Can Invariant Risk Minimization resist the temptation of learning spurious correlations?

Jochem van Lith
Supervisor(s): Rickard Karlsson, Stephan Bongers, Jesse Krijthe
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

Learning algorithms can perform poorly in unseen environments when they learn spurious correlations. This is known as the out-of-domain (OOD) generalization problem. Invariant Risk Minimization (IRM) is a method that attempts to solve this problem by learning invariant relationships. Motivating examples as well as counterexamples have been proposed on the performance of IRM. This work aims to clarify when the method works well and when it fails by testing its ability to learn invariant relationships. Therefore, experiments are done on a synthetic data model which simulates four data distribution shifts: covariate shift (CS), confounder based shift (CF), anti-causal shift (AC), and hybrid shift (HB). The experiments exploit IRM's behaviour with respect to hetero- and homoskedasticity and adaptation of the training environments. We measure the error with regards to the optimal invariant predictor and compare to the non-invariant Empirical Risk Minimization (ERM). The results show that IRM is generally able to learn invariance for the CS and CF shifts, especially when the deviation between the training environments is large. In the AC and HB shifts, this strongly depends on the values of the training environments.

1 Introduction

By using large amounts of data for training, learning models can find prediction rules that have good performance when applied to unseen validation data. As long as that data is from the same environments at least. Data often comes with spurious correlations that a model will recognize as a pattern [15, 2, 1]. When the environment changes, these factors might no longer be present and prediction accuracy will greatly decline in most cases.

Take for example a model to classify a cow in an image [13]. An image with a cow will often have a green background (pasture), which can be learned as a correlation to minimize the error when classifying cows in that same environment. However, when introduced to a cow in uncommon contexts (like a beach with sand and waves), it will usually not be classified correctly. This is called the out-of-domain (OOD) generalization problem.

To solve this problem, a model should learn invariant relationships: properties that remain stable across all environments [9, 3]. And a model should not learn spurious correlations. Let us apply this to the cow-classification example: the shape and the stains are examples of invariant properties. On the other hand, a pasture in the background is an example of a spurious correlation. In other words, invariance in classification can be seen as a causal relationship as to why an object should be assigned a certain label. The goal is to make a model base its classifications only on these specific relationships to increase OOD robustness.

Numerous methods have been introduced for this purpose, for example: Invariant Causal Prediction [6], Risk Extrapolation [4], Calibration-based Invariance [16] and Distributionally Robust Optimization [14].

This research will focus on a method introduced in 2019 called Invariant Risk Minimization (IRM) [10]. The principle of IRM is to find a representation of features, such that the optimal classifier on top of that representation is simultaneously optimal across all environments. This would be the ideal situation and is a large, bilevel optimization problem. For practical reasons, the authors simplify the problem to a version called IRMv1. In the paper they claim to have good performance when deployed in unseen environments. However, a couple of papers that discuss IRM criticize its performance. In particular, the simplified version IRMv1 as well as exposure to non-linear data and finite samples are criticized. This related work will be discussed in the next section.

We aim to clarify when IRM performs well and when it fails with respect to OOD generalization. IRM will be trained on a synthetic dataset that simulates four different data distribution shifts. Then we will determine whether it captures invariance by comparing to the optimal invariant predictor as an upper bound and to the non-invariant Empirical Risk Minimization (ERM) as a lower bound. The experiments are limited to the simplified version IRMv1. So the main research question will be:

For which data distribution shifts is the IRMv1 method able to capture invariance?

To properly answer this question, we will break it down into four sub-questions:

1. Which data distribution shifts are related to the OOD generalization problem?
2. How can we simulate these data distrbiution shifts in a synthetic dataset?
3. For which data distribution shifts is the prediction rule learned by IRMv1 similar to the optimal invariant predictor?
4. For which data distribution shifts does IRMv1 perform better than the non-invariant ERM?

In this work, we will first review several related papers in section 2. Then, we will familiarize with IRM in section 3. Section 4 provides answers to the first two sub-questions by describing the synthetic data model and the data distribution shifts that it can simulate. Next, the experiments are explained along with the generated results in section 5. These results will be discussed in section 6. The conclusion follows in section 7. We end with section 8, where responsible research will be addressed.

2 Related Work

IRM has been discussed in numerous papers. Motivating examples as well as counterexamples were proposed. In this section, the methods and conclusions of these papers are briefly described.

The original paper does experiments from which the authors conclude that the method performs well in unseen environments [10]. The experiments were done on a synthetic dataset and on the semi-synthetic CMNIST dataset.

In the paper called Invariant Risk Minimization Games, the authors state that they have designed a simpler algorithm which performs similar to or better than IRM on several different configurations of the CMNIST dataset [8]. At the same time, their results show that in all of the proposed cases IRM greatly outperforms ERM.

The previously mentioned simplification of IRM into IRMv1 is discussed in [12]. The authors provide theoretical proof that this simplification comes with failure modes. The proof is backed up with experiments on a synthetic dataset and the CMNIST dataset.

Another paper formally analyses the IRM objective [5]. The authors setup a simple data model and consider two cases: a linear and a non-linear scrambler of the observable features. For the linear case, it turns out that IRM only captures invariance when the number of training environments is greater than the number of dimensions of the environmental features. For the non-linear case, IRM is not able to generalize to new environments in general.

Performance of IRM compared to ERM with respect to sample complexity is discussed in [7]. Four different types of distribution shifts that occur in datasets are defined: covariate shift, confounder shift, anti-causal shift and hybrid (confounder and anti-causal) shift. These are translated to variants of the CMNIST dataset. In the covariate shift, ERM and IRM yield similar results. Whereas IRM wins in all the other shifts.

This work will extend on the data distribution shifts defined in [7]. Within the four shifts, we test on heteroskedasticity versus homoskedasticity. This property in combination with the shifts, greatly complicates invariant prediction and has been underexposed in the aforementioned literature. Moreover, where most of the related works hold on to fixed training environments, we take a step further by adapting the values of the training environments and by instantiating different deviations between them. This better reflects the real world and will therefore enhance our understanding of IRM's behaviour in OOD generalization.

3 Invariant Risk Minimization (IRM)

Invariant Risk Minimization (IRM) is a learning algorithm to solve the out-of-domain (OOD) generalization problem. To familiarize with IRM, we first formalize this problem. The invariant predictor will be introduced next. We explain how to find this predictor by defining the constrained optimization problem. Finally, the simplified version IRMv1 will be specified.

First, a formalization of the OOD generalization problem is required. Consider the collection of datasets $\mathrm{D} = \{D_e\}_{e \in \mathcal{E}_{tr}}$ where each dataset $D_e := \{(x_i^e, y_i^e)\}_{i=1}^{n_e}$ is acquired from a training environment $e \in \mathcal{E}_{tr}$. In this notation e represents the environmental index, i is the index of the data point and n_e indicates the total number of data points in the environment. A data point consists of two parts: the feature value $x_i^e \in X^e$ and its corresponding label $y_i^e \in Y^e$. All the data points in an environment e are drawn from some probability distribution $P(X^e, Y^e)$. The goal of OOD generalization is to learn a predictor $Y \approx f(X)$ that performs well in the training environments and many unseen environments described by $\mathcal{E}_{all} \supset \mathcal{E}_{tr}$. The performance of a predictor f in an environment e can be evaluated using the risk $R^e = \mathbb{E}_{X^e, Y^e}[\ell(f(X), Y)]$. The function ℓ is the loss function, which we instantiate as the mean squared error. Now we aim to minimise the expected risk across all environments, given by:

$$R^{OOD}(f) = max_{e \in \mathcal{E}_{all}} R^e(f) \text{ [10]}$$

So by using data obtained in the training environments $\mathcal{E}_{tr}$, one should find a predictor f that is robust in all environments $\mathcal{E}_{all}$. This is challenging, because the probability distribution in an unseen environment can be very different from the training environments.

IRM attempts to solve this problem by learning invariant relationships [10]. Let us explain the idea with the aforementioned cow-classification example. IRM wants to extract only the relevant features by defining a data representation Φ. This could be seen as removing the background from an image. The data representation could, for instance, be an image of the cow with a transparent background. Then it learns a classifier w on top of this data representation. This classifier will only be based on the properties of the cow, because the background has been removed. Now when introduced to new environments, the feature embedder Φ will ensure that only the cow is visible and therefore the classifier w will be simultaneously optimal for all environments. The product of the data representation and the classifier is called the *invariant predictor* $w \cdot \Phi$. This is formally phrased by [10] as:

Definition 1 *A data representation $\Phi : \mathcal{X} \to \mathcal{H}$ elicits an invariant predictor $w \cdot \Phi$ across*

environments $\mathcal{E}$ if there is a classifier $w : \mathcal{H} \rightarrow \mathcal{Y}$ simultaneously optimal for all environments, that is, $w \in arg\ min_{\bar{w}:\mathcal{H}\rightarrow\mathcal{Y}}\ R^e(\bar{w} \cdot \Phi)$ for all $e \in \mathcal{E}$.

This idea applies to many real world problems and in particular to the laws of physics [9]. Consider an apple that falls to the ground. The apple has many features: the color, the shape, the mass etcetera. However, its mass is the only relevant feature (Φ) with respect to the gravitational force. The acceleration can be computed with the same formula (w) regardless of the context. In this example the gravitational force forms the invariant relationship between the apple and its acceleration.

But now the question is: how to find this data representation? Across the training environments the data representation has two constraints: it should yield low risk and elicit an invariant predictor. This leads to the constrained optimization problem, which [10] mathematically describes by the following:

$$\min_{\substack{\Phi:\mathcal{X}\rightarrow\mathcal{H} \\ w:\mathcal{H}\rightarrow\mathcal{Y}}} \sum_{e\in\mathcal{E}_{tr}} R^e(w \cdot \Phi)$$

$$\text{subject to } w \in \arg\min_{\bar{w}:\mathcal{H}\rightarrow\mathcal{Y}} R^e(\bar{w} \cdot \Phi), \text{ for all } e \in \mathcal{E}_{tr}$$

It is very computational intensive to find the optimal solution, because the two constraints make it a bilevel optimization problem. For each variable in the outer optimization task (achieve low risk across all training environments), the entire inner optimization task should be solved (elicit an invariant predictor across all training environments). To make it more practical, the authors of [10] restrict the problem to the simplified version IRMv1:

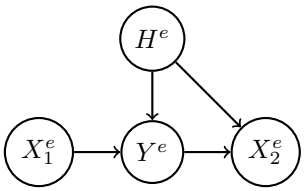
$$\min_{\Phi:\mathcal{X}\rightarrow\mathcal{Y}} \sum_{e\in\mathcal{E}_{tr}} R^e(1.0 \cdot \Phi) + \lambda \cdot ||\nabla_{w|w=1.0} R^e(w \cdot \Phi)||^2$$

The simplification is mainly in the classifier: $w = 1.0$. This makes that the invariant predictor becomes just Φ. Furthermore, the inner optimization problem is expressed by the gradient norm penalty which assesses the optimality of w. The regularizer λ can be increased to enhance importance to the invariance of the predictor. λ can take any positive value. When set to zero, it only optimizes low risk across training environments which is the same as ERM. When set to infinity, IRMv1 is equivalent to the original problem. The experiments in this paper will all be performed on IRMv1.

4 Synthetic Data Model

To be able to test if IRM learns invariant relationships, a simulation of the real world is required. Therefore, we generate data through a model on which we can perform experiments. In this section, the synthetic data model is described along with the data distribution shifts that it can simulate. This answers the first two research sub-questions.

The synthetic data model that we use in the experiments is inspired from the structural equation model defined in [10]. It consists of four random variables of which X_1 and X_2 are the observable features, Y is the underlying label, and H is a hidden confounder. Every variable follows a Gaussian distribution with a variance that depends on the environment. The variables are connected to each other through weights. A weight can either be zero (not connected) or Gaussian (connected). We can set these weights such that they represent various data distribution shifts, which will be explained in section 4.1. The goal is to predict Y^e from $X^e = [X_1^e, X_2^e]$. Figure 1 visualizes the model.



$$
\begin{aligned}
\mathrm{H}^e &\leftarrow \mathcal{N}(0, \sigma_e^2) \\
X_1^e &\leftarrow \mathcal{N}(0, \sigma_e^2) \\
Y^e &\leftarrow \mathcal{N}(0, \sigma_y^2) + W_{1\to y} X_1^e + W_{h\to y} H^e \\
X_2^e &\leftarrow \mathcal{N}(0, \sigma_2^2) + W_{y\to 2} Y^e + W_{h\to 2} H^e
\end{aligned}
$$

Figure 1: The synthetic data model used for the experiments, where Y^e should be predicted from $X^e = [X_1^e, X_2^e]$.

