
Sparse Invariant Risk Minimization

Xiao Zhou^{*1} Yong Lin^{*1} Weizhong Zhang^{*1} Tong Zhang¹²

Abstract

Invariant Risk Minimization (IRM) is an emerging invariant feature extracting technique to help generalization with distributional shift. However, we find that there exists a basic and intractable contradiction between the model trainability and generalization ability in IRM. On one hand, recent studies on deep learning theory indicate the importance of large-sized or even overparameterized neural networks to make the model easy to train. On the other hand, unlike empirical risk minimization that can be benefited from overparameterization, our empirical and theoretical analyses show that the generalization ability of IRM is much easier to be demolished by overfitting caused by overparameterization. In this paper, we propose a simple yet effective paradigm named Sparse Invariant Risk Minimization (**SparseIRM**) to address this contradiction. Our key idea is to employ a global sparsity constraint as a defense to prevent spurious features from leaking in **during the whole IRM process**. Compared with sparsify-after-training prototype by prior work which can discard invariant features, the global sparsity constraint limits the budget for feature selection and enforces SparseIRM to select the invariant features. We illustrate the benefit of SparseIRM through a theoretical analysis on a simple linear case. Empirically we demonstrate the power of SparseIRM through various datasets and models and surpass state-of-the-art methods with a gap up to 29%.

1. Introduction

In the last decade, deep neural networks (DNNs) have achieved unprecedented successes in numerous applications, including but not limited to computer vision (He et al., 2016;

^{*}Equal contribution ¹The Hong Kong University of Science and Technology ²Google Research. Correspondence to: Tong Zhang <tongzhang@tongzhang-ml.org>.

Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Sun et al., 2014) and natural language processing (Bahdanau et al., 2014; Luong et al., 2015). Most deep learning models are trained by the Empirical Risk Minimization (ERM) paradigm, under the I.I.D. assumption that the training and testing samples are *independently* drawn from an *identical distribution*. However, more and more failure cases of DNNs (Beery et al., 2018; Geirhos et al., 2020; DeGrave et al., 2021; Zhang et al., 2022b) are reported in the latest studies where the I.I.D. assumption is violated in application due to distributional shifts.

IRM (Arjovsky et al., 2019; Creager et al., 2021; Krueger et al., 2021; Xie et al., 2020; Chang et al., 2020; Zhang et al., 2021c;b; Jin et al., 2020) is an emerging learning paradigm to enable the generalization with distributional shifts. The key idea of IRM is to learn an invariant feature representation on the datasets drawn from multiple environments, in the sense that based on this representation one should be able to learn a common classifier working well in all these environments. Due to the consistent good performance achieved in these existing environments, the generalization in new environments with unseen distributional shifts can be expected, which has been verified by the promising empirical results (Arjovsky et al., 2019; Krueger et al., 2021; Xie et al., 2020; Chang et al., 2020; Jin et al., 2020). However, IRM is found to be less effective when applied to deep models (Gulrajani & Lopez-Paz, 2020; Lin et al., 2021). In this work, we argue that the reason is that there exists a basic and intractable contradiction between the model trainability and generalization ability in the IRM paradigm:

- On one hand, to make the model easy to train, we should use large-sized or even overparameterized DNNs (Zhang et al., 2021a), i.e., networks with massive neurons. Empirically, it is observed that such DNNs are easy to train, while small-sized ones could easily get stuck at bad local minima, that’s why modern DNNs are always overparameterized. Theoretically, recent studies on deep learning theory show that DNNs behave like convex systems and their loss landscapes become smoother when the number of the neurons goes to infinity (Gu et al., 2020; Mei et al., 2018).
- On the other hand, when overparameterized, the gener-

alization ability of IRM is very easy to be demolished by over-fitting because of mistakenly using some spurious features. We theoretically proved that, when overparameterized, unlike that ERM can have good or better generalization (Zhang et al., 2021a), IRM could fail even in a simple linear case. It can be expected that IRM will collapse in more overparameterized deep neural networks because there are much more parameters than a simple linear model. The details can be seen Section 3.2.

