, 1)$. We enforce definition 1 by multiplying the distribution of X by a scalar e for each domain. In essence, since the distribution is centred at a mean of 0, only the variance σ^2 of X changes. In this paper, we also aim for a harder OOD generalization problem where $e_{test} \geq e_{train}$ which enforces that the variance of X in the test probability is always larger than in the training domain(s). Using this, we can now define our harder OOD problem as:

Definition 2 (Hard OOD)**.** *Given the training domains $\mathcal{D}_{train}$ each with their variance σ^2_D and the testing domain with its variance σ^2_{test}, hard out-of-distribution generalization is the setting where $\forall D \in \mathcal{D}_{train}$, $\sigma^2_{test} \geq \sigma^2_D$ holds.*

In order to measure how much the training domains differ in their variance σ^2 we define the domain distance (definition 3). It represents the the most significant distance between the domains (i.e. the domains with the largest difference in distribution).

Definition 3 (Domain distance)**.** *Given the domains $\mathcal{D}$ with their domain specific scalars $\mathcal{E}$, the domain distance is $\max \mathcal{E} - \min \mathcal{E}$. I.e. for the domains with the maximal σ^2_{max} and minimal σ^2_{min} variance in their normal distributions.*

For this paper, we are interested in the influence of the distributional shift of X relative to the invariant prediction capabilities. If we would also multiply Y by our domain-specific scalar e, we would also introduce a distributional shift in Y. In this case, since we are testing the methods in a hard OOD (definition 2), we are making Y more difficult to predict (more significant errors for larger values of e). This is because distributions with a higher variance tend to have a higher error margin induced by the wider distribution spread. To isolate this problem, we assume homoscedasticity:

Assumption 1 (Homoscedasticity)**.** *Given the input vector X and regression value Y, we assume $\forall D \in \mathcal{D}_{train}$, $P(Y)_D = P(Y)_{test} \wedge P(X)_D \neq P(X)_{test}$.*

Notice that as $P(X)$ still differs between the domains, defintion 1 still holds.

3.2 Invariant relationships in a dataset

We now need to define how the synthetic dataset, described by figure 2, is generated. In the case of a dataset that contains both invariant features and features that do not correlate to the regression value Y, this results in equations 1-3. The covariate input vector $X = [X_1, X_2]$ of d dimensions is split in 2 so that both X_1, X_2, and Y have $d/2$ dimensions.

$$X_1^e = e \cdot \mathcal{N}(0,1) \tag{1}$$

$$X_2^e = e \cdot \mathcal{N}(0,1) \tag{2}$$

$$Y^e = \mathcal{N}(0,1) + X_1 \cdot W_{x_1 \to y} \tag{3}$$

Where e is the domain specific scalar, $\mathcal{N}(0,1)$ is a random standard distribution of size $d/2$, and $W_{x_1 \to y}$ represents the edge weight between variable X_1 and Y. For this paper we will aim for a 1 to 1 (assumption 2) correlation between the invariant features and the regression value (see figure 2). Where I represents the identity (or equivalently 1 if the feature is one-dimensional).

Assumption 2 (Invariance)**.** *Given the invariant features X_1, the weight $W_{x_1 \to y}$, and the regression value Y, we assume $W_{x_1 \to y} = I$.*

3.3 Shifts on datasets

Following the framework of Ahuja et al. (2020), we now categorize four domain shifts for regression. This paper differs from the framework in that we do not change the distribution of Y for each environment (assumption 1). Instead, we aim to analyze pure covariate shifts in these distributions by only changing the distributions of the input variables X_1 and X_2. It also differs in the way edge $W_{x_1 \to y}$ (assumption 2) and edge $W_{y \to x_1}$ (definition 6) are distributed, the paper uses values drawn from a normal distribution while we aim for a stronger correlation between variables X_1, X_2 and Y by setting the edge weights to identity I. In the definitions below, these shifts will shortly be discussed. The visualization of the enumerated shifts can be found in 3-6.

Definition 4 (Covariate shift)**.** *Given input feature $X = [X_1, X_2]$, covariate shift (figure 3) is a domain shift where the distribution $P(X)$ is multiplied by a scalar e to introduce a distributional shift.*

This is essentially what is written in equation 1-3 by multiplying the distribution by e. In this shift, X_2 is thus not correlated to the regression value Y. An example of covariate shift in the case of figure 1 would be the change in size of the cows. Where before larger cows did not occur, in the new domain they could.

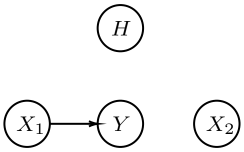


Figure 3:
Covariate (CS)

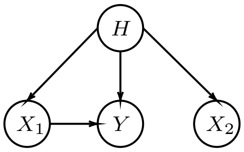


Figure 4:
Confounded (CF)

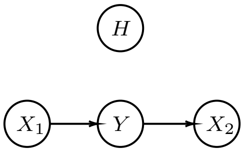


Figure 5:
Anti-causal (AC)

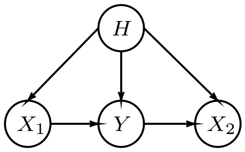


Figure 6:
Hybrid (HB)

Definition 5 (Confounded shift). *Given input feature $X = [X_1, X_2]$, the regression value Y, and a confounder H, confounded shift (figure 4) is a domain shift where X and Y are partially caused by a confounder H which is also by multiplied the domain specific scalar e. For which the weights $W_{h \to x_1}$, $W_{h \to x_2}$, and $W_{h \to y}$ are drawn from $\frac{1}{d}\mathcal{N}(0,1)$. Here d is the dimension of X and Y.*

In this case, the feature X_2 is indirectly connected to the regression value Y through their common confounder H. An example of a confounded shift in the case of figure 1 would be the change of brightness or lighting in the picture (i.e. pictures taken in a desert might be brighter due to the sun).

Definition 6 (Anti-causal shift). *Given input feature $X = [X_1, X_2]$, the regression value Y, anti-causal shift (figure 5) is a domain shift where X_2 is caused by Y with $W_{y \to x_2} = I$.*

X_2 in this case does not directly influence Y but is strongly correlated to it since $W_{y \to x_2}$ is equal to 1. An example of confounded shift in the case of figure 1 could be the grass length in the left figure. The fact that the animal on the picture is a cow (Y) directly influences the grass length (X_2).

Definition 7 (Hybrid shift). *Given confounded shift S_{CF} (definition 5) and anti-causal shift S_{AC} (definition 6), hybrid shift (figure 6) is a domain shift where $S_{CF} \wedge S_{AC}$.*

Notice that since we are still multiplying the scalar e with the distribution $P(X)$, we are adding the aforementioned (confounded, anti-causal and hybrid) shifts to covariate shift. We are thus using covariate shift as a baseline to see how introducing new correlations influences the performance.

Another way to look at this model is to set the edge weight $W_{a \to b}$ from one variable a to another variable b to zero 0 when such an edge does not exist and to non-zero when it does. The representation of the edge weights can then be defined as in table 1. Next to that, we can now extend the equations 1-3 to fit these four distributional shifts we get equations 4-7. When combining the generative equations with the edge weights in table 1 we can now generate a dataset for each shift in each domain e. Note that since we are trying to prove whether the method of REx finds invariant relationships, $W_{x_1 \to y}$ should always be equal to the identity (assumption 2).

$$H^e = e \cdot \mathcal{N}(0,1) \tag{4}$$

$$X_1^e = e \cdot \mathcal{N}(0,1) + H^e \cdot W_{h \to x_1} \tag{5}$$

$$Y^e = \mathcal{N}(0,1) + H^e \cdot W_{h \to y} + X_1^e \cdot W_{x_1 \to y} \tag{6}$$

$$X_2^e = e \cdot \mathcal{N}(0,1) + H^e \cdot W_{h \to x_2} + Y^e \cdot W_{y \to x_2} \tag{7}$$

	Covariate Shift	Confounded shift	Anti-causal shift	Hybrid shift
$W_{h\to x_1}$	0	$\frac{1}{d}\mathcal{N}(0,1)$	0	$\frac{1}{d}\mathcal{N}(0,1)$
$W_{h\to x_2}$	0	$\frac{1}{d}\mathcal{N}(0,1)$	0	$\frac{1}{d}\mathcal{N}(0,1)$
$W_{h\to y}$	0	$\frac{1}{d}\mathcal{N}(0,1)$	0	$\frac{1}{d}\mathcal{N}(0,1)$
$W_{y\to x_2}$	0	0	I	I
$W_{x_1\to y}$	I	I	I	I

Table 1: Edge weights of the distributional shifts.