The model allows for a specific adaptation: stable or varying Y-noise across environments. This is called *homoskedastic* ($\sigma_y^2 = 1$ and $\sigma_2^2 = \sigma_e^2$) or *heteroskedastic* ($\sigma_y^2 = \sigma_e^2$ and $\sigma_2^2 = 1$) Y-noise respectively. Note that σ_2^2 changes as well in this adaptation. This is done to minimize side effects in X_2 (to which Y is added), because the adaptation is meant to only change the Y-noise. This will be further explained in section 5.1.

In this model the connection between X_1 and Y represents an invariant relationship. On the other hand, X_2 is spuriously correlated to Y. To better understand why this is the case, we discard the confounder H, set the remaining weights to identity, and define a least-squares predictor $\hat{Y} = X_1\hat{\alpha_1} + X_2\hat{\alpha_1}$. Regression from X_1 yields $\hat{\alpha_1} = 1$ and $\hat{\alpha_2} = 0$. Regression from X_2 yields $\hat{\alpha_1} = 0$ and $\hat{\alpha_2} = \frac{e^2}{e^2+\frac{1}{2}}$. And regression from X_1 and X_2 yields $\hat{\alpha_1} = \frac{1}{e^2+1}$ and $\hat{\alpha_2} = \frac{e^2}{e^2+1}$. To predict well in unseen environments, we want the coefficients to be independent of the environment. This is only the case when regressing from X_1. Then the optimal invariant predictor becomes: $\hat{Y} = X_1 \cdot 1 + X_2 \cdot 0$. This predictor is optimal with respect to invariance, because it is a correct causal explanation of how Y responds to changes of the features across all environments.

4.1 Data Distribution Shifts

Following [7] we describe four data distribution shifts and represent them in the model. This solves research sub-questions one and two. The distinction between the shifts can be found in the (indirect) relation between the label Y^e and the spurious feature X_2^e. The relation between the causal feature X_1^e and the label Y^e remains invariant in every shift. The shifts are visually represented in figure 2.

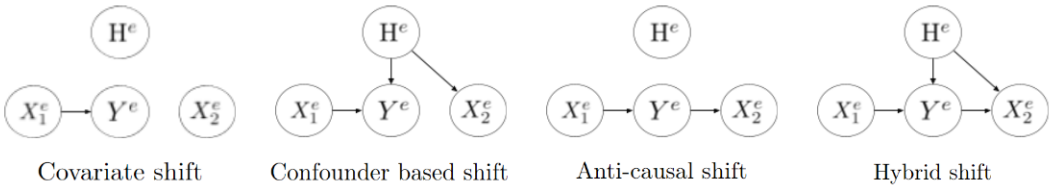


Figure 2: The four data distribution shifts represented in the synthetic data model.

The covariate shift (CS) occurs when the training data is not representative of the real world. For instance, we can have a dataset containing images of only brown cows. The brown color would then be the spurious feature X_2^e which might seem related to the label Y^e. But in the real world the X_2^e will follow a different distribution, because there exist

cows with different colors. A model that learned the brown color as a property of the cow, can still classify brown cows correctly. However, it might not recognize cows with different colors, because there exists no relation between the label Y^e and X_2^e. This shift is present in all the other shifts and therefore serves as a baseline. In this case the spurious correlation is weak, so we expect IRM to easily be able to capture invariance.

The confounder based shift induces spurious correlations through a confounding factor H^e that acts upon both Y^e and X_2^e. As an example, we can think of the confounder as a country. The Netherlands has many grassy landscapes and many cows. On the other hand, Egypt has a lot of sandy landscapes and many camels. The correlation through the confounding factor can easily be established: cows belong to grassy backgrounds and camels belong to sandy backgrounds. When introduced to a new environment this confounder can be very different. Let us say we move to India, where we find many cows on the beach. The correlation that sandy backgrounds belong to camels is no longer valid. In other words, this is a spurious correlation induced by the country as a confounder. It will be slightly harder to learn the invariant relationship in this shift. However, we expect IRM to succeed when the training environments are sufficiently different.

The anti-causal shift is characterized by the direct relation from Y^e to X_2^e. This induces a correlation between the label and an observable feature, but it is not causal and therefore not invariant. In the cow-classification example, this can be illustrated by the correlation between the cow (label) and the length of the grass (observable feature). Because cows eat grass, the grass tends to be shorter in pastures where cows live. But now we are introduced to a golf course, where the grass is mowed. The grass length is identical for any animal in the environment, so it cannot be used to classify a cow. Therefore this is a spurious correlation caused by an anti-causal feature. This correlation is very strong, so it will be challenging for IRM to capture invariance.

Lastly, the hybrid shift is a combination of the confounder based and anti-causal shift. This means that Y^e is directly (anti-causally) related to X_2^e as well as through a confounding factor H^e. This combination makes invariant prediction even harder, because there are many similarities between the label and the spurious feature.

5 Experiments and Results

By performing experiments on the synthetic data model from section 4, we wish to answer the main research question; for which data distribution shifts is the IRMv1 method able to capture invariance? The data distribution shifts are defined in section 4.1. IRM will be trained on environments that correspond to these shifts. We will measure the error of the learned prediction rule as opposed to the optimal invariant predictor, also referred to as the model estimation error. Furthermore, we compare the model estimation error of IRM to that of ERM. This will answer the last two research sub-questions. Ultimately, we want IRM to have a low model estimation error and that it outperforms ERM. In that case we can acknowledge that IRM captures invariance.

We begin with testing on heteroskedastic and homoskedastic Y-noise. Then, we will adapt the values of the training environments and consider different deviations between them.

There are some default settings throughout the experiments. When they are not mentioned, the following parameters will have these values:

- Number of dimensions $D = 10$

- Number of samples $n_s = 1000$
- Number of repetitions for average $n_r = 10$
- Training environments $\mathcal{E}_{tr} = \{0.2, 2.0, 5.0\}$
- Regularizer selection set $P = \{0, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$
- Regularizer cross-validation in last training environment
- Number of training iterations $n_{it} = 10000$
- Learning rate $\alpha = 1e-3$
- Gradient steps $\beta = 5e-4$
- Heteroskedastic Y-noise, where $\sigma_y^2 = \sigma_e^2$ and $\sigma_2^2 = 1$
- Weights $W_{h\to y}$ and $W_{h\to 2}$ set to $\frac{1}{D}\mathcal{N}(0,1)$ when there is a relation and 0 otherwise
- Weights $W_{1\to y}$ and $W_{y\to 2}$ set to I when there is a relation and 0 otherwise

Note that the weights related to the confounder ($W_{h\to y}$ and $W_{h\to 2}$) are Gaussian. This is done to simulate the real world, where confounding factors are present with a degree of randomness. The weights related to the label ($W_{1\to y}$ and $W_{y\to 2}$) would ideally also be Gaussian. However, the methods require many more samples and iterations to acquire the same results. Keeping in mind the scope of the project, we decided to assign fixed weights.

To clarify the data generating process, we instantiated $\mathcal{E}_{tr}$ with 100 samples for all shifts. The plots can be found in figure 3. These instances show that the variance of X_1 is proportional to the environment in all shifts. The variance of X_2 is constant across environments in the CS shift. In the other shifts its variance deviates a lot per environment, due to addition of H and Y. Similar instances with homoskedastic Y-noise and the 3 dimensional representations (with Y) can be found in the appendix A.

In all experiments, the methods learn a prediction rule of the form $\hat{Y} = [\hat{W}_{1\to y}, \hat{W}_{y\to 2}]X$. The optimal invariant predictor is given by $\hat{Y} = [W_{1\to y}, 0]X$ which is equal to $\hat{Y} = [1, 0]X$, because we always set $W_{1\to y}$ to I. The model estimation error is the distance between the learned prediction rule and the optimal invariant predictor: $||[\hat{W}_{1\to y}, \hat{W}_{y\to 2}] - [1, 0]||^2$.

5.1 Heteroskedastic and Homoskedastic Y-noise

In the real world the noise of the underlying label can be varying or stable [11]. We call this hetero- and homoskedastic Y-noise. To simulate heteroskedasticity, we set $\sigma_y^2 = \sigma_e^2$ and $\sigma_2^2 = 1$. Whereas homoskedasticity is reproduced by $\sigma_y^2 = 1$ and $\sigma_2^2 = \sigma_e^2$. As mentioned in section 4, we change σ_2^2 here as well to reduce side effects. To see what happens with side effects, this experiment also includes the case $\sigma_y^2 = 1$ and $\sigma_2^2 = 1$ which we call homoskedastic Y-noise with constant X_2. The results are displayed in figure 4.

First, we consider heteroskedasticity. In the CS shift, the spurious features follow a constant distribution ($\mathcal{N}(0,1)$). When only minimizing the risk across training environments, like ERM, we would expect these features to already mostly be discarded for prediction. This corresponds to the results. IRM outperforms ERM in all the shifts. This is also conform to expectations, because these shifts produce spurious correlations that ERM eagerly uses for minimizing its training error. IRM successfully ignores all spurious features in the

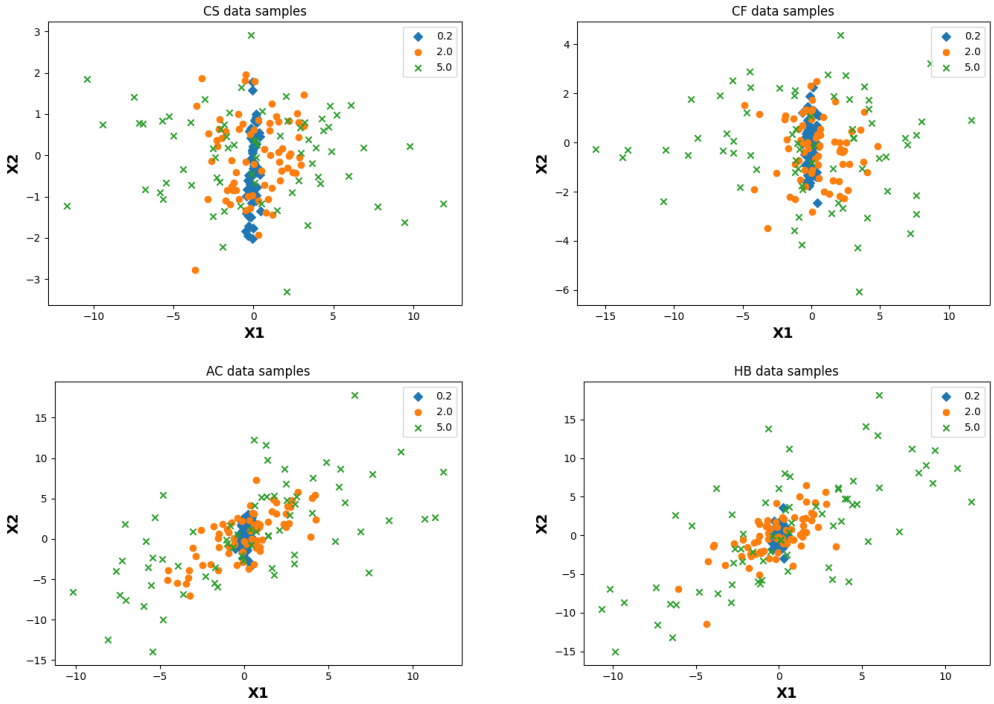


Figure 3: 2D instances of training environments $\mathcal{E}_{tr} = \{0.2, 2.0, 5.0\}$ with 100 samples and heteroskedastic Y-noise for every data distribution shift.

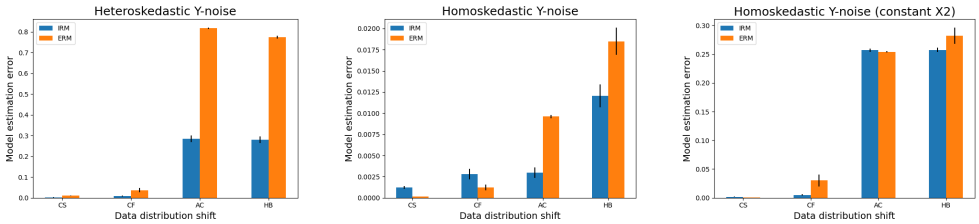


Figure 4: The results of the heteroskedastic and homoskedastic Y-noise experiment. The black bars indicate the standard error of measurement. The corresponding tables are in the appendix B.

CF shift with an error of almost 0. The error is slightly higher in the AC and HB shifts. This is because of the anti-causal relation between Y and X_2, which forms a strong spurious correlation. The fact that it wins from ERM with a large margin implies that the regularizer is set to an appropriate value. So the sub-optimal results are probably caused by problems in the gradient estimation.

Both methods have a significantly smaller error in the homoskedastic case. This is plausible, because in this situation the Y-noise is constant across environments which makes regression simpler. IRM wins from ERM in the AC and HB shifts, just like with heteroskedastic Y-noise. Remarkably, IRM performs less well than ERM in the CS and CF

shifts under homoskedastic Y-noise. By definition IRM becomes ERM when the regularizer is set to 0. So the regularizer should have been smaller for these shifts.