In this paper, we propose a simple yet effective Sparse Invariant Risk Minimization (SparseIRM) paradigm to address contradiction above. Our key idea is to employ a global sparsity constraint as a defense to prevent spurious features from leaking into the submodel we work on **during the whole IRM process**. Compared with the sparsify-after-training prototype adopted by prior work (Zhang et al., 2021b), which can discard invariant features misled by the undetected spurious features, our paradigm successfully sets up a barrier to spurious and random features with sparsity constraint throughout training, leading to better generalization performance. Specifically, during the training process, because of our sparsity constraint, the subnetwork we work on is too small to include all the spurious and random features, as the number of these features is always significantly larger than invariant features. Therefore, the network has to identify and focus on the invariant features to minimize the loss function. We provide our understanding of this phenomenon through theoretical analysis on a simple linear case (Theorem 1). We validate the superior performance in defending overfitting through various datasets and models, and find that we surpass state-of-the-art methods on various datasets with a gap up to 29%. In addition, we perform ablation studies to verify that our method is indeed effective in removing the spurious features in Section 5.3.

The contributions and novelties of this work are summarized as follows:

- We demonstrate that when overparameterized, the generalization ability of IRM is easy to be demolished by overfitting because of mistakenly using some spurious features (see Section 3.2).
- We propose a sparse invariant risk minimization (SparseIRM) method which enforces sparsity constraint during the whole training process and illustrate its benefit through theoretical analysis in a simple linear case.(see Sections 4).
- We demonstrate the superiority of our method in overparameterized settings and show that the improvement over state-of-the-art methods can be up to 29% in accuracy (see Section 5).

Notations: Let $\|\cdot\|_1$ and $\|\cdot\|_2$ be the ℓ_1 and ℓ_2 norm of a real valued vector, respectively. We denote $\mathbf{1}^n/\mathbf{0}^n \in \mathbb{R}^n$ to be a vector with all components equal to 1/0 with length n . In addition, $\{0, 1\}^n$ is the set of n -dimensional vectors with each coordinate valued in $\{0, 1\}$. $\mathbf{u} \circ \mathbf{v}$ denotes the element-wise product between two vectors.

2. Related Work

2.1. Causality and invariance

IRM (Arjovsky et al., 2019) is proposed to learn the features invariant among different environments based on the invariance principle first raised in (Peters et al., 2016) which aims to build model on the direct cause of target. Numerous variants have been developed recently in the community. (Ahuja et al., 2020a; Jin et al., 2020) provides new perspectives by introducing game theory and regret minimization into invariant risk minimization. (Krueger et al., 2021; Xie et al., 2020; Chang et al., 2020; Xu & Jaakkola, 2021; Xu et al., 2020; 2022) propose more effective methods motivated by penalizing the variance of losses among environments, estimating the violation of invariance, improving transferability among environments, etc. (Wang et al., 2022) proposes to reweight the training samples based on influence function. (Zhou et al., 2022b) reweights the training samples to improve out-of-domain generalization. (Lin et al., 2022a) shows that IRM suffers from the overfitting problem caused by overparameterized neural networks. There are also some analyses concerning the sufficiency of invariance principle (Ahuja et al., 2021) and the relationship of IRM to out-of-distribution generalization ability based on discrepancy measures (Zhang et al., 2021c). Another line of works try to learn invariant features when explicit environment indices are not provided (Creager et al., 2021; Liu et al., 2021b;a; Lin et al., 2022b; Xu et al., 2021; Zhang et al., 2022a). (Lin et al., 2022b) proposes a mini-max framework based on auxiliary information that can provably infer the environment indexes and learn invariance.

From the theoretical perspective, (Arjovsky et al., 2019; Rosenfeld et al., 2020; Chen et al., 2021c) investigate IRM’s dependence on the environment numbers when the model is linear. (Chen et al., 2021c) proposes to take advantage of the intrinsically low dimensional structure of spurious features to identify the invariant features with logarithmic environments. (Rosenfeld et al., 2020) also studies the performance of IRM when applied to non-linear models. (Kamath et al., 2021) analyzes the success and failure cases of IRM in different scenarios and (Ahuja et al., 2020b) compares the sample complexity of IRM with ERM. In this paper we want to investigate the effectiveness of enforcing sparsity in improving generalization performance of IRM.

2.2. Sparsity in Deep Neural Networks

In the recent years, sparsity (Han et al., 2016) has been introduced into DNNs to improve the inference efficiency or reduce the model size. The key idea (Han et al., 2016; Kusupati et al., 2020; Yuan et al., 2020b;a; Lym et al., 2019; Zhou et al., 2021b; Zou et al., 2019) is to identify and remove unimportant weights from the neural networks during or after training by developing some proper pruning rules. The most typical rule is based on the weight magnitude and some others are learning based. The empirical results demonstrate that one can reduce the model size and improve the inference efficiency significantly with slight or even negligible loss on performance. This makes it possible to deploy modern DNNs on the devices with limited computational and memory budget. Most of the existing methods are developed for the neural networks trained by ERM in the I.I.D. scenarios. In this work, we will introduce sparsity into the IRM training to boost the generalization performance.