3.4 Metrics

This paper will try to analyze the performance of the in section 3.3 mentioned elemental forms of shifts for REx. Within the focus on these shifts, this paper will analyze how the algorithms behave with varying training domains. The tested methods are trained on all domains except the last one, the test domain. The performance of the method is measured using the metric of model estimation error:

Definition 8 (Model estimation error)**.** *Given the models estimated edge weights $\hat{W} = [\hat{W}_{x_1\to y}, \hat{W}_{x_2\to y}]$ and the true model edge weights $W = [W_{x_1\to y}, W_{x_2\to y}]$, the model estimation error is defined as the mean of the squared distances $||\hat{W} - W||^2$.*

Where W is generated by equations 4-7. Since we assume $W_{x_1\to y} = I$ (assumption 2), and $W_{x_2\to y} = 0$ since these features should not correlate to the regression value, we are essentially comparing if the model converges to a $W = [1, 0]$. For each method and shift, this error is split into three; the causal ($\hat{W}_{x_1\to y}$), the non-causal or anti-causal ($\hat{W}_{x_2\to y}$) and the average of the two. By measuring the causal error, we can see if the model recognizes the invariant features, and by measuring the anti-causal error, we can see whether the methods ignore the possibly spurious features. Based on these results, the paper will try to solve the problem of if and when REx learns invariant relationships.

4 Invariant prediction capabilities of REx

This paper's contribution is thus to see how REx performs on the shifts as mentioned earlier in a simple synthetic dataset. This is done by comparing this performance to ERM and IRM in the same setting. Based on this, and the in section 3.4 mentioned metrics, the learning of invariant relationships by REx is analyzed. The methods for comparing the aforementioned shifts all use the same expected loss function for generalization. This function is called the risk function and is defined as:

$$R_e(\theta) = \int L(y, f_\theta(x))dP(x,y) \tag{8}$$

In this function one needs to find the $f_\theta(x)$ with the lowest loss for regression value y. This function takes as its input a value feature x. Here the θ represents the parameters we need to optimize. The loss function for regression (squared error) for parameter θ can then be defined as:

$$L(y, f_\theta(x)) = (y - f_\theta(x))^2 \tag{9}$$

In cases where the test domain distribution is unknown, a value for $P(x, y)$ in formula 8 should be approximated. The function that does this is called the empirical risk function. The function takes, for n samples, the mean of the sum of errors (or equivalently, the Mean Squared Error). When combining this function with equation 9 we get:

$$R_{emp}(\theta) = \frac{1}{n}\sum_{i=1}^{n} L_i(y, f_\theta(x)) = \frac{1}{n}\sum_{i=1}^{n}(y - f_\theta(x_i))^2 \tag{10}$$

Empirical Risk Minimization (Vapnik, 1999) minimizes the sum of all these empirical risks and thus tries to minimize the average loss of all instances in all training domains. It is the standard method used for learning problems and is therefore used as a baseline for the domain shifts under investigation. For regression, ERM minimizes the loss by minimizing equation 11. Notice that we use R_e for the risk in the environment, or equivalently domain, e. This does not mean we use equation 8 for calculating this risk, but that we use the empirical equation 10 for environment e in the range of m domains.

$$R_{ERM}(\theta) = \sum_{e=1}^{m} R_e(\theta) \tag{11}$$

Another method that uses this risk minimization is called Invariant Risk Minimization (Arjovsky et al., 2019). IRM aims to predict well by minimizing the risk while also finding an invariant predictor in the training domains. The regression function of IRMv1, a practical version of IRM, defined by Arjovsky et al. (2019) (p. 5) can be found in equation 12. In this function $w = 1$ is a fixed scalar (Assumption IRMv1), λ is used a regularizer with values between 0 and ∞, and $\bigtriangledown$ is the gradient that is used to measure the norm penalty of w for $R_e(w \cdot \theta)$. In theory, θ now becomes the only invariant predictor since w is fixed.

$$R_{IRM}(\theta) = \sum_{e=1}^{m} R_e(\theta) + \lambda \cdot || \bigtriangledown_{w|w=1.0} R_e(w \cdot \theta)||^2 \tag{12}$$

This paper is investigating the method of Risk Extrapolation (Krueger et al., 2020). The method aims to reduce the sum of training risks whilst also increasing the similarity of training risks. This might seem to contradict at first, but it effectively achieves its goal by trying to flatten out the risk plane. What is meant by this is that it may sometimes increase the training risk in the training domains such that the predictive trend line, the line drawn through points of the training risks, for unseen domains, looks more like a 'flat line'. An example of this can be found in figure 7. The idea behind this is that if the model has equivalent risk in the training domains, the risk in the test domain will also be equivalent. Of course, this is still a prediction, and since we do not know the distribution of the test domain, it could still be the case that this does not always result in a good performance. The method of REx has two versions; MM-REx and V-Rex. For this paper, the V-REx method will be used because it is simpler, more stable and more effective (Krueger et al., 2020). For regression, with Var being the function that calculates the variance of risks, the risk function is defined as:

$$R_{REx}(\theta) = \beta \cdot Var(\{R_1(\theta), R_2(\theta) \cdots R_m(\theta)\}) + \sum_{e=1}^{m} R_e(\theta) \tag{13}$$

REx is similar to IRM in that they both enforce that the best linear classifier should be equal across all training domains. IRM specifically aims for Invariant Prediction, this is

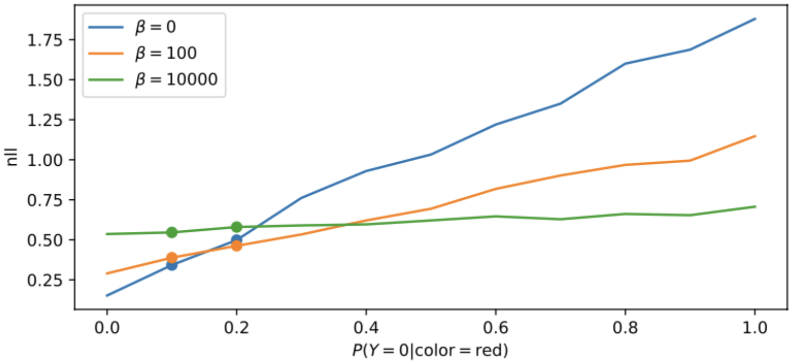


Figure 7: Flattening the risk plane by REx. The graph is plotted with training risks on the y-axis and the change in distribution or shift on the x-axis. The dots represent measurements in two different domains. The β in the figure legend is the V-REx penalty; this favours a flatter risk plane when increased. Figure copied from Krueger et al. (2020)(p. 3).

different from REx which aims to improve robustness for any sort of distributional/domain shifts (Krueger et al., 2020). Also notice that if β (or Var) is zero in equation 13 we get equation 11 and that if we set $|| \bigtriangledown_{w|w=1.0} R_e(w \cdot \theta)||^2$ or λ to zero in equation 12 we also get equation 11. In essence we are thus testing what the influence of $\sum_{e=1}^{m} R_e(\theta)$, and what influence of the regularizer of REx and IRM are on their invariant prediction capability.