The outer right plot in figure 4 shows the results for homoskedastic Y-noise with constant X_2. IRM succeeds in filtering out the confounder in this situation. This follows from its low error in the CF shift and the comparison between the AC and HB shifts. The only difference between these two shifts is the confounder. ERM's error increases as the confounder is added, whereas IRM's error remains constant. On the other hand, the anti-causal link between Y and X_2 is the reason for IRM's bad performance in the AC and HB shifts. This is not surprising, because in this setting the X_2-noise follows the same distribution as the Y-noise. When the anti-causal link is present, a spurious correlation is formed which is almost indistinguishable from an invariant relationship. Therefore, we will not consider this case for the other experiments.

5.2 Adapting Training Environments

The previous experiment was done on a fixed set of training environments. But in OOD generalization we cannot make any assumptions on this set. In this section, we will perform experiments with different training environments. First, we will adapt the values and keep the mutual deviation fixed. Then, we will test on different deviations between them.

5.2.1 Values of Training Environments

In this experiment, we adapt the values of the training environments while keeping the deviation between them constant. The set of training environments is given by $\mathcal{E}_{tr} = \{0.2 + A_{\mathcal{E}_{tr}}, 2.0 + A_{\mathcal{E}_{tr}}, 5.0 + A_{\mathcal{E}_{tr}}\}$ where $A_{\mathcal{E}_{tr}}$ is the amount of adaptation ranging from 0 to 15.

The results are displayed in figure 5. The errors clearly increase for a larger adaptation. This is what we would expect, because the variance of the Y-noise depends on the environment. The data of the label will therefore be more skewed, which makes it harder to predict. In case of homoskedastic Y-noise, this is the other way around. The results of the same experiment under homoskedastic Y-noise are in the appendix C.1.3.

Note that the plots of the CS and CF shifts in figure 5 are convex, whereas those of the AC and HB shifts are concave. To understand these different curves, we should look at the weights in the learned prediction rules. Recall that the optimal invariant predictor has the weights $W = [1, 0]$. The results with the separate error for the causal and non-causal weights are in the appendix C.1.1.

These results show that in the CS and CF shifts, as σ_e^2 grows to infinity, the methods seem to learn the weights $\hat{W} = [\sim 1, \infty]$. So it learns the causal weight correctly, but assigns too much weight to the spurious feature. This is because X_2 follows a distribution which is not influenced (or only slightly in the CF shift) by the environment. Hence, it multiplies this X_2 with a weight proportional to σ_e^2 to predict Y.

Meanwhile in the AC and HB shifts, as σ_e^2 grows to infinity, the methods seem to learn the weights $\hat{W} = [\sim 0, \sim 1]$ and $\hat{W} = [\sim 0.1, \sim 0.9]$ respectively. This is opposite to the optimal invariant predictor. The reason for this is that the distribution of X_2 converges to the same distribution as Y when σ_e^2 increases. So the models set the non-causal weight to approximately 1, such that the prediction rule yields the same distribution as Y.

This explains the behavior of the methods with respect to the optimal invariant predictor. But ERM and IRM also have different performance. In the CS and CF shifts, for a large $A_{\mathcal{E}_{tr}}$, IRM has a greater error than ERM. As mentioned before, we know by definition that

the regularizer is then set too large. The regularizer is cross validated in the last training environment, which also has the most variance. This makes it hard to set an appropriate value. In the AC and HB shifts, IRM outperforms ERM when $A_{\mathcal{E}_{tr}}$ is small. So before X_2 converges to the same distribution as Y, IRM is actually able to identify the invariant relationship.

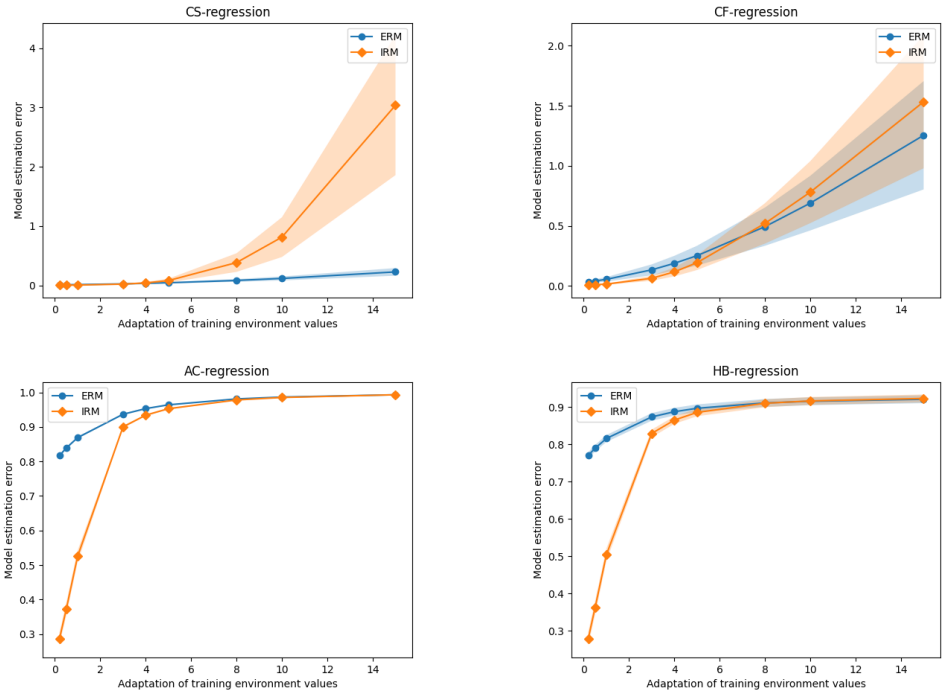


Figure 5: The results of adapting the values of the training environments with heteroskedastic Y-noise. The corresponding tables are in the appendix C.1.2.

5.2.2 Deviation Between Training Environments

In the real world it can occur that the training environments are very similar, but just as well that there is a large difference between them. In this experiment we will test on different deviations between the training environments. The set of training environments is given by $\mathcal{E}_{tr} = \{0.2, 0.2 + D_{\mathcal{E}_{tr}}, 0.2 + 2 \cdot D_{\mathcal{E}_{tr}}\}$ where $D_{\mathcal{E}_{tr}}$ is the deviation ranging from 0.1 to 15. We expect the error to be lower for a larger deviation; it should be easier to find the invariant relationship when the spurious correlations differ more across the training environments.

The results can be found in figure 6. Both methods clearly perform less well when the deviation is larger, which is against expectation. The reason for this is that a larger deviation comes with a larger value of σ_e^2 (we got ourselves a confounder!). Recall from the previous experiment that a larger value of σ_e^2 yields a greater error under heteroskedastic Y-noise. For homoskedastic Y-noise this is the other way around. We did this experiment as well with homoskedastic Y-noise of which the results are in the appendix C.2.2. These results, just like before, show a decreasing trend. Hence, we should keep the outcome of the previous experiment in mind when observing the current results.

In the CS and CF shifts, we can see similar curves as in the previous experiment which we can devote to the increasing σ_e^2. Nevertheless, IRM performs much better in comparison to ERM in this experiment. So IRM captures invariance particularly well for a large $D_{\mathcal{E}_{tr}}$. The regularizer seems to be set to an appropriate value in contrast to the previous experiment. This is probably because now the regularizer is cross validated in a very different environment than the other training environments, which makes it easier to identify invariance.

Likewise, the curves of the plots in the AC and HB shifts are similar to the previous experiment. ERM yields low error for a small deviation, because the values of σ_e^2 are so small that it corresponds to the CS shift. Other than that, there is no difference in the ratio between the error of IRM and ERM. This means that the deviation by itself has no effect on IRM's performance in these shifts. So even though the training environments are very different, IRM apparently has problems to recognize the invariant relationship in the presence of the anti-causal link when σ_e^2 is large. In this situation the spurious correlation becomes very strong and finding the invariant data representation through the gradient estimation seems to fail.

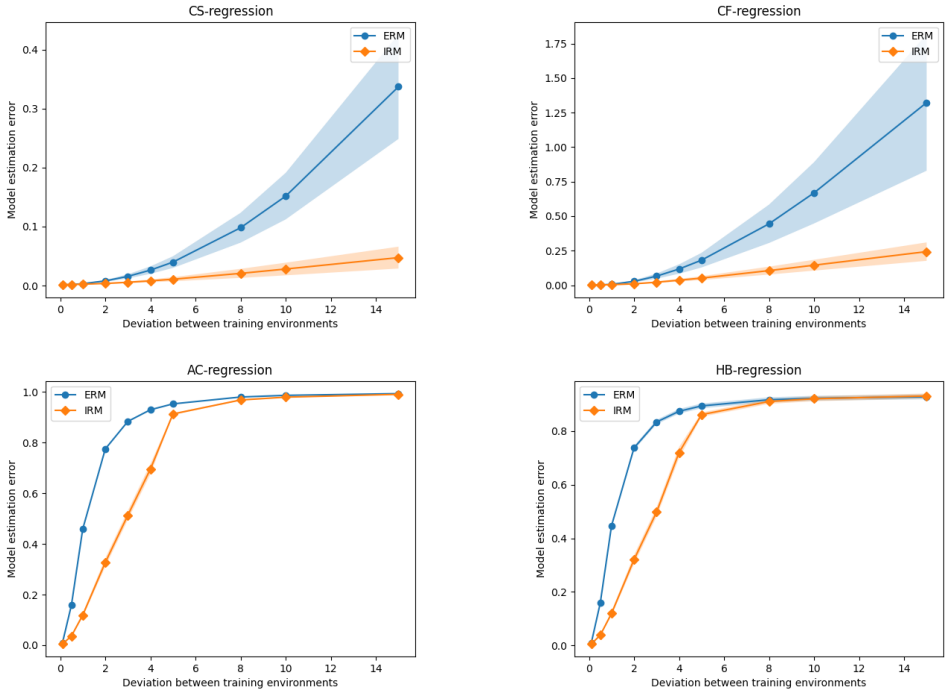


Figure 6: The results of the deviation between training environments experiment with heteroskedastic Y-noise. The corresponding tables are in the appendix C.2.1.

6 Discussion

The IRM learning method attempts to solve the OOD generalization problem by learning invariant relationships [10]. The practical version IRMv1 optimizes risk and attempts to elicit an invariant predictor across the training environments. In section 3, the method

is described in detail. We exposed IRM to different data distribution shifts in a series of experiments to test its ability to capture invariance. In this section we will discuss the outcome of the experiments and consider the limitations.

In the covariate shift (CS) there is no relation between the label Y and the spurious features X_2. It is present in all the other shifts and serves as a baseline. IRM generally learns the invariant relationship in this shift, but sometimes it fails to set an appropriate value for the regularizer. This is especially the case for large values of the training environments and homoskedastic Y-noise. However, when increasing the deviation between the training environments this problem seems to be solved. The regularizer is then cross validated in a very different environment, which reveals the spurious correlations.

In the confounder based shift (CF), the label Y and the spurious features X_2 are related through a confounder H. This spurious correlation makes it slightly harder to recognize the invariant features. IRM is able to capture invariance for small values of σ_e^2 and under heteroskedastic Y-noise. For larger values of σ_e^2 and homoskedastic Y-noise, it yields a greater error than ERM. Similar to the CS shift, this is probably caused by selecting a regularizer that is too large. This problem seems to disappear in the same way as for the CS shift: when the deviation between the training environments is increased.

The anti-causal shift (AC) is characterized by the relation from Y to X_2. This induces a strong spurious correlation, because the spurious features are influenced by the label. In general IRM performs better than ERM in this shift. Only when the values of the training environments are large, the two methods yield similar errors. This implies that the regularizer is selected appropriately; it only applies the gradient norm penalty when it can gain advantage over ERM. So it generally performs better than the non-invariant ERM, but at the same time it often fails to find the optimal invariant solution. The simplified version IRMv1 uses gradient estimation to assess the optimality of a predictor. But this seems to give problems in the presence of strong spurious correlations, which then yields a significant model estimation error.

The hybrid shift (HB) includes the confounder H and the anti-causal relation between Y and X_2. The anti-causal link induces a very strong spurious correlation, which dominates this shift. Hence, IRM performs similar to the AC shift in this case.

This work also has its limitations. The experiment where different deviations between the training environments are considered (in section 5.2.2), does not only reflect on the deviation. By increasing the deviation, the value of σ_e^2 also increases which dominates the results. A possible solution to this would be to change the synthetic data model, such that the means of the random variables depend on the environment rather than the variance. A larger deviation will then only have impact on the difference between the environments and not affect the environments itself.