2.3. Overparameterized Deep Neural Networks

Modern neural networks are always overparameterized (He et al., 2016; Simonyan & Zisserman, 2015), that is, the number of the trainable parameters in them is always significantly larger than the training dataset size. They can be trained easily to fit a random labeling of the training data (Zhang et al., 2021a). However, in the applications of I.I.D. scenario trained with ERM, we repeatedly observed that such networks can always be trained more easily and generalize better than the small-sized networks, which can get stuck in bad local minima. That’s why the community is continuing exploring larger-sized networks. Latest theoretical studies (Gu et al., 2020; Mei et al., 2018; Jacot et al., 2021; Kawaguchi et al., 2019) investigated the above phenomenon of overparameterized neural networks and suggested that DNNs behave like convex systems and their loss landscapes become smoother when the number of the neurons goes to infinity, which makes it easy to train and be able to avoid bad local minima. In this paper, we show that unlike in the applications of I.I.D. scenario, overfitting can be a catastrophic problem of overparameterized networks in the applications where distributional shifts occur.

3. Investigating the Effects of Overparameterization on IRM

In this section, we first present the formulation of IRM and then we investigate the effects of overparameterization on the generalization of IRM.

3.1. Preliminaries

Consider a set of E environments $\mathcal{E} := \{e_1, e_2, \dots, e_E\}$ in the sample space $\mathcal{X} \times \mathcal{Y}$ with different joint distribu-

tions $\Pr^e(\mathbf{x}, \mathbf{y})$, where $e \in \mathcal{E}$, \mathcal{X} and \mathcal{Y} are the input and target spaces, respectively. Let $\mathcal{E}_{tr} \subset \mathcal{E}$ be the training environments and $\mathcal{D}^e := \{(\mathbf{x}_i^e, \mathbf{y}_i^e)\}_{i=1}^{n_e}$ be the data set drawn from $e \in \mathcal{E}_{tr}$ with n_e being data set size. Based on these training data sets, the problem is to learn a *robust* model $f(\cdot; \mathbf{w}) : \mathcal{X} \rightarrow \mathcal{Y}$, in the sense that it can predicts \mathbf{y} well when given \mathbf{x} for all $e \in \mathcal{E}$ including the unseen environments $\mathcal{E} \setminus \mathcal{E}_{tr}$, where \mathbf{w} is the parameters of f .

IRM first formulates the predictor $f(\cdot; \mathbf{w})$ as a composite function of $g(\cdot; \Phi)$ and $h(\cdot; \mathbf{v})$, i.e., $f(\cdot; \mathbf{w}) = h(g(\cdot; \Phi); \mathbf{v})$, where $\mathbf{w} = \{\mathbf{v}, \Phi\}$ are the trainable parameters. Here, $g(\cdot; \Phi) : \mathcal{X} \rightarrow \mathcal{H}$ maps \mathcal{X} to the representation space \mathcal{H} to extract invariant features among \mathcal{E}_{tr} . $h(\cdot; \mathbf{v}) : \mathcal{H} \rightarrow \mathcal{Y}$ is the classifier, which is simultaneously optimal for all training environments. Existing IRM methods learn $g(\cdot; \Phi)$ and $h(\cdot; \mathbf{v})$ by solving the following minimization problem:

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) := \sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(\mathbf{w}) + \lambda \mathcal{J}(\mathbf{w}), \quad (1)$$

where $\mathcal{R}^e(\mathbf{w}) = \frac{1}{n_e} \sum_{i=1}^{n_e} \ell(f(\mathbf{x}_i^e; \mathbf{w}), \mathbf{y}_i^e)$ and ℓ is the loss function. $\mathcal{J}(\mathbf{w})$ is the regularizer encouraging $f(\cdot; \mathbf{w})$ to be optimal in all \mathcal{E}_{tr} . Different methods can have different $\mathcal{J}(\mathbf{w})$. Two representative IRM methods IRMv1 (Arjovsky et al., 2019) and REx (Krueger et al., 2021) adopt the following two regularizers:

$$\mathcal{J}(\mathbf{w}) := \sum_{e \in \mathcal{E}_{tr}} \|\nabla_{\mathbf{v}} \mathcal{R}^e(\mathbf{w})\|_2^2, \quad (\text{J-IRMv1})$$

$$\mathcal{J}(\mathbf{w}) := \mathbb{V}[\mathcal{R}^e(\mathbf{w})], \quad (\text{J-REx})$$

where $\mathbb{V}[\mathcal{R}^e(\mathbf{w})]$ is the variance of the losses $\mathcal{R}^e(\mathbf{w})$ in \mathcal{E}_{tr} . Intuitively, to encourages $f(\cdot; \mathbf{w})$ to be simultaneously optimal, the former enforces the gradients $\nabla_{\mathbf{v}} \mathcal{R}^e(\mathbf{w})$ to be 0, and the later reduces the loss variance to be 0.

3.2. Analysis of Overparameterized IRM in a Linear Case

In this section, we will show the difficulty of IRM to learn invariant features with a overparameterized linear model. We consider a anti-causal linear data generation procedure similar to (Arjovsky et al., 2019; Ahuja et al., 2020a). We will show IRM can struggle in such a simple linear case, not to mention the case with the overparameterized deep neural networks. Different from existing theoretical analysis on IRM (Arjovsky et al., 2019; Rosenfeld et al., 2020) that assume infinite samples are accessible, we consider the case that the model is with access to limited samples. Further, we consider the existence of huge amount of random features (Gaussian noise) apart from the invariant and spurious features, which is a direct consequence of model overparameterization (Jacot et al., 2018; Sagawa et al., 2020).

[Settings]. Suppose we have two training environments, i.e., $\mathcal{E}_{tr} = \{e_1, e_2\}$ and denote \mathbf{x}^e to be the input feature of

environment $e \in \mathcal{E}_{tr}$, which is a concatenation of the invariant feature $\mathbf{x}_{inv}^e \in \mathbb{R}^{d_{inv}}$, the spurious $\mathbf{x}_s^e \in \mathbb{R}^{d_s}$ and the random feature $\mathbf{x}_r^e \in \mathbb{R}^{d_r}$, i.e., $\mathbf{x}^e := [\mathbf{x}_{inv}^e, \mathbf{x}_s^e, \mathbf{x}_r^e] \in \mathbb{R}^d$. We consider an anti-causal setting as (Arjovsky et al., 2019; Rosenfeld et al., 2020) (Noticeably, some of our results, e.g., Proposition 1 and Corollary 1, can be immediately extended to more general data generation process). The data is generated as follows:

$$\begin{aligned} y^e &= \gamma^\top \mathbf{x}_{inv}^e + \epsilon_{inv}, \\ \mathbf{x}_s^e &= y^e \mathbf{1}^s + \alpha^e \circ \epsilon_s \\ \mathbf{x}_r^e &= \epsilon_r, \end{aligned}$$

where $e \in \mathcal{E}_{tr}$, ϵ_{inv} , ϵ_s and ϵ_r are independent random noise that follows sub-Gaussian distributions with zero mean and bounded variance. The label y^e is generated from the invariant feature \mathbf{x}_{inv}^e with a fixed vector $\gamma \in \mathbb{R}^{d_{inv}}$ that is invariant in $\forall e \in \mathcal{E}_{tr}$. The spurious feature \mathbf{x}_s^e is generated from y^e by the non-invariant vectors $\alpha^e \in \mathbb{R}^{d_s}$ that depend on the environment e . More detailed settings are placed in appendix due to space limitation.

We aim to learn a linear model to predict y based on \mathbf{x} . To be precise, the predictor $f(\cdot; \mathbf{w})$ can be expressed as:

$$f(\mathbf{x}; \mathbf{w}) = (\Phi \circ \mathbf{x})^\top \mathbf{v} + b, \quad (2)$$

where $\Phi \in \{0, 1\}^{d_{inv}+d_s+d_r}$ is a binary vector to perform feature selection. $\mathbf{v} \in \mathbb{R}^{d_{inv}+d_s+d_r}$ is the parameter of the linear function on the top of Φ and \circ stands for the element-wise product operation. We also use $\Phi(\mathbf{x})$ and $\Phi \circ \mathbf{x}$ interchangeably in this example's analysis when it is clear from the context. Our analysis focuses on IRMv1 as an example, with $\mathcal{R}^e(\mathbf{w}) = \frac{1}{n_e} \sum_{i=1}^{n_e} (y_i^e - (\Phi \circ \mathbf{x}_i^e)^\top \mathbf{v} - b)^2$.