5 Experiments

All the experiments are done with Python 3.9.12 and Pytorch 1.11.0. The code for these experiments can be found at: https://gitlab.com/hofland.jeroen/rex-distributional-shift. Each experiment is repeated $r = 10$ times. The results are the mean of these 10 measures. For each calculation, we also take the standard error, i.e. the standard deviation σ divided by the square root of the number of repetitions r. The input feature $X = [X_1, X_2]$ has a dimension $d = 10$. This means that there are 5 invariant features and 5 possibly spurious features. Each method is trained on the training domains $\mathcal{D}_{train} = [D_1, D_2, \cdots, D_m]$ each with their domain-specific distribution scalar $e_{train} = [e_1, e_2, \cdots, e_m]$ where m is the amount of training domains. All experiments use a similar domain structure with different parameters changed for each experiment. This is the same domain configuration Ahuja et al. (2020) used and is as follows: $e_{test} = [0.2, 2.0]$ and $e_{test} = 5.0$. The method of ERM is implemented using the linear regression method of sklearn. IRM and REx are implemented using 50.000 gradient descent steps with a learning rate of 0.001. They both use the training domain validation described by Gulrajani and Lopez-Paz (2020) for optimizing their regularizer. After training, REx and IRM select the best regularizer for the test domain D_{test} with domain scalar e_{test} and return their optimal model. This might not fully capture the concept of OOD generalization (definition 1) as the regularizer is optimized for the test domain, but it does give an insight into the optimal behaviour of the method in the test domain. IRM selects its penalty value from $[0, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1]$ and REx selects its β out of $[1, 10, 100, 1000, 10000, 100000]$. We perform multiple experiments involving the in section 3.3 mentioned shifts:

Experiment 1 (Sample complexity)**.** This experiment tests the performance of the methods relative to the number of samples n. The domains used for this experiment are equal to those described above. The results are displayed in figures 8a, 8d, 8g, and 8j.

REx performs well when the amount of samples increases. This was expected as the method learns more about the distribution of the samples when there are more samples to learn from. In the case of hybrid shift, the most complexly correlated model, the method performs better than ERM and IRM for larger samples. REx thus performs relatively better in situations with more (stronger) correlations between the input features and the regression value. This was expected as REx aims for general robustness to distributional shifts. The results also show that there is no significant difference between the causal and non-causal errors (see appendix A.1 and A.4). This implies that REx recognizes the invariant features and ignores the possibly spurious features.

Experiment 2 (Quantity of training domains)**.** This experiment tests the methods' performance relative to the number of training domains. The first two training domains and the test domain are the same as the sample complexity experiment. The domain scalars added after that are calculated based on the total domains and the two starter domains. We aim to increment the domain scalar between 0.2 and 2.0 linearly. For a total of $m = 7$ training domains, this results in an increment of 0.3 per domain. This is calculated in this way to increase the number of training domains without changing the domain distance (e_{min} and e_{max} stay the same). The training domain scalars then become $e_{train} = [0.2, 2.0, 0.2 + (0.3 \cdot 1), \cdots 0.2 + (0.3 \cdot (t - 2)]$ for the performance of the methods in t training domains. This means that when a method is tested against 4 training domains ($t = 4$) we get $e_{train} = [0.2, 2.0, 0.5, 0.8]$. The experiment is run on a dataset with $n = 1000$ samples per domain D. This amount of samples was chosen since the method's performance is relatively stable after $n = 1000$ in the sample complexity experiment (see figure 8 column 1). By doing so, we are trying to isolate the property (quantity of training domains) we want to test. The results are displayed in figures 8b, 8e, 8h, and 8k.

For the number of training domains, the method of REx performs better when more domains are added for covariate and confounded shift. This is also expected as this, just as an increment in the number of samples per domain, enables the method to better capture the concept of the model. For the anti-causal and hybrid shift, the method performs better than IRM and ERM for a larger number of training domains. For REx in the confounded shift, there is a slight difference between the causal and non-causal error (see appendix A.2). This visualization can be found in appendix A.4.

Experiment 3 (Domain distance)**.** This experiment tests the methods' performance relative to the domain distance between e_1 and e_2. The domain distance ranges from 0.0 to 1.8 with steps of 0.2. e_1 is defined as 0.2. The training domains can then be defined as $e_{train} = [0.2, 0.2 + distance]$. The test domain e_{test}, just as in the previous experiments, is equal to 5.0. The experiment is ran on a dataset with $n = 1000$ samples per domain D. The results are displayed in figures 8c, 8f, 8i, and 8l.

REx performs better when the domain distance increases. Especially for anti-causal and hybrid shift, where it performs better for more considerable distances and has a steeper learning curve than IRM. Just as in the sample complexity, there is no significant difference between the causal and non-causal errors for the method of REx (see appendix A.3 for details). There is, however, a difference in errors for IRM in this experiment which can be seen in appendix A.4.

Notice that if $t = 2$ in experiment 2 and $distance = 1.8$ in experiment 3 the domains are equal to that of experiment 1, and that in these cases REx has an equivalent error in all distributional shifts (table 2). It also shows the out-performance of REx in the hybrid distributional shift.

	Covariate Shift	Confounded shift	Anti-causal shift	Hybrid shift
REx	0.0006±0.0001	0.0010±0.0002	0.0065±0.0013	0.0051±0.0006
IRM	0.0006±0.0001	0.0010±0.0002	0.0011±0.0002	0.0073±0.0014
ERM	0.0001±0.0000	0.0008±0.0002	0.0089±0.0002	0.0142±0.0012

Table 2: Average model estimation error for REx, IRM, and ERM with $n = 2000$ samples.