Additionally, the weights in the synthetic data model should ideally all be Gaussian. In our experiments we set the weights related to the label ($W_{1\to y}$ and $W_{y\to 2}$) to identity for simplicity. But it would be a better simulation of the real world to include a degree of randomness in these relations.

Furthermore, we argued about the regularizer and the gradient estimation, but it would be of great addition if these values would be included in the results. For future work it would also be interesting to perform experiments with a different regularizer selection set and to use different environments for its cross-validation.

7 Conclusion

We presented an analysis of IRM to answer the question: for which data distribution shifts is the IRMv1 method able to capture invariance? To properly answer this question we instantiated four data distribution shifts: covariate shift (CS), confounder based shift (CF), anti-causal shift (AC), and hybrid shift (HB). These shifts were simulated in a synthetic data model. Using the model, we performed experiments which considered hetero- and homoskedasticity as well as adaptation to the values of the training environments and their mutual deviation. We compared IRM's learned prediction rule to the optimal invariant predictor and to the non-invariant ERM.

We found that in the CS and CF shifts, IRM is generally able to capture invariance. Only when the deviation between training environments is small, it performs poorly. This seems to be caused by the selection of an inappropriate value for the regularizer. On the other hand, it strongly depends on the values of the training environments whether IRM can learn the invariant relationships in the AC and HB shifts. IRM always outperforms ERM in these shifts, but has problems with finding the optimal solutions. This is especially the case for large values of the training environments. It seems that the gradient estimation could be part of the problem.

8 Responsible Research

For research to be reliable, it is important to adhere to scientific integrity. A researcher should be honest, transparent, and avoid any external influence without scientific motive. In this section, we will reflect on our own integrity and explain how we tried to ensure this.

We ensured that the research data is reliable. The entire experimental setup is described in the paper such that it is reproducible. The data points in our results were obtained by averaging over 10 repetitions. This accounts for randomness and coincidental side effects. All data points include a standard error of measurement and no data points were discarded. We performed several experiments, which could not all be included in the paper. However, the additional results are attached in the appendix. Whenever we chose for certain settings, we carefully explained why we used them. This was in order to show that we did not select them to simply manipulate the results.

This work builds upon existing research. All the papers that were extended on were referenced and it was carefully explained what parts were adopted. We used multiple sources to increase reliability. Moreover, we critically discussed the results of related works and extended on parts that were underexposed. However, we used some sources that were published as a part of the same conference: *ICLR*. This could mean that works are biased, but we believe that this is not the case in this situation. For example, [10] which introduces IRM and [5] which criticizes IRM, were both published as a part of *ICLR*. This criticism indicates that there is no bias induced by this conference. *ICLR* is also a universal event that is supported by many researchers from many different countries, which is another indication for reliability.

The experiments that we performed also had its limitations. It is important to be aware of these limitations and to describe how these could be improved. In the discussion section, the limitations were discussed along with possible solutions for future work.

References

[1] Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. Revisiting visual question answering baselines. *ECCV*, 2016.

[2] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. *CVPR*, 2011.

[3] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 2016.

[4] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Remi Le Priol, Dinghuai Zhang, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *ICLR*, 2021.

[5] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *ICLR*, 2020.

[6] Jonas Peters, Peter Buhlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society*, 2016.

[7] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R. Varshney. Empirical or invariant risk minimization? a sample complexity perspective. *ICLR*, 2020.

[8] Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. Invariant risk minimization games. *arXiv*, 2020.

[9] David Lopez-Paz. From dependence to causation. *PhD thesis, University of Cambridge*, 2016.

[10] Martin Arjovsky, Leon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *ICLR*, 2019.

[11] Patrick J. Rosopa, Meline M. Schaffer, and Amber N. Schroeder. Managing heteroscedasticity in general linear models. *Psychological Methods*, 2013.

[12] Pritish Kamath, Akilesh Tangella, Danica J. Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? *CoRR*, 2021.

[13] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. *ECCV*, 2018.

[14] 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. *ICLR*, 2020.

[15] Bob L. Sturm. A simple method to determine if a music information retrieval system is a âhorseâ. *IEEE Transactions on Multimedia*, 2014.

[16] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *NeurIPS*, 2021.

A Instances of training environments

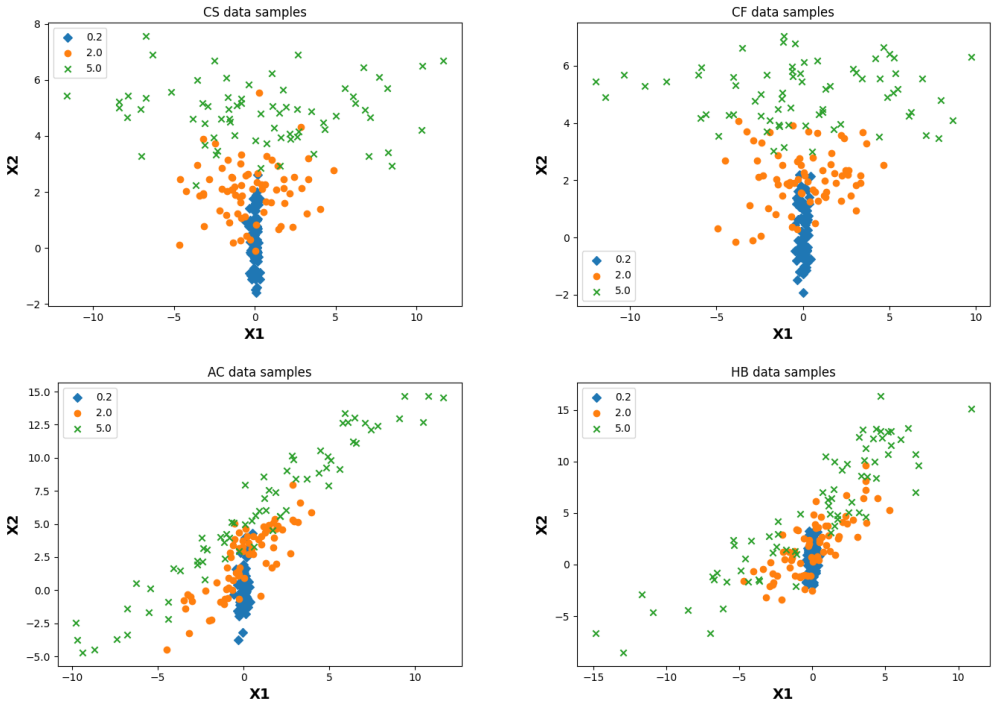


Figure 7: 2D instances of training environments $\mathcal{E}_{tr} = \{0.2, 2.0, 5.0\}$ with 100 samples for every data distribution shift under homoskedastic Y-noise.

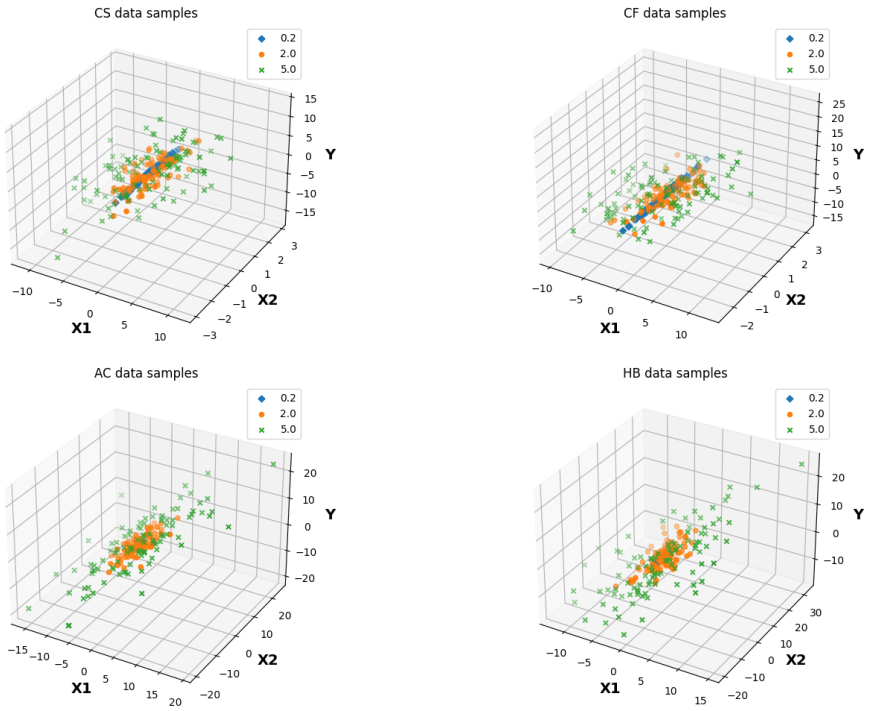


Figure 8: 3D instances of training environments $\mathcal{E}_{tr} = \{0.2, 2.0, 5.0\}$ with 100 samples for every data distribution shift under heteroskedastic Y-noise.

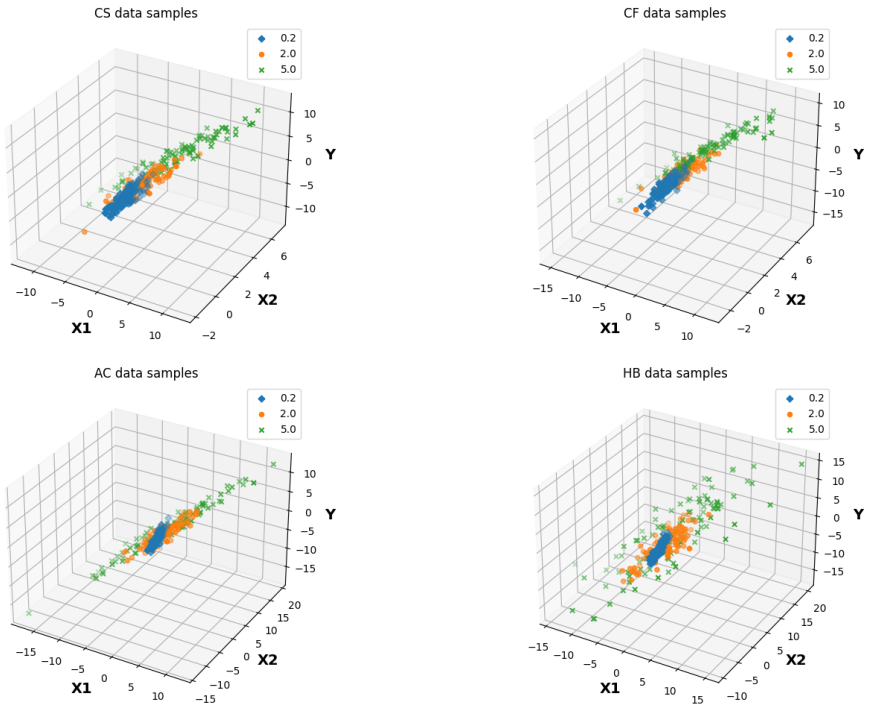


Figure 9: 3D instances of training environments $\mathcal{E}_{tr} = \{0.2, 2.0, 5.0\}$ with 100 samples for every data distribution shift under homoskedastic Y-noise.

B Heteroskedastic and Homoskedastic Y-noise Tables

Method	Shift	Model estimation error
ERM	CS	0.0098 ±0.0019
IRM	CS	0.0026 ±0.0005
ERM	CF	0.0372 ±0.0103
IRM	CF	0.0084 ±0.0019
ERM	AC	0.8173 ±0.0028
IRM	AC	0.2851 ±0.0159
ERM	HB	0.7749 ±0.0075
IRM	HB	0.2802 ±0.0156

Table 1: The results for heteroskedastic Y-noise corresponding to figure 4.

Method	Shift	Model estimation error
ERM	CS	0.0002 ±0.0000
IRM	CS	0.0012 ±0.0002
ERM	CF	0.0012 ±0.0003
IRM	CF	0.0028 ±0.0006
ERM	AC	0.0096 ±0.0002
IRM	AC	0.0030 ±0.0006
ERM	HB	0.0185 ±0.0016
IRM	HB	0.0120 ±0.0014

Table 2: The results for homoskedastic Y-noise corresponding to figure 4.

Method	Shift	Model estimation error
ERM	CS	0.0002 ±0.0000
IRM	CS	0.0012 ±0.0002
ERM	CF	0.0012 ±0.0003
IRM	CF	0.0028 ±0.0006
ERM	AC	0.0096 ±0.0002
IRM	AC	0.0030 ±0.0006
ERM	HB	0.0185 ±0.0016
IRM	HB	0.0120 ±0.0014

Table 3: The results for homoskedastic Y-noise with constant X_2 corresponding to figure 4.