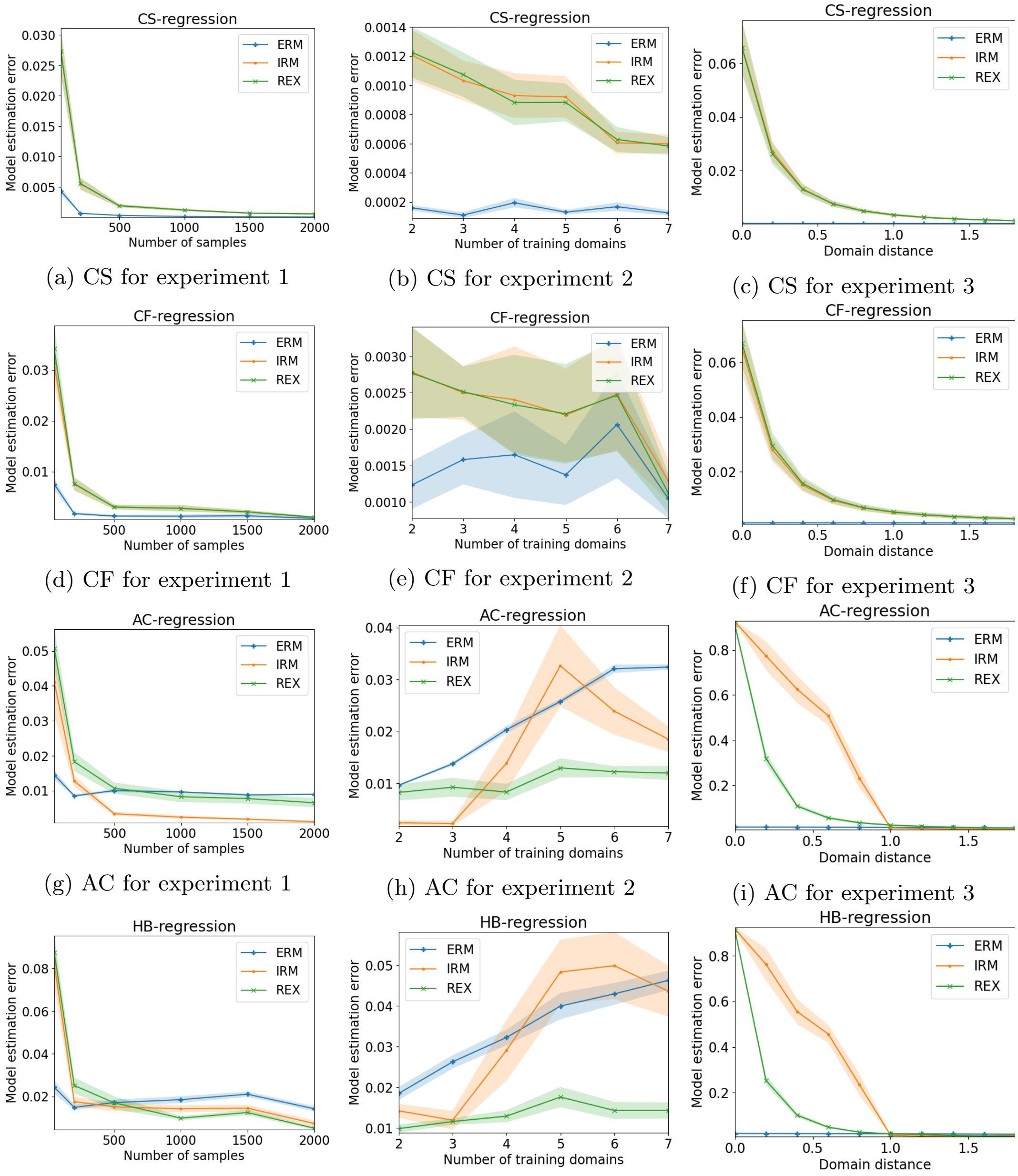
6 Responsible Research

This paper uses publicly available code to generate its results. This means that anyone can repeat the experiment and get approximately the same results. By doing so, this paper aims to be robust against manipulation, fabrication, and sloppiness. The experiments have all been done with 10 repetitions with a mean and standard error to show these experiments are consistent in their behaviour, conform to the industry standard, and are reproducible. All data and experiments performed have been included either in the paper itself or in the appendix to show that the data was not manipulated or trimmed for more favourable results. In an attempt to rule out mistakes made by human error, all data used in the experiments (both tables and figures) have been directly generated by Python into LaTeX-code. This paper only uses generated synthetic datasets with no affiliation with external people or companies. By doing so, it tries to remove any ethical aspects from the equation. For sources or ideas used in other researchers/papers, a reference or quote has always been provided to clarify that these ideas are not of our own or adapted.

7 Discussion

Smalller samples. Based on the results, we can conclude that in a simple synthetic dataset with 10 features, REx performs worse for invariant prediction when the number of samples is low. For a smaller number of samples, the risk $R(\theta)$ is based on a regression function f_θ that cannot capture the actual distribution and correct correlation between features of input X and regression value Y. When REx increases the similarity of the risks in the best performing domain, it is, depending on the scale of penalty β, reinforcing this uncertain correlation of features. It is, in essence, taking a gamble on risk prediction in the test domain. Note that REx can not choose a β penalty of 0 to turn off this increasing of similarity. Since ERM is just REX with a β penalty of 0 it is not doing this reinforcing which results in better invariant prediction capability for the smaller number of samples. To get a clearer picture of how the equalizing of similarity works for small samples, the β regularizer could be researched for smaller steps of samples in the future, especially since the most significant drop in error is between 50 and 200 samples.

Complexity of the distributional shift. When the number of samples increases, the performance of REx, and equivalently IRM, reaches the same accurate asymptotic solution



(a) CS for experiment 1

(b) CS for experiment 2

(c) CS for experiment 3

(d) CF for experiment 1

(e) CF for experiment 2

(f) CF for experiment 3

(g) AC for experiment 1

(h) AC for experiment 2

(i) AC for experiment 3

(j) HB for experiment 1

(k) HB for experiment 2

(l) HB for experiment 3

Figure 8: Performance of REx, IRM and ERM for covariate shift (CS), confounded shift (CF), anti-causal shift (AC), and hybrid shift (HB). The methods are plotted against the average model estimation error on the y-axis, with the spread representing the standard error. **Left:** Sample complexity (experiment 1) with the number of samples on the x-axis. **Center:** Quantity of training domain (experiment 2) with the number of training domains on the x-axis. **Right:** Domain distance (experiment 3) with domain distance on x-axis.

as ERM in the case of covariate and confounded shift. REx in these situations does not perform much better than the other methods because the model of the dataset is relatively simple. Since we assume homoscedasticity and $W_{x_1 \to y} = I$ (assumption 2), the correlation between the feature X_1 and Y is constant across the domains. Recognizing which features in X cause Y is not difficult as X only has 10 features. Minimizing the risk $R(\theta)$ for these features then is enough. When adding confounded shift, the confounder H will also partly cause Y. Since H and X are correlated through domain scalar e, stimulating similarity to find invariant relationships decreases the difference between REx and ERM (see table 2). When replacing the confounded shift with the anti-causal one, we see a slight increment in the error due to the stronger correlation between X_2 and Y with edge $W_{y \to x_2} = I$. Y causing X_2 here can be observed as X_2 causing Y if no counterexample is given. IRM performs better in this situation since it aims to predict which features are in X_1 and X_2 and does not try to optimize its robustness for this distributional shift of Y causing X_2. The opposite can be seen in the hybrid shift. Krueger et al. (2020) showed that "Broadly speaking, REx optimizes for robustness to the forms of distributional shift that have been observed to have the largest impact on performance in training domains. This can be a significant advantage over the more focused (but also limited) robustness that IRM targets." (p. 2). REx here thus optimizes for any sort of shift that is causing the largest performance drop (Anti-Causal). This is in contrast with IRM, which is still aiming for invariant prediction.

Domain distance. Based on the results of the domain distance experiment, we can conclude that the performance of REx is optimal when the domain distance increases. We assume that this is caused by the increased similarity of the distributions of D_2 and D_{test} and increased dissimilarity between the distributions D_1 and D_2. By increasing the dissimilarities, the domain shifts are magnified and easier to detect when compared to the similarity of the risk. By increasing the similarities of the distributions D_2 and D_{test}, we are solving a problem that is more similar to the distribution on which the edge weights are based. It would be interesting to see if the performance changes when we change the distance between X_2 and X_{test} relative to the distance of X_1 and X_2. By doing so, we can isolate if the similarity to the test distribution is the largest influence or if the domain distance is.

Quantity of training domains. For all domain shifts, the performance of REx stays relatively the same[1] when adding more domains between $e_1 = 0.2$ and $e_2 = 2.0$. Distributions that cause the domain distance thus have the most influence on equalizing the risk since the smallest risk is increased until it looks more like the highest risk. When new domains are introduced with a distribution that lies within the already known distributions, the risks will be close to the line (figure 7) drawn through the domains that cause the domain distance. REx now does not need to increase the similarity since they are already similar and on the predictive trend line. We conclude that the first part of equation 13 (equalizing risk) is mostly based on the domains with the most significant domain distance and that, when adding more domains that do not change the domain distance, the performance does not drastically change. This is in contrast to IRM and ERM, which perform worse when the average distribution variance is shifted. To extend this experiment, the addition of training domains outside the domains that cause the domain distance could be researched.

Due to computation power constraints, the experiments are performed on 10-dimensional features to reduce the number of iterations required to achieve a satisfactory result. For the same reason, the number of repetitions $n = 10$ was chosen instead of higher repetitions.

[1] In the case of covariate and confounded shift, there is an increment in error, but since the difference is so small, we consider this as equal.

8 Conclusions

This paper has presented a synthetic dataset with four different domain shifts containing invariant and spurious features. This dataset was tested on the sample complexity, quantity of training domains, and domain distance to answer the question: When is REx able to learn an invariant relationship in a synthetic dataset, and when does it fail to do so?

This was done by first explaining invariant relationships and how they can be modeled using the generative equations 1-3. Based on this model, four datasets were created that implement either Covariate, Confounded, Anti-Causal or Hybrid shift. We showed that REx's absolute performance (smaller model estimation error) is better when covariate and confounded shift occurs. In the situations where anti-causal and hybrid shifts occur, REx performs better with respect to IRM and ERM. The paper also showed that for REx to find invariant relationships, the number of samples and domain distance should be large. In which the latter is the most significant one. The quantity of training domains experiment showed that the performance of REx is based on the two domains with the most significant domain distance. It showed that adding more domains that do not increase this distance does not change the invariant prediction capabilities of REx. This contrasts with the methods of IRM and ERM, whose performance decreases when these domains are added to the anti-causal and hybrid shifts.

In the future, it would be interesting to see how the performance of these methods and experiments differs when we in- or decrease the number of dimensions of X. We expect that REx will out-perform in the task of invariant prediction relative to ERM and IRM, even more so than in our experiments since the model becomes more complex when more dimensions are added. Another thing to look into would be to see how the performance changes when we do not determine the regularizers of REx and IRM on the testing domain but on the training domains. It could also be interesting to see how the performance of IRM and REx relate to each other in a heteroscedastic setting (i.e. where some domains are harder than others) with these domain shifts as IRM should out-perform REx in these situations (Krueger et al., 2020).

References

Ahuja, K., Wang, J., Dhurandhar, A., Shanmugam, K., & Varshney, K. R. (2020, October). Empirical or Invariant Risk Minimization? A Sample Complexity Perspective. *arXiv e-prints*, arXiv:2010.16412.

Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2019, July). Invariant Risk Minimization. *arXiv e-prints*, arXiv:1907.02893.

Beery, S., van Horn, G., & Perona, P. (2018, July). Recognition in Terra Incognita. *arXiv e-prints*, arXiv:1807.04975.

Gulrajani, I., & Lopez-Paz, D. (2020, July). In Search of Lost Domain Generalization. *arXiv e-prints*, arXiv:2007.01434.

Kamath, P., Tangella, A., Sutherland, D. J., & Srebro, N. (2021, January). Does Invariant Risk Minimization Capture Invariance? *arXiv e-prints*, arXiv:2101.01134.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., ... Courville, A. (2020, March). Out-of-Distribution Generalization via Risk Extrapolation (REx). *arXiv e-prints*, arXiv:2003.00688.

Nagarajan, V., Andreassen, A., & Neyshabur, B. (2020, October). Understanding the Failure Modes of Out-of-Distribution Generalization. *arXiv e-prints*, arXiv:2010.15775.

Sagawa, S., Wei Koh, P., Hashimoto, T. B., & Liang, P. (2019, November). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv:1911.08731v2*.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., & Cui, P. (2021, August). Towards Out-Of-Distribution Generalization: A Survey. *arXiv e-prints*, arXiv:2108.13624.

Vapnik, V. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988-999. doi: 10.1109/72.788640

Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., ... Yu, P. S. (2021, March). Generalizing to Unseen Domains: A Survey on Domain Generalization. *arXiv e-prints*, arXiv:2103.03097.

A Appendix

A.1 Sample complexity results

Method	Number of samples	Average	Causal	Non-causal
ERM	50	0.0043±0.0006	0.0044±0.0009	0.0043±0.0009
IRM	50	0.0272±0.0038	0.0272±0.0052	0.0273±0.0059
REX	50	0.0273±0.0038	0.0295±0.0049	0.0252±0.0058
ERM	200	0.0007±0.0001	0.0007±0.0001	0.0007±0.0001
IRM	200	0.0055±0.0010	0.0051±0.0016	0.0059±0.0012
REX	200	0.0056±0.0010	0.0050±0.0016	0.0061±0.0013
ERM	500	0.0003±0.0000	0.0003±0.0001	0.0003±0.0001
IRM	500	0.0019±0.0003	0.0020±0.0003	0.0018±0.0005
REX	500	0.0019±0.0003	0.0020±0.0002	0.0018±0.0005
ERM	1000	0.0002±0.0000	0.0001±0.0000	0.0002±0.0000
IRM	1000	0.0012±0.0002	0.0009±0.0002	0.0015±0.0003
REX	1000	0.0012±0.0002	0.0010±0.0002	0.0015±0.0003
ERM	1500	0.0001±0.0000	0.0001±0.0000	0.0001±0.0000
IRM	1500	0.0007±0.0001	0.0008±0.0002	0.0007±0.0001
REX	1500	0.0007±0.0001	0.0008±0.0002	0.0007±0.0001
ERM	2000	0.0001±0.0000	0.0001±0.0000	0.0001±0.0000
IRM	2000	0.0006±0.0001	0.0005±0.0001	0.0006±0.0001
REX	2000	0.0006±0.0001	0.0006±0.0001	0.0006±0.0001

Table 3: Sample complexity of covariate shift with the errors displayed being the model estimation error ± standard error.

Method	Number of samples	Average	Causal	Non-causal
ERM	50	0.0075±0.0011	0.0076±0.0013	0.0075±0.0019
IRM	50	0.0313±0.0037	0.0382±0.0064	0.0244±0.0027
REX	50	0.0342±0.0045	0.0370±0.0066	0.0313±0.0064
ERM	200	0.0017±0.0003	0.0013±0.0004	0.0022±0.0003
IRM	200	0.0075±0.0013	0.0079±0.0025	0.0072±0.0011
REX	200	0.0076±0.0013	0.0077±0.0023	0.0075±0.0012
ERM	500	0.0013±0.0002	0.0007±0.0001	0.0018±0.0004
IRM	500	0.0030±0.0004	0.0028±0.0005	0.0032±0.0007
REX	500	0.0030±0.0005	0.0026±0.0006	0.0035±0.0008
ERM	1000	0.0012±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	1000	0.0028±0.0006	0.0017±0.0005	0.0039±0.0011
REX	1000	0.0028±0.0006	0.0016±0.0005	0.0039±0.0011
ERM	1500	0.0013±0.0004	0.0002±0.0000	0.0025±0.0007
IRM	1500	0.0021±0.0003	0.0012±0.0002	0.0030±0.0006
REX	1500	0.0021±0.0004	0.0012±0.0002	0.0030±0.0006
ERM	2000	0.0008±0.0002	0.0002±0.0000	0.0014±0.0003
IRM	2000	0.0010±0.0002	0.0005±0.0000	0.0015±0.0003
REX	2000	0.0010±0.0002	0.0005±0.0000	0.0016±0.0003

Table 4: Sample complexity of confounded shift with the errors displayed being the model estimation error ± standard error.

Method	Number of samples	Average	Causal	Non-causal
ERM	50	0.0145±0.0014	0.0157±0.0025	0.0133±0.0014
IRM	50	0.0411±0.0096	0.0546±0.0182	0.0276±0.0043
REX	50	0.0507±0.0055	0.0503±0.0076	0.0511±0.0083
ERM	200	0.0085±0.0003	0.0085±0.0005	0.0085±0.0005
IRM	200	0.0127±0.0016	0.0145±0.0030	0.0110±0.0013
REX	200	0.0183±0.0027	0.0193±0.0037	0.0172±0.0041
ERM	500	0.0100±0.0005	0.0105±0.0008	0.0095±0.0007
IRM	500	0.0034±0.0006	0.0039±0.0009	0.0028±0.0008
REX	500	0.0107±0.0016	0.0115±0.0025	0.0099±0.0021
ERM	1000	0.0096±0.0002	0.0096±0.0004	0.0096±0.0002
IRM	1000	0.0024±0.0004	0.0029±0.0008	0.0019±0.0003
REX	1000	0.0082±0.0015	0.0084±0.0023	0.0081±0.0022
ERM	1500	0.0088±0.0002	0.0091±0.0003	0.0085±0.0003
IRM	1500	0.0018±0.0003	0.0025±0.0004	0.0011±0.0002
REX	1500	0.0077±0.0014	0.0085±0.0020	0.0069±0.0021
ERM	2000	0.0089±0.0002	0.0089±0.0002	0.0090±0.0003
IRM	2000	0.0011±0.0002	0.0014±0.0004	0.0008±0.0002
REX	2000	0.0065±0.0013	0.0068±0.0020	0.0063±0.0018

Table 5: Sample complexity of anti-causal shift with the errors displayed being the model estimation error ± standard error.