C Adapting Training Environments Additional Material

C.1 Values of Training Environments

C.1.1 Heteroskedastic Y-noise Causal and Non-causal Figures

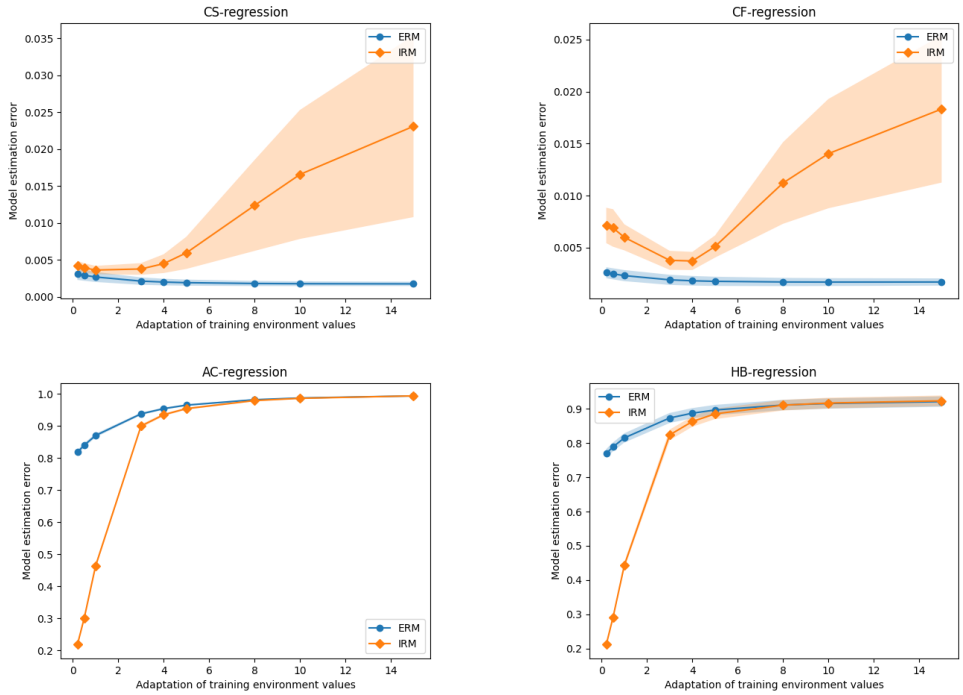


Figure 10: The causal model estimation error of adapting the values of the training environments under heteroskedastic Y-noise.

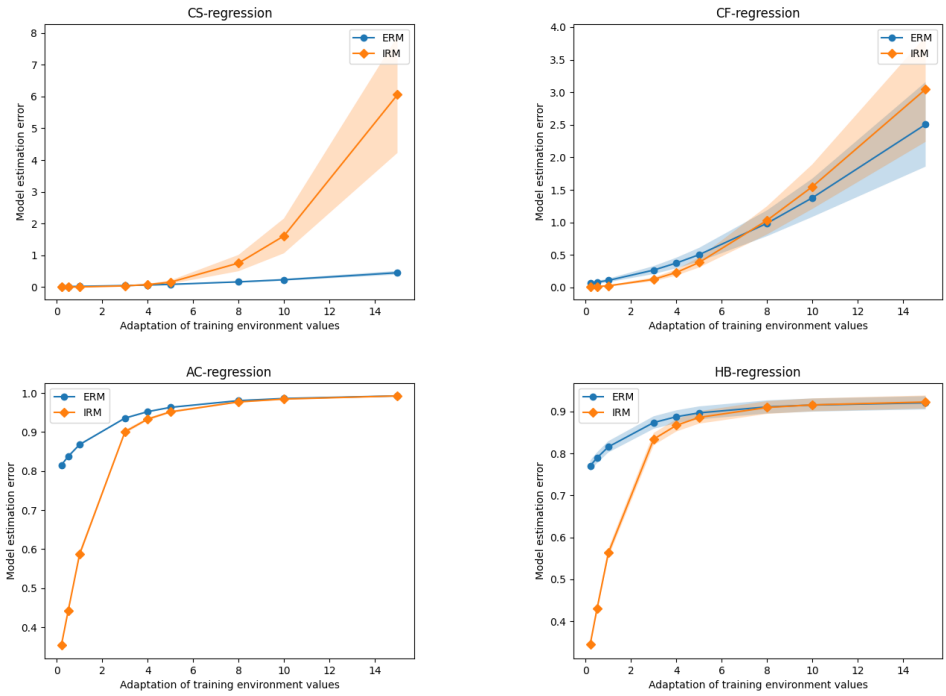


Figure 11: The anti-causal model estimation error of adapting the values of the training environments under heteroskedastic Y-noise.

C.1.2 Heteroskedastic Y-noise Tables

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.0101 ± 0.0023	0.0031 ± 0.0008	0.0172 ± 0.0028
IRM	0.2	0.0029 ± 0.0006	0.0042 ± 0.0007	0.0016 ± 0.0007
ERM	0.5	0.0112 ± 0.0027	0.0029 ± 0.0007	0.0195 ± 0.0032
IRM	0.5	0.0033 ± 0.0005	0.0039 ± 0.0006	0.0026 ± 0.0008
ERM	1	0.0133 ± 0.0033	0.0027 ± 0.0007	0.0239 ± 0.0039
IRM	1	0.0049 ± 0.0011	0.0036 ± 0.0006	0.0062 ± 0.0020
ERM	3	0.0257 ± 0.0070	0.0021 ± 0.0005	0.0493 ± 0.0073
IRM	3	0.0212 ± 0.0075	0.0038 ± 0.0008	0.0387 ± 0.0123
ERM	4	0.0342 ± 0.0095	0.0020 ± 0.0005	0.0664 ± 0.0095
IRM	4	0.0418 ± 0.0163	0.0045 ± 0.0013	0.0792 ± 0.0271
ERM	5	0.0443 ± 0.0124	0.0019 ± 0.0004	0.0866 ± 0.0119
IRM	5	0.0807 ± 0.0323	0.0060 ± 0.0022	0.1554 ± 0.0536
ERM	8	0.0834 ± 0.0235	0.0018 ± 0.0003	0.1650 ± 0.0216
IRM	8	0.3846 ± 0.1567	0.0124 ± 0.0061	0.7568 ± 0.2561
ERM	10	0.1170 ± 0.0331	0.0018 ± 0.0003	0.2322 ± 0.0301
IRM	10	0.8139 ± 0.3349	0.0166 ± 0.0087	1.6113 ± 0.5468
ERM	15	0.2272 ± 0.0647	0.0018 ± 0.0003	0.4527 ± 0.0583
IRM	15	3.0392 ± 1.1789	0.0231 ± 0.0123	6.0553 ± 1.8321

Table 4: The results of adapting the values of the training environments under heteroskedastic Y-noise with CS-regression. This corresponds to figure 5.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.0347 ± 0.0115	0.0026 ± 0.0005	0.0669 ± 0.0165
IRM	0.2	0.0082 ± 0.0021	0.0071 ± 0.0017	0.0092 ± 0.0040
ERM	0.5	0.0417 ± 0.0140	0.0025 ± 0.0005	0.0809 ± 0.0200
IRM	0.5	0.0107 ± 0.0030	0.0069 ± 0.0018	0.0146 ± 0.0055
ERM	1	0.0552 ± 0.0188	0.0023 ± 0.0005	0.1081 ± 0.0267
IRM	1	0.0153 ± 0.0050	0.0060 ± 0.0013	0.0246 ± 0.0090
ERM	3	0.1343 ± 0.0452	0.0019 ± 0.0005	0.2667 ± 0.0611
IRM	3	0.0642 ± 0.0212	0.0038 ± 0.0009	0.1246 ± 0.0297
ERM	4	0.1886 ± 0.0623	0.0018 ± 0.0005	0.3753 ± 0.0817
IRM	4	0.1171 ± 0.0372	0.0037 ± 0.0009	0.2305 ± 0.0475
ERM	5	0.2522 ± 0.0820	0.0017 ± 0.0004	0.5027 ± 0.1045
IRM	5	0.1939 ± 0.0612	0.0051 ± 0.0011	0.3827 ± 0.0767
ERM	8	0.4938 ± 0.1592	0.0017 ± 0.0004	0.9859 ± 0.1985
IRM	8	0.5210 ± 0.1680	0.0112 ± 0.0039	1.0307 ± 0.2162
ERM	10	0.6897 ± 0.2275	0.0017 ± 0.0004	1.3778 ± 0.2942
IRM	10	0.7813 ± 0.2580	0.0140 ± 0.0053	1.5485 ± 0.3421
ERM	15	1.2542 ± 0.4506	0.0017 ± 0.0003	2.5068 ± 0.6496
IRM	15	1.5332 ± 0.5531	0.0183 ± 0.0071	3.0480 ± 0.8096

Table 5: The results of adapting the values of the training environments under heteroskedastic Y-noise with CF-regression. This corresponds to figure 5.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.8168 ± 0.0034	0.8177 ± 0.0066	0.8158 ± 0.0024
IRM	0.2	0.2863 ± 0.0179	0.2178 ± 0.0048	0.3548 ± 0.0030
ERM	0.5	0.8387 ± 0.0032	0.8396 ± 0.0062	0.8377 ± 0.0024
IRM	0.5	0.3718 ± 0.0187	0.3006 ± 0.0053	0.4430 ± 0.0044
ERM	1	0.8689 ± 0.0030	0.8698 ± 0.0056	0.8680 ± 0.0024
IRM	1	0.5251 ± 0.0168	0.4630 ± 0.0075	0.5872 ± 0.0070
ERM	3	0.9365 ± 0.0021	0.9373 ± 0.0038	0.9358 ± 0.0021
IRM	3	0.9000 ± 0.0034	0.8993 ± 0.0053	0.9008 ± 0.0046
ERM	4	0.9529 ± 0.0018	0.9536 ± 0.0032	0.9522 ± 0.0018
IRM	4	0.9339 ± 0.0025	0.9346 ± 0.0039	0.9332 ± 0.0034
ERM	5	0.9639 ± 0.0016	0.9645 ± 0.0027	0.9632 ± 0.0017
IRM	5	0.9527 ± 0.0021	0.9536 ± 0.0033	0.9517 ± 0.0028
ERM	8	0.9810 ± 0.0011	0.9814 ± 0.0019	0.9805 ± 0.0013
IRM	8	0.9780 ± 0.0014	0.9787 ± 0.0021	0.9772 ± 0.0019
ERM	10	0.9864 ± 0.0009	0.9868 ± 0.0016	0.9860 ± 0.0011
IRM	10	0.9851 ± 0.0011	0.9857 ± 0.0017	0.9845 ± 0.0016
ERM	15	0.9930 ± 0.0007	0.9933 ± 0.0011	0.9927 ± 0.0008
IRM	15	0.9930 ± 0.0008	0.9934 ± 0.0011	0.9926 ± 0.0011

Table 6: The results of adapting the values of the training environments under heteroskedastic Y-noise with AC-regression. This corresponds to figure 5.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.7702 ± 0.0085	0.7696 ± 0.0126	0.7709 ± 0.0123
IRM	0.2	0.2791 ± 0.0176	0.2120 ± 0.0043	0.3461 ± 0.0051
ERM	0.5	0.7893 ± 0.0088	0.7886 ± 0.0130	0.7899 ± 0.0128
IRM	0.5	0.3613 ± 0.0183	0.2921 ± 0.0049	0.4305 ± 0.0067
ERM	1	0.8154 ± 0.0092	0.8148 ± 0.0135	0.8160 ± 0.0134
IRM	1	0.5037 ± 0.0168	0.4440 ± 0.0090	0.5633 ± 0.0103
ERM	3	0.8733 ± 0.0101	0.8729 ± 0.0146	0.8738 ± 0.0149
IRM	3	0.8290 ± 0.0103	0.8235 ± 0.0155	0.8344 ± 0.0144
ERM	4	0.8872 ± 0.0102	0.8868 ± 0.0148	0.8877 ± 0.0152
IRM	4	0.8648 ± 0.0102	0.8625 ± 0.0151	0.8671 ± 0.0146
ERM	5	0.8965 ± 0.0104	0.8961 ± 0.0149	0.8969 ± 0.0154
IRM	5	0.8854 ± 0.0102	0.8847 ± 0.0152	0.8860 ± 0.0148
ERM	8	0.9108 ± 0.0105	0.9105 ± 0.0151	0.9111 ± 0.0156
IRM	8	0.9099 ± 0.0105	0.9101 ± 0.0155	0.9097 ± 0.0153
ERM	10	0.9153 ± 0.0105	0.9151 ± 0.0151	0.9156 ± 0.0157
IRM	10	0.9162 ± 0.0106	0.9163 ± 0.0155	0.9160 ± 0.0154
ERM	15	0.9207 ± 0.0105	0.9205 ± 0.0151	0.9209 ± 0.0157
IRM	15	0.9228 ± 0.0106	0.9229 ± 0.0155	0.9228 ± 0.0155

Table 7: The results of adapting the values of the training environments under heteroskedastic Y-noise with HB-regression. This corresponds to figure 5.

C.1.3 Homoskedastic Y-noise Figures

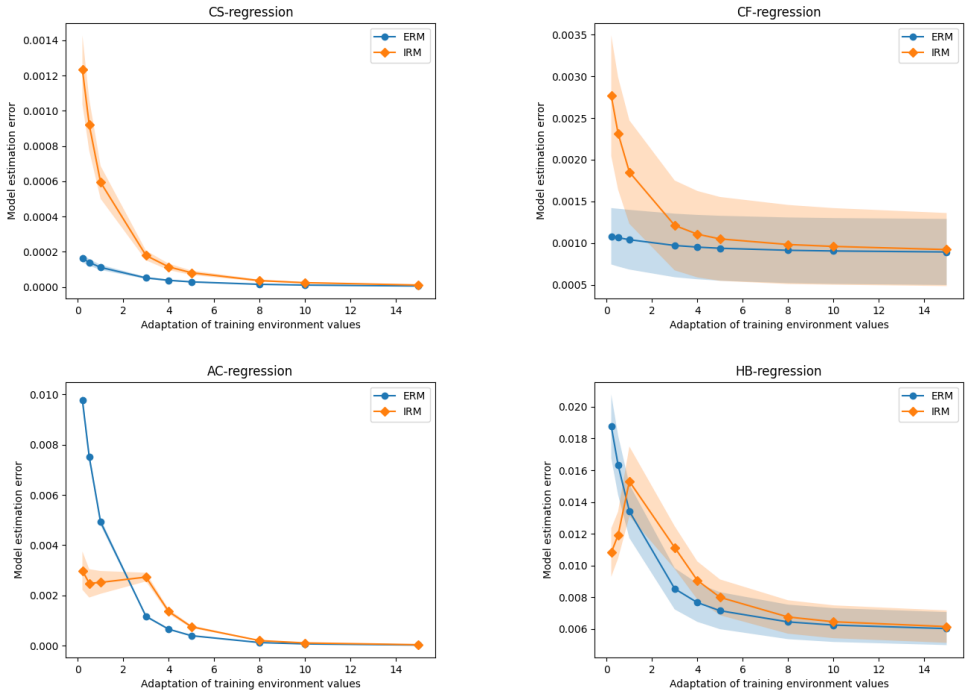


Figure 12: The results of adapting the values of the training environments under homoskedastic Y-noise.

C.1.4 Homoskedastic Y-noise Tables

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.0002 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000
IRM	0.2	0.0012 ± 0.0002	0.0010 ± 0.0002	0.0014 ± 0.0003
ERM	0.5	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000
IRM	0.5	0.0009 ± 0.0001	0.0008 ± 0.0001	0.0011 ± 0.0002
ERM	1	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
IRM	1	0.0006 ± 0.0001	0.0005 ± 0.0001	0.0007 ± 0.0001
ERM	3	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
IRM	3	0.0002 ± 0.0000	0.0002 ± 0.0000	0.0002 ± 0.0000
ERM	4	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	4	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
ERM	5	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	5	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
ERM	8	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	8	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
ERM	10	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	10	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
ERM	15	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	15	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000

Table 8: The results of adapting the values of the training environments under homoskedastic Y-noise with CS-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.0011 ± 0.0003	0.0004 ± 0.0001	0.0017 ± 0.0006
IRM	0.2	0.0028 ± 0.0007	0.0020 ± 0.0005	0.0035 ± 0.0013
ERM	0.5	0.0011 ± 0.0003	0.0004 ± 0.0001	0.0017 ± 0.0006
IRM	0.5	0.0023 ± 0.0007	0.0015 ± 0.0004	0.0031 ± 0.0013
ERM	1	0.0010 ± 0.0004	0.0004 ± 0.0001	0.0017 ± 0.0006
IRM	1	0.0019 ± 0.0006	0.0010 ± 0.0002	0.0027 ± 0.0012
ERM	3	0.0010 ± 0.0004	0.0003 ± 0.0001	0.0017 ± 0.0007
IRM	3	0.0012 ± 0.0005	0.0004 ± 0.0001	0.0020 ± 0.0010
ERM	4	0.0010 ± 0.0004	0.0002 ± 0.0001	0.0017 ± 0.0007
IRM	4	0.0011 ± 0.0005	0.0003 ± 0.0001	0.0019 ± 0.0010
ERM	5	0.0009 ± 0.0004	0.0002 ± 0.0001	0.0017 ± 0.0007
IRM	5	0.0010 ± 0.0005	0.0003 ± 0.0001	0.0018 ± 0.0009
ERM	8	0.0009 ± 0.0004	0.0002 ± 0.0001	0.0016 ± 0.0007
IRM	8	0.0010 ± 0.0005	0.0002 ± 0.0001	0.0017 ± 0.0009
ERM	10	0.0009 ± 0.0004	0.0002 ± 0.0001	0.0016 ± 0.0007
IRM	10	0.0010 ± 0.0005	0.0002 ± 0.0001	0.0017 ± 0.0009
ERM	15	0.0009 ± 0.0004	0.0002 ± 0.0001	0.0016 ± 0.0007
IRM	15	0.0009 ± 0.0004	0.0002 ± 0.0001	0.0016 ± 0.0008

Table 9: The results of adapting the values of the training environments under homoskedastic Y-noise with CF-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.0098 ± 0.0002	0.0098 ± 0.0004	0.0097 ± 0.0002
IRM	0.2	0.0030 ± 0.0008	0.0035 ± 0.0012	0.0025 ± 0.0009
ERM	0.5	0.0075 ± 0.0002	0.0076 ± 0.0003	0.0075 ± 0.0002
IRM	0.5	0.0025 ± 0.0006	0.0027 ± 0.0009	0.0023 ± 0.0007
ERM	1	0.0049 ± 0.0001	0.0050 ± 0.0002	0.0049 ± 0.0001
IRM	1	0.0025 ± 0.0005	0.0025 ± 0.0006	0.0026 ± 0.0007
ERM	3	0.0012 ± 0.0000	0.0012 ± 0.0001	0.0011 ± 0.0001
IRM	3	0.0027 ± 0.0002	0.0027 ± 0.0003	0.0027 ± 0.0002
ERM	4	0.0007 ± 0.0000	0.0007 ± 0.0000	0.0006 ± 0.0000
IRM	4	0.0014 ± 0.0001	0.0014 ± 0.0002	0.0013 ± 0.0001
ERM	5	0.0004 ± 0.0000	0.0004 ± 0.0000	0.0004 ± 0.0000
IRM	5	0.0007 ± 0.0001	0.0008 ± 0.0001	0.0007 ± 0.0001
ERM	8	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
IRM	8	0.0002 ± 0.0000	0.0002 ± 0.0000	0.0002 ± 0.0000
ERM	10	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
IRM	10	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
ERM	15	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	15	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000

Table 10: The results of adapting the values of the training environments under homoskedastic Y-noise with AC-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.2	0.0188 ± 0.0020	0.0187 ± 0.0029	0.0188 ± 0.0030
IRM	0.2	0.0108 ± 0.0015	0.0103 ± 0.0021	0.0114 ± 0.0023
ERM	0.5	0.0163 ± 0.0019	0.0163 ± 0.0027	0.0164 ± 0.0028
IRM	0.5	0.0119 ± 0.0015	0.0114 ± 0.0019	0.0124 ± 0.0023
ERM	1	0.0134 ± 0.0017	0.0134 ± 0.0025	0.0134 ± 0.0025
IRM	1	0.0153 ± 0.0022	0.0149 ± 0.0031	0.0157 ± 0.0033
ERM	3	0.0085 ± 0.0013	0.0085 ± 0.0019	0.0085 ± 0.0019
IRM	3	0.0111 ± 0.0013	0.0110 ± 0.0019	0.0112 ± 0.0021
ERM	4	0.0077 ± 0.0012	0.0077 ± 0.0018	0.0077 ± 0.0018
IRM	4	0.0091 ± 0.0012	0.0090 ± 0.0017	0.0091 ± 0.0018
ERM	5	0.0072 ± 0.0012	0.0072 ± 0.0017	0.0071 ± 0.0017
IRM	5	0.0080 ± 0.0011	0.0080 ± 0.0016	0.0080 ± 0.0017
ERM	8	0.0064 ± 0.0011	0.0065 ± 0.0016	0.0064 ± 0.0016
IRM	8	0.0068 ± 0.0011	0.0068 ± 0.0015	0.0068 ± 0.0016
ERM	10	0.0062 ± 0.0011	0.0063 ± 0.0016	0.0062 ± 0.0016
IRM	10	0.0065 ± 0.0010	0.0065 ± 0.0015	0.0064 ± 0.0016
ERM	15	0.0060 ± 0.0010	0.0060 ± 0.0015	0.0060 ± 0.0015
IRM	15	0.0062 ± 0.0010	0.0062 ± 0.0015	0.0061 ± 0.0015

Table 11: The results of adapting the values of the training environments under homoskedastic Y-noise with HB-regression.

C.2 Deviation Between Training Environments

C.2.1 Heteroskedastic Y-noise Tables

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.0010 ± 0.0003	0.0019 ± 0.0003	0.0001 ± 0.0000
IRM	0.1	0.0019 ± 0.0006	0.0036 ± 0.0008	0.0001 ± 0.0000
ERM	0.5	0.0018 ± 0.0003	0.0025 ± 0.0005	0.0011 ± 0.0001
IRM	0.5	0.0017 ± 0.0004	0.0031 ± 0.0004	0.0003 ± 0.0001
ERM	1	0.0031 ± 0.0004	0.0026 ± 0.0005	0.0036 ± 0.0005
IRM	1	0.0023 ± 0.0005	0.0039 ± 0.0005	0.0008 ± 0.0003
ERM	2	0.0079 ± 0.0015	0.0027 ± 0.0005	0.0130 ± 0.0018
IRM	2	0.0035 ± 0.0007	0.0038 ± 0.0007	0.0032 ± 0.0011
ERM	3	0.0156 ± 0.0035	0.0027 ± 0.0005	0.0284 ± 0.0038
IRM	3	0.0056 ± 0.0014	0.0038 ± 0.0007	0.0074 ± 0.0026
ERM	4	0.0262 ± 0.0063	0.0028 ± 0.0005	0.0496 ± 0.0067
IRM	4	0.0079 ± 0.0023	0.0038 ± 0.0007	0.0121 ± 0.0043
ERM	5	0.0397 ± 0.0099	0.0028 ± 0.0005	0.0767 ± 0.0104
IRM	5	0.0107 ± 0.0035	0.0038 ± 0.0007	0.0176 ± 0.0064
ERM	8	0.0981 ± 0.0253	0.0028 ± 0.0005	0.1934 ± 0.0262
IRM	8	0.0207 ± 0.0077	0.0038 ± 0.0007	0.0375 ± 0.0136
ERM	10	0.1517 ± 0.0395	0.0028 ± 0.0005	0.3007 ± 0.0406
IRM	10	0.0280 ± 0.0107	0.0038 ± 0.0007	0.0523 ± 0.0188
ERM	15	0.3374 ± 0.0886	0.0028 ± 0.0005	0.6720 ± 0.0908
IRM	15	0.0474 ± 0.0185	0.0038 ± 0.0007	0.0910 ± 0.0320

Table 12: The results of adapting the deviation between the training environments under heteroskedastic Y-noise with CS-regression. This corresponds to figure 6.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.0010 ± 0.0003	0.0018 ± 0.0004	0.0002 ± 0.0000
IRM	0.1	0.0016 ± 0.0005	0.0031 ± 0.0006	0.0002 ± 0.0000
ERM	0.5	0.0021 ± 0.0003	0.0022 ± 0.0005	0.0021 ± 0.0003
IRM	0.5	0.0025 ± 0.0007	0.0044 ± 0.0012	0.0007 ± 0.0002
ERM	1	0.0062 ± 0.0014	0.0023 ± 0.0005	0.0102 ± 0.0020
IRM	1	0.0037 ± 0.0008	0.0049 ± 0.0011	0.0026 ± 0.0009
ERM	2	0.0278 ± 0.0078	0.0023 ± 0.0005	0.0533 ± 0.0105
IRM	2	0.0099 ± 0.0024	0.0060 ± 0.0016	0.0139 ± 0.0043
ERM	3	0.0656 ± 0.0187	0.0023 ± 0.0005	0.1288 ± 0.0241
IRM	3	0.0216 ± 0.0056	0.0063 ± 0.0017	0.0369 ± 0.0087
ERM	4	0.1170 ± 0.0337	0.0023 ± 0.0005	0.2318 ± 0.0433
IRM	4	0.0362 ± 0.0095	0.0064 ± 0.0018	0.0661 ± 0.0136
ERM	5	0.1813 ± 0.0530	0.0022 ± 0.0005	0.3604 ± 0.0688
IRM	5	0.0513 ± 0.0141	0.0064 ± 0.0018	0.0961 ± 0.0197
ERM	8	0.4448 ± 0.1390	0.0022 ± 0.0005	0.8873 ± 0.1950
IRM	8	0.1057 ± 0.0286	0.0065 ± 0.0018	0.2048 ± 0.0357
ERM	10	0.6691 ± 0.2220	0.0022 ± 0.0004	1.3359 ± 0.3305
IRM	10	0.1450 ± 0.0391	0.0065 ± 0.0018	0.2835 ± 0.0469
ERM	15	1.3224 ± 0.4932	0.0022 ± 0.0004	2.6427 ± 0.7998
IRM	15	0.2432 ± 0.0667	0.0065 ± 0.0018	0.4800 ± 0.0795