Method	Number of samples	Average	Causal	Non-causal
ERM	50	0.0243±0.0031	0.0262±0.0050	0.0225±0.0038
IRM	50	0.0822±0.0109	0.1021±0.0165	0.0623±0.0118
REX	50	0.0874±0.0079	0.1032±0.0097	0.0716±0.0107
ERM	200	0.0149±0.0012	0.0155±0.0022	0.0143±0.0013
IRM	200	0.0176±0.0022	0.0209±0.0033	0.0143±0.0028
REX	200	0.0251±0.0037	0.0294±0.0050	0.0209±0.0052
ERM	500	0.0171±0.0013	0.0177±0.0022	0.0166±0.0016
IRM	500	0.0151±0.0020	0.0158±0.0030	0.0145±0.0028
REX	500	0.0170±0.0031	0.0192±0.0050	0.0148±0.0037
ERM	1000	0.0185±0.0016	0.0183±0.0023	0.0187±0.0024
IRM	1000	0.0142±0.0017	0.0137±0.0025	0.0148±0.0025
REX	1000	0.0098±0.0010	0.0100±0.0014	0.0096±0.0015
ERM	1500	0.0210±0.0012	0.0213±0.0017	0.0207±0.0018
IRM	1500	0.0145±0.0015	0.0143±0.0020	0.0148±0.0023
REX	1500	0.0125±0.0013	0.0135±0.0021	0.0114±0.0015
ERM	2000	0.0142±0.0012	0.0141±0.0019	0.0144±0.0017
IRM	2000	0.0073±0.0014	0.0067±0.0018	0.0079±0.0023
REX	2000	0.0051±0.0006	0.0053±0.0009	0.0050±0.0010

Table 6: Sample complexity of hybrid shift with the errors displayed being the model estimation error ± standard error.

A.2 Quantity of training domains results

Method	Number of training domains	Average	Causal	Non-causal
ERM	2	0.0002±0.0000	0.0001±0.0000	0.0002±0.0000
IRM	2	0.0012±0.0002	0.0009±0.0002	0.0015±0.0003
REX	2	0.0012±0.0002	0.0010±0.0002	0.0015±0.0003
ERM	3	0.0001±0.0000	0.0001±0.0000	0.0001±0.0000
IRM	3	0.0010±0.0001	0.0008±0.0002	0.0013±0.0002
REX	3	0.0011±0.0002	0.0009±0.0002	0.0013±0.0003
ERM	4	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	4	0.0009±0.0002	0.0009±0.0001	0.0010±0.0003
REX	4	0.0009±0.0002	0.0008±0.0001	0.0010±0.0003
ERM	5	0.0001±0.0000	0.0001±0.0000	0.0002±0.0000
IRM	5	0.0009±0.0001	0.0008±0.0001	0.0011±0.0003
REX	5	0.0009±0.0001	0.0007±0.0001	0.0011±0.0002
ERM	6	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	6	0.0006±0.0001	0.0005±0.0001	0.0007±0.0001
REX	6	0.0006±0.0001	0.0005±0.0001	0.0008±0.0001
ERM	7	0.0001±0.0000	0.0001±0.0000	0.0002±0.0000
IRM	7	0.0006±0.0001	0.0005±0.0000	0.0007±0.0001
REX	7	0.0006±0.0001	0.0005±0.0000	0.0007±0.0001

Table 7: Quantity of training domains for covariate shift with the errors displayed being the model estimation error ± standard error.

Method	Number of training domains	Average	Causal	Non-causal
ERM	2	0.0012±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	2	0.0028±0.0006	0.0017±0.0005	0.0039±0.0011
REX	2	0.0028±0.0006	0.0016±0.0005	0.0039±0.0011
ERM	3	0.0016±0.0003	0.0004±0.0001	0.0027±0.0004
IRM	3	0.0025±0.0004	0.0014±0.0002	0.0036±0.0005
REX	3	0.0025±0.0003	0.0014±0.0002	0.0036±0.0005
ERM	4	0.0016±0.0006	0.0004±0.0001	0.0029±0.0011
IRM	4	0.0024±0.0007	0.0011±0.0002	0.0037±0.0013
REX	4	0.0023±0.0007	0.0011±0.0002	0.0036±0.0013
ERM	5	0.0014±0.0004	0.0004±0.0001	0.0023±0.0007
IRM	5	0.0022±0.0006	0.0010±0.0002	0.0034±0.0012
REX	5	0.0022±0.0007	0.0010±0.0002	0.0035±0.0013
ERM	6	0.0021±0.0007	0.0003±0.0001	0.0038±0.0013
IRM	6	0.0025±0.0008	0.0009±0.0001	0.0041±0.0014
REX	6	0.0025±0.0008	0.0007±0.0001	0.0042±0.0013
ERM	7	0.0010±0.0003	0.0003±0.0000	0.0018±0.0004
IRM	7	0.0013±0.0003	0.0005±0.0001	0.0021±0.0004
REX	7	0.0011±0.0003	0.0004±0.0001	0.0019±0.0004

Table 8: Quantity of training domains for confounded shift with the errors displayed being the model estimation error ± standard error.

Method	Number of training domains	Average	Causal	Non-causal
ERM	2	0.0096±0.0002	0.0096±0.0004	0.0096±0.0002
IRM	2	0.0024±0.0004	0.0029±0.0008	0.0019±0.0003
REX	2	0.0082±0.0015	0.0084±0.0023	0.0081±0.0022
ERM	3	0.0137±0.0003	0.0135±0.0005	0.0139±0.0004
IRM	3	0.0022±0.0005	0.0026±0.0007	0.0019±0.0006
REX	3	0.0092±0.0018	0.0092±0.0026	0.0093±0.0027
ERM	4	0.0203±0.0007	0.0201±0.0012	0.0205±0.0009
IRM	4	0.0138±0.0053	0.0162±0.0097	0.0115±0.0049
REX	4	0.0083±0.0016	0.0078±0.0022	0.0088±0.0024
ERM	5	0.0257±0.0005	0.0255±0.0008	0.0259±0.0007
IRM	5	0.0326±0.0078	0.0344±0.0126	0.0308±0.0100
REX	5	0.0130±0.0019	0.0141±0.0030	0.0118±0.0023
ERM	6	0.0321±0.0008	0.0324±0.0014	0.0317±0.0009
IRM	6	0.0239±0.0045	0.0236±0.0063	0.0242±0.0069
REX	6	0.0122±0.0011	0.0122±0.0019	0.0123±0.0012
ERM	7	0.0324±0.0005	0.0322±0.0009	0.0325±0.0006
IRM	7	0.0185±0.0025	0.0192±0.0041	0.0178±0.0030
REX	7	0.0120±0.0013	0.0127±0.0023	0.0113±0.0015

Table 9: Quantity of training domains for anti-causal shift with the errors displayed being the model estimation error ± standard error.

Method	Number of training domains	Average	Causal	Non-causal
ERM	2	0.0185±0.0016	0.0183±0.0023	0.0187±0.0024
IRM	2	0.0142±0.0017	0.0137±0.0025	0.0148±0.0025
REX	2	0.0098±0.0010	0.0100±0.0014	0.0096±0.0015
ERM	3	0.0263±0.0016	0.0262±0.0024	0.0263±0.0023
IRM	3	0.0119±0.0023	0.0133±0.0037	0.0105±0.0028
REX	3	0.0116±0.0008	0.0127±0.0011	0.0104±0.0010
ERM	4	0.0323±0.0019	0.0326±0.0027	0.0319±0.0029
IRM	4	0.0291±0.0075	0.0325±0.0121	0.0257±0.0092
REX	4	0.0129±0.0015	0.0142±0.0022	0.0116±0.0021
ERM	5	0.0400±0.0032	0.0409±0.0046	0.0390±0.0048
IRM	5	0.0484±0.0079	0.0543±0.0129	0.0424±0.0096
REX	5	0.0176±0.0025	0.0192±0.0040	0.0160±0.0031
ERM	6	0.0430±0.0027	0.0441±0.0043	0.0419±0.0033
IRM	6	0.0499±0.0083	0.0582±0.0136	0.0417±0.0094
REX	6	0.0143±0.0021	0.0161±0.0036	0.0125±0.0023
ERM	7	0.0463±0.0024	0.0472±0.0033	0.0454±0.0036
IRM	7	0.0437±0.0062	0.0451±0.0100	0.0422±0.0081
REX	7	0.0143±0.0020	0.0154±0.0032	0.0132±0.0025

Table 10: Quantity of training domains for hybrid shift with the errors displayed being the model estimation error ± standard error.