Table 13: The results of adapting the deviation between the training environments under heteroskedastic Y-noise with CF-regression. This corresponds to figure 6.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.0090 ± 0.0005	0.0097 ± 0.0010	0.0082 ± 0.0002
IRM	0.1	0.0049 ± 0.0005	0.0059 ± 0.0009	0.0040 ± 0.0002
ERM	0.5	0.1589 ± 0.0024	0.1593 ± 0.0047	0.1586 ± 0.0012
IRM	0.5	0.0363 ± 0.0020	0.0296 ± 0.0027	0.0431 ± 0.0007
ERM	1	0.4588 ± 0.0036	0.4588 ± 0.0070	0.4588 ± 0.0024
IRM	1	0.1189 ± 0.0079	0.0854 ± 0.0036	0.1524 ± 0.0011
ERM	2	0.7752 ± 0.0031	0.7752 ± 0.0059	0.7753 ± 0.0024
IRM	2	0.3268 ± 0.0169	0.2538 ± 0.0042	0.3999 ± 0.0028
ERM	3	0.8841 ± 0.0023	0.8840 ± 0.0044	0.8841 ± 0.0019
IRM	3	0.5135 ± 0.0180	0.4360 ± 0.0045	0.5910 ± 0.0035
ERM	4	0.9301 ± 0.0018	0.9301 ± 0.0034	0.9302 ± 0.0015
IRM	4	0.6957 ± 0.0204	0.6400 ± 0.0282	0.7513 ± 0.0168
ERM	5	0.9534 ± 0.0015	0.9534 ± 0.0028	0.9534 ± 0.0013
IRM	5	0.9134 ± 0.0035	0.9088 ± 0.0059	0.9181 ± 0.0035
ERM	8	0.9804 ± 0.0009	0.9804 ± 0.0017	0.9805 ± 0.0008
IRM	8	0.9688 ± 0.0024	0.9681 ± 0.0043	0.9695 ± 0.0024
ERM	10	0.9871 ± 0.0008	0.9871 ± 0.0014	0.9871 ± 0.0007
IRM	10	0.9800 ± 0.0020	0.9796 ± 0.0035	0.9804 ± 0.0020
ERM	15	0.9939 ± 0.0005	0.9939 ± 0.0009	0.9939 ± 0.0005
IRM	15	0.9910 ± 0.0013	0.9909 ± 0.0024	0.9911 ± 0.0013

Table 14: The results of adapting the deviation between the training environments under heteroskedastic Y-noise with AC-regression. This corresponds to figure 6.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.0087 ± 0.0004	0.0090 ± 0.0006	0.0083 ± 0.0003
IRM	0.1	0.0054 ± 0.0005	0.0067 ± 0.0009	0.0042 ± 0.0002
ERM	0.5	0.1583 ± 0.0026	0.1565 ± 0.0039	0.1600 ± 0.0035
IRM	0.5	0.0381 ± 0.0017	0.0322 ± 0.0019	0.0440 ± 0.0009
ERM	1	0.4468 ± 0.0048	0.4441 ± 0.0071	0.4495 ± 0.0068
IRM	1	0.1193 ± 0.0077	0.0864 ± 0.0022	0.1523 ± 0.0022
ERM	2	0.7373 ± 0.0071	0.7354 ± 0.0103	0.7393 ± 0.0102
IRM	2	0.3203 ± 0.0166	0.2493 ± 0.0039	0.3913 ± 0.0047
ERM	3	0.8339 ± 0.0081	0.8327 ± 0.0117	0.8351 ± 0.0118
IRM	3	0.4975 ± 0.0174	0.4241 ± 0.0057	0.5710 ± 0.0068
ERM	4	0.8741 ± 0.0086	0.8733 ± 0.0124	0.8749 ± 0.0126
IRM	4	0.7200 ± 0.0203	0.6847 ± 0.0338	0.7554 ± 0.0180
ERM	5	0.8941 ± 0.0088	0.8936 ± 0.0127	0.8946 ± 0.0130
IRM	5	0.8616 ± 0.0079	0.8589 ± 0.0118	0.8644 ± 0.0111
ERM	8	0.9171 ± 0.0091	0.9171 ± 0.0131	0.9172 ± 0.0134
IRM	8	0.9106 ± 0.0088	0.9114 ± 0.0132	0.9098 ± 0.0124
ERM	10	0.9227 ± 0.0092	0.9228 ± 0.0131	0.9226 ± 0.0135
IRM	10	0.9209 ± 0.0091	0.9220 ± 0.0136	0.9197 ± 0.0127
ERM	15	0.9282 ± 0.0092	0.9285 ± 0.0132	0.9279 ± 0.0136
IRM	15	0.9312 ± 0.0093	0.9328 ± 0.0140	0.9295 ± 0.0131

Table 15: The results of adapting the deviation between the training environments under heteroskedastic Y-noise with HB-regression. This corresponds to figure 6.

C.2.2 Homoskedastic Y-noise Figures

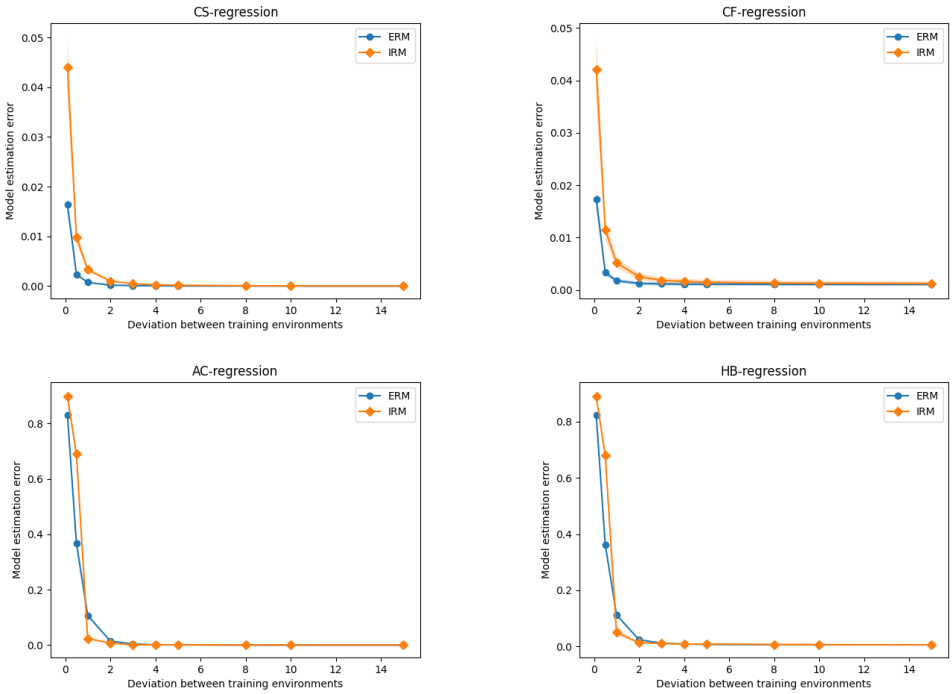


Figure 13: The results of adapting the deviation between the training environments under homoskedastic Y-noise.

C.2.3 Homoskedastic Y-noise Tables

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.0164 ± 0.0016	0.0171 ± 0.0027	0.0156 ± 0.0019
IRM	0.1	0.0441 ± 0.0063	0.0452 ± 0.0087	0.0429 ± 0.0097
ERM	0.5	0.0024 ± 0.0002	0.0022 ± 0.0004	0.0025 ± 0.0003
IRM	0.5	0.0098 ± 0.0014	0.0085 ± 0.0017	0.0111 ± 0.0023
ERM	1	0.0007 ± 0.0001	0.0007 ± 0.0001	0.0008 ± 0.0001
IRM	1	0.0033 ± 0.0005	0.0026 ± 0.0005	0.0040 ± 0.0008
ERM	2	0.0002 ± 0.0000	0.0002 ± 0.0000	0.0002 ± 0.0000
IRM	2	0.0010 ± 0.0001	0.0008 ± 0.0001	0.0012 ± 0.0002
ERM	3	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
IRM	3	0.0005 ± 0.0001	0.0004 ± 0.0001	0.0006 ± 0.0001
ERM	4	0.0001 ± 0.0000	0.0000 ± 0.0000	0.0001 ± 0.0000
IRM	4	0.0003 ± 0.0000	0.0002 ± 0.0000	0.0003 ± 0.0001
ERM	5	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	5	0.0002 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000
ERM	8	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	8	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
ERM	10	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	10	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0001 ± 0.0000
ERM	15	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	15	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000

Table 16: The results of adapting the deviation between the training environments under homoskedastic Y-noise with CS-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.0174 ± 0.0022	0.0154 ± 0.0030	0.0193 ± 0.0032
IRM	0.1	0.0421 ± 0.0065	0.0445 ± 0.0112	0.0397 ± 0.0073
ERM	0.5	0.0033 ± 0.0005	0.0022 ± 0.0005	0.0045 ± 0.0007
IRM	0.5	0.0115 ± 0.0018	0.0111 ± 0.0027	0.0119 ± 0.0025
ERM	1	0.0018 ± 0.0004	0.0008 ± 0.0002	0.0027 ± 0.0006
IRM	1	0.0052 ± 0.0009	0.0042 ± 0.0011	0.0063 ± 0.0015
ERM	2	0.0013 ± 0.0003	0.0004 ± 0.0001	0.0022 ± 0.0005
IRM	2	0.0025 ± 0.0006	0.0014 ± 0.0004	0.0036 ± 0.0010
ERM	3	0.0012 ± 0.0003	0.0003 ± 0.0001	0.0020 ± 0.0005
IRM	3	0.0019 ± 0.0005	0.0008 ± 0.0002	0.0030 ± 0.0009
ERM	4	0.0011 ± 0.0003	0.0003 ± 0.0001	0.0020 ± 0.0005
IRM	4	0.0016 ± 0.0005	0.0006 ± 0.0002	0.0027 ± 0.0009
ERM	5	0.0011 ± 0.0003	0.0002 ± 0.0001	0.0020 ± 0.0005
IRM	5	0.0015 ± 0.0005	0.0004 ± 0.0001	0.0025 ± 0.0008
ERM	8	0.0011 ± 0.0003	0.0002 ± 0.0001	0.0020 ± 0.0005
IRM	8	0.0013 ± 0.0004	0.0003 ± 0.0001	0.0023 ± 0.0008
ERM	10	0.0011 ± 0.0003	0.0002 ± 0.0001	0.0020 ± 0.0005
IRM	10	0.0013 ± 0.0004	0.0003 ± 0.0001	0.0023 ± 0.0008
ERM	15	0.0011 ± 0.0003	0.0002 ± 0.0001	0.0020 ± 0.0005
IRM	15	0.0013 ± 0.0004	0.0003 ± 0.0001	0.0022 ± 0.0007

Table 17: The results of adapting the deviation between the training environments under homoskedastic Y-noise with CF-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.8317 ± 0.0066	0.8325 ± 0.0132	0.8309 ± 0.0028
IRM	0.1	0.8975 ± 0.0072	0.9115 ± 0.0130	0.8835 ± 0.0031
ERM	0.5	0.3678 ± 0.0045	0.3678 ± 0.0088	0.3678 ± 0.0025
IRM	0.5	0.6891 ± 0.0201	0.7663 ± 0.0183	0.6119 ± 0.0072
ERM	1	0.1068 ± 0.0015	0.1068 ± 0.0030	0.1068 ± 0.0009
IRM	1	0.0236 ± 0.0022	0.0213 ± 0.0034	0.0260 ± 0.0028
ERM	2	0.0148 ± 0.0003	0.0148 ± 0.0005	0.0148 ± 0.0002
IRM	2	0.0072 ± 0.0010	0.0070 ± 0.0014	0.0075 ± 0.0015
ERM	3	0.0038 ± 0.0001	0.0038 ± 0.0002	0.0038 ± 0.0001
IRM	3	0.0009 ± 0.0002	0.0011 ± 0.0003	0.0007 ± 0.0001
ERM	4	0.0014 ± 0.0000	0.0014 ± 0.0001	0.0014 ± 0.0000
IRM	4	0.0011 ± 0.0001	0.0011 ± 0.0002	0.0010 ± 0.0002
ERM	5	0.0006 ± 0.0000	0.0006 ± 0.0000	0.0006 ± 0.0000
IRM	5	0.0015 ± 0.0001	0.0015 ± 0.0002	0.0015 ± 0.0001
ERM	8	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
IRM	8	0.0003 ± 0.0000	0.0003 ± 0.0000	0.0003 ± 0.0000
ERM	10	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000
IRM	10	0.0002 ± 0.0000	0.0002 ± 0.0000	0.0002 ± 0.0000
ERM	15	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
IRM	15	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0000 ± 0.0000