A.3 Domain distance results

Method	Domain distance	Average	Causal	Non-causal
ERM	0	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	0	0.0661±0.0099	0.0695±0.0130	0.0627±0.0155
REX	0	0.0658±0.0100	0.0675±0.0131	0.0641±0.0157
ERM	0.2	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	0.2	0.0271±0.0038	0.0264±0.0050	0.0279±0.0059
REX	0.2	0.0262±0.0040	0.0248±0.0056	0.0275±0.0059
ERM	0.4	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	0.4	0.0131±0.0018	0.0118±0.0024	0.0144±0.0029
REX	0.4	0.0129±0.0019	0.0115±0.0025	0.0144±0.0028
ERM	0.6	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	0.6	0.0075±0.0011	0.0066±0.0014	0.0084±0.0017
REX	0.6	0.0075±0.0011	0.0064±0.0014	0.0086±0.0017
ERM	0.8	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	0.8	0.0048±0.0007	0.0039±0.0007	0.0058±0.0012
REX	0.8	0.0049±0.0007	0.0041±0.0009	0.0056±0.0011
ERM	1	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	1	0.0033±0.0005	0.0027±0.0005	0.0040±0.0008
REX	1	0.0034±0.0005	0.0028±0.0006	0.0040±0.0007
ERM	1.2	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	1.2	0.0024±0.0004	0.0020±0.0004	0.0029±0.0006
REX	1.2	0.0025±0.0004	0.0020±0.0004	0.0030±0.0006
ERM	1.4	0.0002±0.0000	0.0002±0.0000	0.0002±0.0000
IRM	1.4	0.0019±0.0003	0.0015±0.0003	0.0023±0.0004
REX	1.4	0.0019±0.0003	0.0015±0.0003	0.0023±0.0004
ERM	1.6	0.0002±0.0000	0.0001±0.0000	0.0002±0.0000
IRM	1.6	0.0015±0.0002	0.0011±0.0002	0.0018±0.0004
REX	1.6	0.0015±0.0002	0.0012±0.0002	0.0018±0.0003
ERM	1.8	0.0002±0.0000	0.0001±0.0000	0.0002±0.0000
IRM	1.8	0.0012±0.0002	0.0009±0.0002	0.0015±0.0003
REX	1.8	0.0012±0.0002	0.0010±0.0002	0.0015±0.0003

Table 11: Domain distance for covariate shift with the errors displayed being the model estimation error ± standard error.

Method	Domain distance	Average	Causal	Non-causal
ERM	0	0.0013±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	0	0.0651±0.0095	0.0727±0.0166	0.0574±0.0098
REX	0	0.0669±0.0084	0.0751±0.0136	0.0587±0.0097
ERM	0.2	0.0013±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	0.2	0.0280±0.0044	0.0287±0.0073	0.0274±0.0054
REX	0.2	0.0295±0.0047	0.0320±0.0080	0.0270±0.0051
ERM	0.4	0.0013±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	0.4	0.0150±0.0023	0.0145±0.0035	0.0155±0.0032
REX	0.4	0.0157±0.0026	0.0160±0.0042	0.0154±0.0032
ERM	0.6	0.0013±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	0.6	0.0095±0.0015	0.0088±0.0021	0.0102±0.0022
REX	0.6	0.0098±0.0016	0.0095±0.0025	0.0101±0.0021
ERM	0.8	0.0013±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	0.8	0.0067±0.0011	0.0057±0.0014	0.0077±0.0017
REX	0.8	0.0068±0.0012	0.0059±0.0015	0.0077±0.0018
ERM	1	0.0013±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	1	0.0051±0.0009	0.0041±0.0010	0.0061±0.0015
REX	1	0.0052±0.0009	0.0041±0.0011	0.0063±0.0015
ERM	1.2	0.0012±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	1.2	0.0042±0.0008	0.0032±0.0009	0.0052±0.0013
REX	1.2	0.0042±0.0008	0.0031±0.0009	0.0054±0.0013
ERM	1.4	0.0012±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	1.4	0.0036±0.0007	0.0023±0.0007	0.0048±0.0013
REX	1.4	0.0036±0.0007	0.0024±0.0007	0.0047±0.0012
ERM	1.6	0.0012±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	1.6	0.0031±0.0007	0.0019±0.0005	0.0043±0.0012
REX	1.6	0.0031±0.0007	0.0020±0.0006	0.0043±0.0011
ERM	1.8	0.0012±0.0003	0.0004±0.0001	0.0021±0.0005
IRM	1.8	0.0028±0.0006	0.0017±0.0005	0.0039±0.0011
REX	1.8	0.0028±0.0006	0.0016±0.0005	0.0039±0.0011

Table 12: Domain distance for confounded shift with the errors displayed being the model estimation error ± standard error.

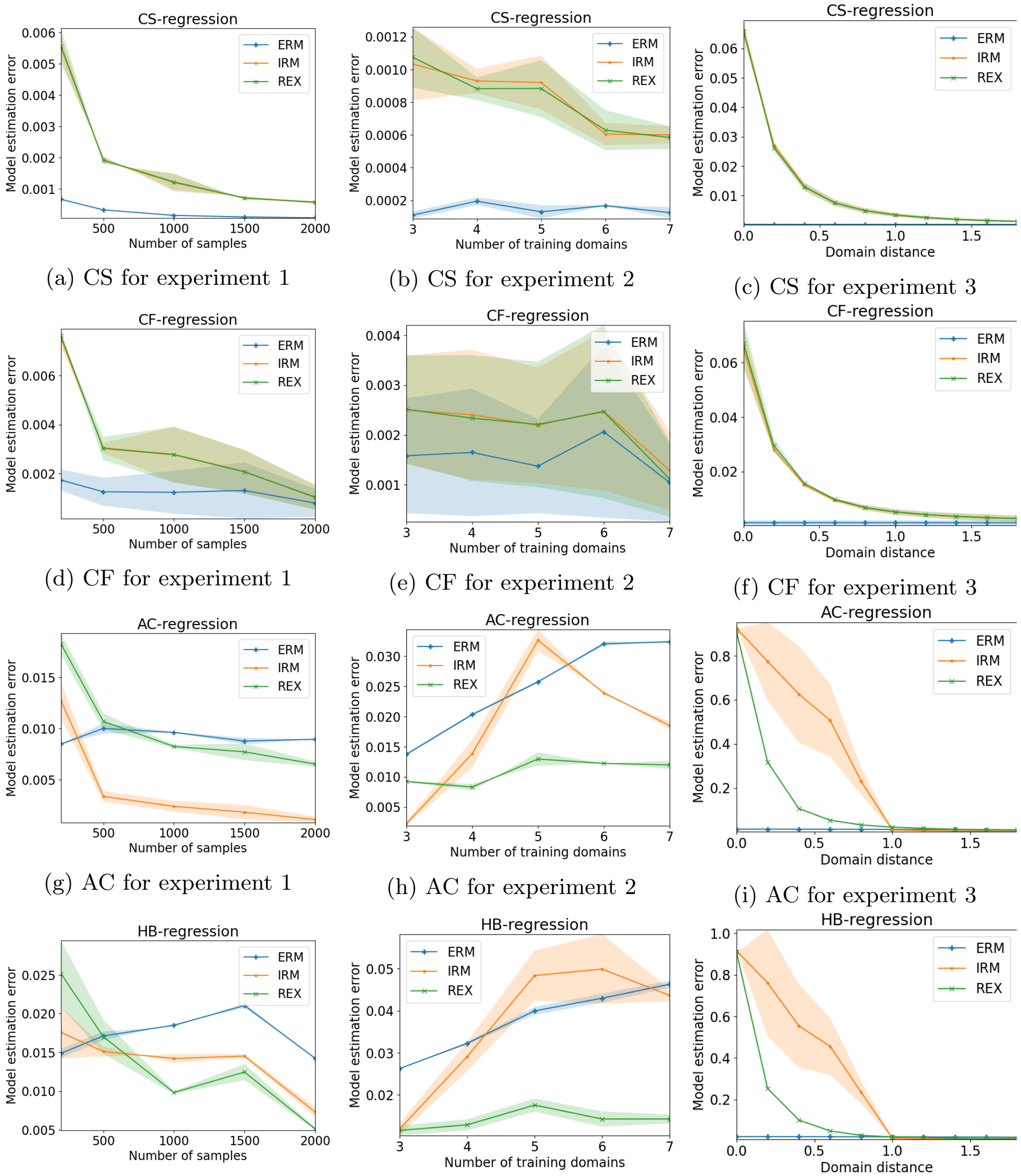
Method	Domain distance	Average	Causal	Non-causal
ERM	0	0.0127±0.0003	0.0128±0.0007	0.0125±0.0003
IRM	0	0.9253±0.0042	0.9244±0.0082	0.9263±0.0026
REX	0	0.9113±0.0108	0.9000±0.0213	0.9227±0.0037
ERM	0.2	0.0126±0.0003	0.0127±0.0006	0.0124±0.0003
IRM	0.2	0.7743±0.0633	0.9506±0.0522	0.5979±0.0852
REX	0.2	0.3173±0.0256	0.3196±0.0345	0.3151±0.0396
ERM	0.4	0.0124±0.0003	0.0125±0.0006	0.0123±0.0003
IRM	0.4	0.6254±0.0598	0.8445±0.0451	0.4063±0.0489
REX	0.4	0.1062±0.0102	0.1047±0.0149	0.1078±0.0148
ERM	0.6	0.0121±0.0003	0.0122±0.0006	0.0120±0.0003
IRM	0.6	0.5075±0.0414	0.6719±0.0168	0.3431±0.0309
REX	0.6	0.0533±0.0071	0.0530±0.0106	0.0535±0.0100
ERM	0.8	0.0118±0.0003	0.0119±0.0005	0.0117±0.0002
IRM	0.8	0.2300±0.0454	0.2916±0.0767	0.1684±0.0445
REX	0.8	0.0319±0.0052	0.0320±0.0077	0.0318±0.0074
ERM	1	0.0114±0.0003	0.0115±0.0005	0.0114±0.0002
IRM	1	0.0094±0.0015	0.0111±0.0026	0.0077±0.0013
REX	1	0.0217±0.0039	0.0219±0.0057	0.0215±0.0055
ERM	1.2	0.0110±0.0003	0.0111±0.0005	0.0110±0.0002
IRM	1.2	0.0059±0.0010	0.0071±0.0018	0.0048±0.0008
REX	1.2	0.0163±0.0029	0.0162±0.0043	0.0163±0.0041
ERM	1.4	0.0106±0.0002	0.0106±0.0004	0.0105±0.0002
IRM	1.4	0.0041±0.0007	0.0050±0.0013	0.0033±0.0006
REX	1.4	0.0122±0.0023	0.0122±0.0035	0.0122±0.0033
ERM	1.6	0.0101±0.0002	0.0101±0.0004	0.0101±0.0002
IRM	1.6	0.0031±0.0005	0.0037±0.0010	0.0024±0.0004
REX	1.6	0.0095±0.0019	0.0096±0.0029	0.0095±0.0028
ERM	1.8	0.0096±0.0002	0.0096±0.0004	0.0096±0.0002
IRM	1.8	0.0024±0.0004	0.0029±0.0008	0.0019±0.0003
REX	1.8	0.0082±0.0015	0.0084±0.0023	0.0081±0.0022

Table 13: Domain distance for anti-causal shift with the errors displayed being the model estimation error ± standard error.

Method	Domain distance	Average	Causal	Non-causal
ERM	0	0.0216±0.0019	0.0215±0.0027	0.0218±0.0027
IRM	0	0.9166±0.0064	0.9078±0.0123	0.9254±0.0025
REX	0	0.9117±0.0072	0.9055±0.0131	0.9179±0.0064
ERM	0.2	0.0215±0.0019	0.0213±0.0027	0.0217±0.0027
IRM	0.2	0.7615±0.0704	1.0172±0.0530	0.5058±0.0597
REX	0.2	0.2534±0.0205	0.2564±0.0320	0.2504±0.0274
ERM	0.4	0.0213±0.0018	0.0212±0.0027	0.0215±0.0027
IRM	0.4	0.5548±0.0547	0.7563±0.0416	0.3533±0.0433
REX	0.4	0.1015±0.0101	0.1044±0.0156	0.0987±0.0135
ERM	0.6	0.0211±0.0018	0.0209±0.0026	0.0212±0.0026
IRM	0.6	0.4568±0.0386	0.5944±0.0268	0.3192±0.0368
REX	0.6	0.0489±0.0052	0.0508±0.0085	0.0471±0.0065
ERM	0.8	0.0207±0.0018	0.0206±0.0026	0.0209±0.0026
IRM	0.8	0.2350±0.0465	0.2894±0.0771	0.1806±0.0503
REX	0.8	0.0275±0.0027	0.0288±0.0046	0.0261±0.0031
ERM	1	0.0204±0.0018	0.0202±0.0025	0.0205±0.0026
IRM	1	0.0140±0.0021	0.0161±0.0032	0.0120±0.0028
REX	1	0.0199±0.0020	0.0206±0.0032	0.0191±0.0027
ERM	1.2	0.0199±0.0017	0.0198±0.0025	0.0201±0.0025
IRM	1.2	0.0103±0.0015	0.0116±0.0021	0.0089±0.0022
REX	1.2	0.0173±0.0021	0.0177±0.0034	0.0168±0.0028
ERM	1.4	0.0195±0.0017	0.0193±0.0024	0.0197±0.0025
IRM	1.4	0.0114±0.0022	0.0122±0.0033	0.0106±0.0029
REX	1.4	0.0130±0.0014	0.0134±0.0022	0.0126±0.0019
ERM	1.6	0.0190±0.0016	0.0188±0.0024	0.0192±0.0024
IRM	1.6	0.0111±0.0015	0.0109±0.0020	0.0112±0.0023
REX	1.6	0.0109±0.0011	0.0112±0.0016	0.0106±0.0016
ERM	1.8	0.0185±0.0016	0.0183±0.0023	0.0187±0.0024
IRM	1.8	0.0142±0.0017	0.0137±0.0025	0.0148±0.0025
REX	1.8	0.0098±0.0010	0.0100±0.0014	0.0096±0.0015

Table 14: Domain distance for hybrid shift with the errors displayed being the model estimation error ± standard error.

A.4 Confounded shift for quantity of training domains experiment



(a) CS for experiment 1

(b) CS for experiment 2

(c) CS for experiment 3

(d) CF for experiment 1

(e) CF for experiment 2

(f) CF for experiment 3

(g) AC for experiment 1

(h) AC for experiment 2

(i) AC for experiment 3

(j) HB for experiment 1

(k) HB for experiment 2

(l) HB for experiment 3

Figure 9: Performance of REx, IRM and ERM for: covariate shift (CS), confounded shift (CF), anti-causal shift (AC), and hybrid shift (HB). The methods are plotted against the average model estimation error on the y-axis where the spread represents the causal and non-causal error. **Left:** Sample complexity (experiment 1) with the number of samples on the x-axis. **Center:** Quantity of training domain (experiment 2) with the amount of training samples on the x-axis. **Right:** Domain distance (experiment 3) with domain distance plotted on the x-axis.