Table 18: The results of adapting the deviation between the training environments under homoskedastic Y-noise with AC-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	0.1	0.8221 ± 0.0044	0.8136 ± 0.0071	0.8305 ± 0.0038
IRM	0.1	0.8887 ± 0.0078	0.8963 ± 0.0154	0.8811 ± 0.0032
ERM	0.5	0.3613 ± 0.0037	0.3573 ± 0.0048	0.3654 ± 0.0054
IRM	0.5	0.6791 ± 0.0191	0.7567 ± 0.0126	0.6016 ± 0.0071
ERM	1	0.1122 ± 0.0028	0.1108 ± 0.0038	0.1135 ± 0.0042
IRM	1	0.0506 ± 0.0111	0.0497 ± 0.0167	0.0515 ± 0.0155
ERM	2	0.0240 ± 0.0017	0.0238 ± 0.0025	0.0242 ± 0.0025
IRM	2	0.0133 ± 0.0014	0.0127 ± 0.0017	0.0140 ± 0.0022
ERM	3	0.0119 ± 0.0013	0.0119 ± 0.0019	0.0120 ± 0.0019
IRM	3	0.0110 ± 0.0015	0.0108 ± 0.0020	0.0113 ± 0.0023
ERM	4	0.0087 ± 0.0011	0.0087 ± 0.0016	0.0088 ± 0.0016
IRM	4	0.0084 ± 0.0010	0.0083 ± 0.0014	0.0085 ± 0.0015
ERM	5	0.0075 ± 0.0010	0.0075 ± 0.0015	0.0076 ± 0.0015
IRM	5	0.0092 ± 0.0010	0.0092 ± 0.0013	0.0092 ± 0.0015
ERM	8	0.0064 ± 0.0009	0.0064 ± 0.0013	0.0064 ± 0.0013
IRM	8	0.0071 ± 0.0009	0.0071 ± 0.0012	0.0071 ± 0.0013
ERM	10	0.0062 ± 0.0009	0.0061 ± 0.0013	0.0062 ± 0.0013
IRM	10	0.0067 ± 0.0008	0.0067 ± 0.0012	0.0067 ± 0.0013
ERM	15	0.0060 ± 0.0009	0.0059 ± 0.0013	0.0060 ± 0.0013
IRM	15	0.0063 ± 0.0008	0.0063 ± 0.0011	0.0062 ± 0.0012

Table 19: The results of adapting the deviation between the training environments under homoskedastic Y-noise with HB-regression.

D Additional Experiments

D.1 Sample Complexity

This experiment takes the perspective of sample complexity similar to [7]. We run the experiment for every data distribution shift with the number of samples per training environment ranging from 50 to 2000.

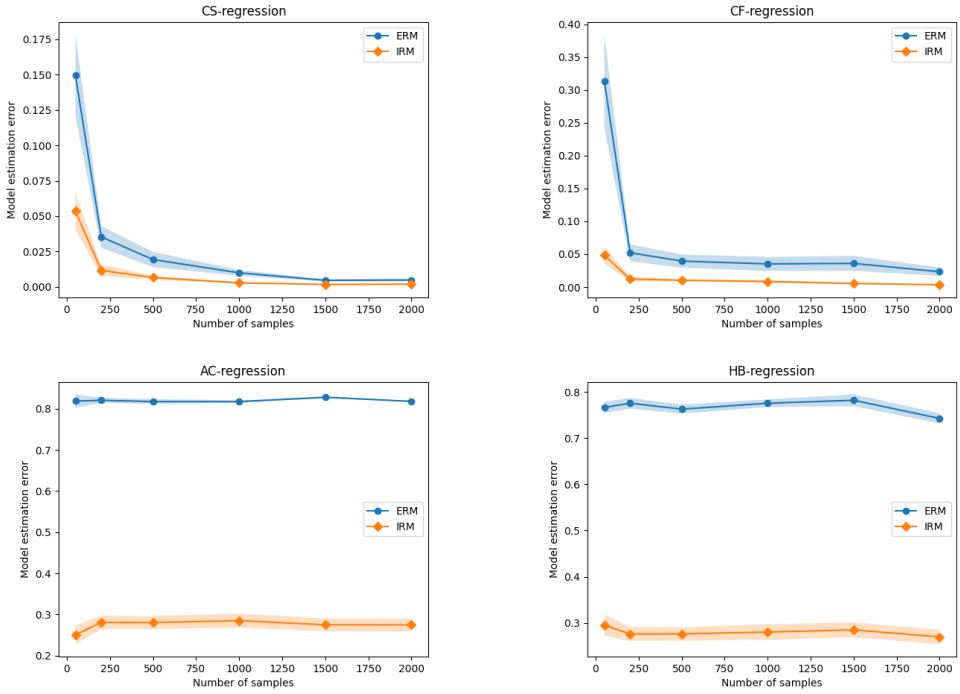


Figure 14: The results of the sample complexity experiment.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	50	0.1495 ± 0.0289	0.0628 ± 0.0161	0.2362 ± 0.0377
IRM	50	0.0535 ± 0.0128	0.0867 ± 0.0180	0.0202 ± 0.0099
ERM	200	0.0353 ± 0.0077	0.0131 ± 0.0021	0.0576 ± 0.0110
IRM	200	0.0117 ± 0.0041	0.0206 ± 0.0071	0.0028 ± 0.0011
ERM	500	0.0193 ± 0.0054	0.0069 ± 0.0013	0.0317 ± 0.0091
IRM	500	0.0066 ± 0.0013	0.0084 ± 0.0013	0.0048 ± 0.0022
ERM	1000	0.0099 ± 0.0021	0.0032 ± 0.0007	0.0166 ± 0.0025
IRM	1000	0.0028 ± 0.0005	0.0042 ± 0.0006	0.0015 ± 0.0006
ERM	1500	0.0046 ± 0.0008	0.0032 ± 0.0005	0.0060 ± 0.0014
IRM	1500	0.0017 ± 0.0005	0.0029 ± 0.0006	0.0006 ± 0.0005
ERM	2000	0.0048 ± 0.0011	0.0017 ± 0.0004	0.0080 ± 0.0016
IRM	2000	0.0019 ± 0.0004	0.0017 ± 0.0004	0.0022 ± 0.0006

Table 20: Results of the sample complexity experiment for CS-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	50	0.3139 ± 0.0706	0.0817 ± 0.0147	0.5461 ± 0.0864
IRM	50	0.0485 ± 0.0126	0.0878 ± 0.0163	0.0093 ± 0.0053
ERM	200	0.0522 ± 0.0128	0.0159 ± 0.0038	0.0885 ± 0.0188
IRM	200	0.0123 ± 0.0037	0.0198 ± 0.0060	0.0047 ± 0.0031
ERM	500	0.0396 ± 0.0102	0.0082 ± 0.0015	0.0709 ± 0.0138
IRM	500	0.0102 ± 0.0016	0.0102 ± 0.0020	0.0103 ± 0.0026
ERM	1000	0.0352 ± 0.0106	0.0024 ± 0.0005	0.0680 ± 0.0146
IRM	1000	0.0084 ± 0.0020	0.0065 ± 0.0016	0.0104 ± 0.0037
ERM	1500	0.0360 ± 0.0115	0.0026 ± 0.0006	0.0694 ± 0.0167
IRM	1500	0.0054 ± 0.0011	0.0037 ± 0.0007	0.0071 ± 0.0019
ERM	2000	0.0233 ± 0.0063	0.0034 ± 0.0006	0.0432 ± 0.0083
IRM	2000	0.0036 ± 0.0006	0.0022 ± 0.0002	0.0050 ± 0.0010

Table 21: Results of the sample complexity experiment for CF-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	50	0.8186 ± 0.0164	0.8207 ± 0.0257	0.8166 ± 0.0220
IRM	50	0.2504 ± 0.0227	0.1972 ± 0.0255	0.3035 ± 0.0290
ERM	200	0.8206 ± 0.0061	0.8175 ± 0.0088	0.8236 ± 0.0088
IRM	200	0.2808 ± 0.0159	0.2193 ± 0.0093	0.3423 ± 0.0069
ERM	500	0.8174 ± 0.0061	0.8218 ± 0.0103	0.8131 ± 0.0067
IRM	500	0.2803 ± 0.0158	0.2194 ± 0.0097	0.3412 ± 0.0066
ERM	1000	0.8173 ± 0.0031	0.8176 ± 0.0058	0.8169 ± 0.0024
IRM	1000	0.2849 ± 0.0169	0.2162 ± 0.0045	0.3537 ± 0.0028
ERM	1500	0.8279 ± 0.0020	0.8306 ± 0.0035	0.8253 ± 0.0017
IRM	1500	0.2749 ± 0.0155	0.2120 ± 0.0028	0.3378 ± 0.0042
ERM	2000	0.8179 ± 0.0013	0.8169 ± 0.0021	0.8190 ± 0.0016
IRM	2000	0.2746 ± 0.0162	0.2089 ± 0.0040	0.3403 ± 0.0045

Table 22: Results of the sample complexity experiment for AC-regression.

Method	Number of samples	MER Average	MER Causal	MER Non-causal
ERM	50	0.7663 ± 0.0122	0.7633 ± 0.0160	0.7694 ± 0.0194
IRM	50	0.2948 ± 0.0225	0.2696 ± 0.0214	0.3200 ± 0.0392
ERM	200	0.7755 ± 0.0117	0.7730 ± 0.0181	0.7780 ± 0.0159
IRM	200	0.2760 ± 0.0150	0.2315 ± 0.0163	0.3204 ± 0.0142
ERM	500	0.7626 ± 0.0098	0.7646 ± 0.0137	0.7607 ± 0.0148
IRM	500	0.2759 ± 0.0145	0.2219 ± 0.0087	0.3300 ± 0.0093
ERM	1000	0.7754 ± 0.0083	0.7745 ± 0.0122	0.7763 ± 0.0122
IRM	1000	0.2801 ± 0.0166	0.2126 ± 0.0038	0.3475 ± 0.0047
ERM	1500	0.7819 ± 0.0128	0.7825 ± 0.0189	0.7813 ± 0.0185
IRM	1500	0.2848 ± 0.0161	0.2207 ± 0.0053	0.3489 ± 0.0066
ERM	2000	0.7426 ± 0.0110	0.7401 ± 0.0172	0.7450 ± 0.0148
IRM	2000	0.2695 ± 0.0158	0.2062 ± 0.0049	0.3328 ± 0.0059

Table 23: Results of the sample complexity experiment for HB-regression.

D.2 Scrambling of the Observable Features Experiment

A real-world phenomenon is the presence of noise in observations. Think of distortion in an image. This can be simulated by scrambling the observable features. For that purpose, we map X^e to $S \cdot X^e$ where S is an orthogonal matrix with random values.

It turns out that this scrambling destroys any chance of learning invariant relationships, which is why we decided not to include in the paper.

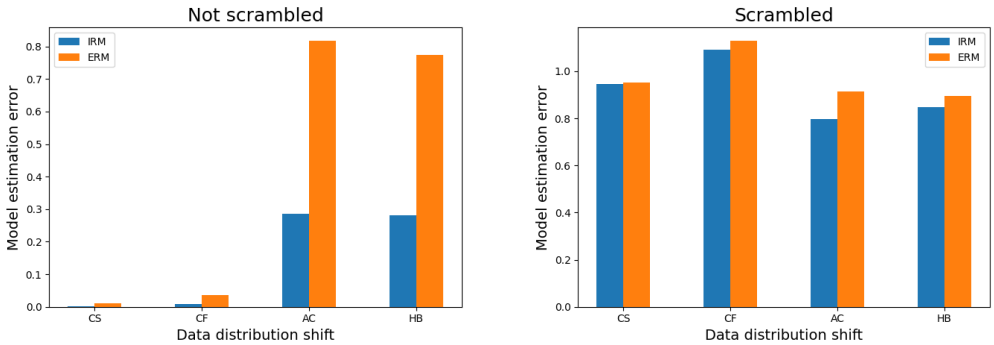


Figure 15: The results of the scrambling of the observable features experiment.