We denote the $\hat{\mathcal{L}}(\Phi)$ as loss of a given Φ when \mathbf{v} is solved optimally, $\hat{\mathcal{L}}(\Phi) := \min_{\mathbf{v}} \mathcal{L}(\mathbf{w})$.

The ideal feature selector is $\Phi_{inv} = [\mathbf{1}^{d_{inv}}, \mathbf{0}^{d_s+d_r}]$, merely selecting the invariant feature \mathbf{x}_{inv} and discarding spurious features \mathbf{x}_s and random features \mathbf{x}_r . IRM learns \mathbf{w} by minimizing $\mathcal{L}(\mathbf{w})$, therefore, it can finally find the ideal feature selector Φ_{inv} if and only if the following condition holds

$$\hat{\mathcal{L}}(\Phi_{inv}) < \hat{\mathcal{L}}(\Phi), \forall \Phi \neq \Phi_{inv}. \quad (3)$$

The proposition below shows that the above condition (3) does not hold in the following conditions:

Proposition 1. (Failure of IRM in Overparameterization Region). *If $d_{inv} + d_s + d_r > n_{e_1} + n_{e_2}$, then*

$$\hat{\mathcal{L}}(\Phi_{all}) = 0 \leq \hat{\mathcal{L}}(\Phi_{inv}), \quad (4)$$

where $\Phi_{all} = \mathbf{1}^{d_{inv}+d_s+d_r}$.

Proposition 1 demonstrates that Φ_{all} that includes all the spurious and noisy features achieves smaller or equal loss

than Φ_{inv} , which implies that IRM can not identify the invariant features. Notably, Proposition 1 does not impose any constraint on the structure of environments, which further indicates that whatever structure of environments (Arjovsky et al., 2019; Rosenfeld et al., 2020) can not rescue IRM as long as overparameterization occurs.

The following corollary demonstrates that significant overparameterization can completely demolish the generalization of IRM.

Corollary 1. (Worse Case) *If $d_s + d_r > n_{e_1} + n_{e_2}$, then*

$$\hat{\mathcal{L}}(\Phi_{all}) = \hat{\mathcal{L}}(\Phi_{sr}) = 0 \leq \hat{\mathcal{L}}(\Phi_{inv}) \quad (5)$$

where $\Phi_{sr} = [\mathbf{0}^{d_{inv}}, \mathbf{1}^{d_s+d_r}]$.

Corollary 1 shows that if the model is significantly overparameterized, e.g., we have massive bad (spurious or random) features such that $d_s + d_r > n_{e_1} + n_{e_2}$, then the objective function $\hat{\mathcal{L}}(\mathbf{w})$ can be reduced to 0 when predictor $f(\cdot, \mathbf{w})$ purely relies on the spurious and random features. This means that a totally wrong model could be learned.

[Empirical Verification]. Figure 1 presents the training and testing accuracy of ERM, IRM (i.e., IRMv1) and Oracle (ERM trained on datasets without spurious features) on ColoredMNIST. We can see that as the hidden dimension increases, the training and testing accuracy of ERM and Oracle increases steadily. However, the testing accuracy of IRM decreases while its training accuracy increases. This verifies that IRM is indeed much easier to be demolished by overfitting originated from overparameterization.

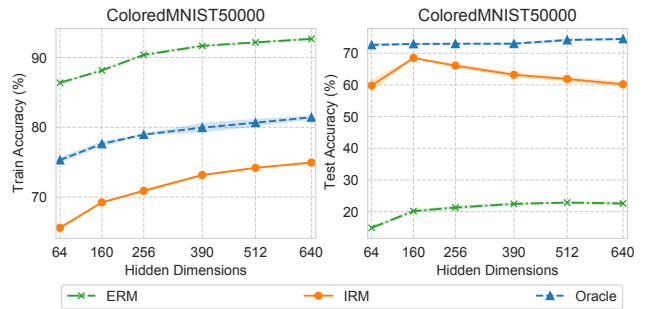


Figure 1. The overparameterization effects on IRM. The performance of IRM drops as the model sizes become larger after hidden dimension is larger than 160.

4. Sparse Invariant Risk Minimization

Below, we present our SparseIRM framework and give its theoretical properties to show its superiority.

4.1. SparseIRM Framework

Sparsity is a natural idea to promote generalization ability. The previous work (Zhang et al., 2021b) applied lottery

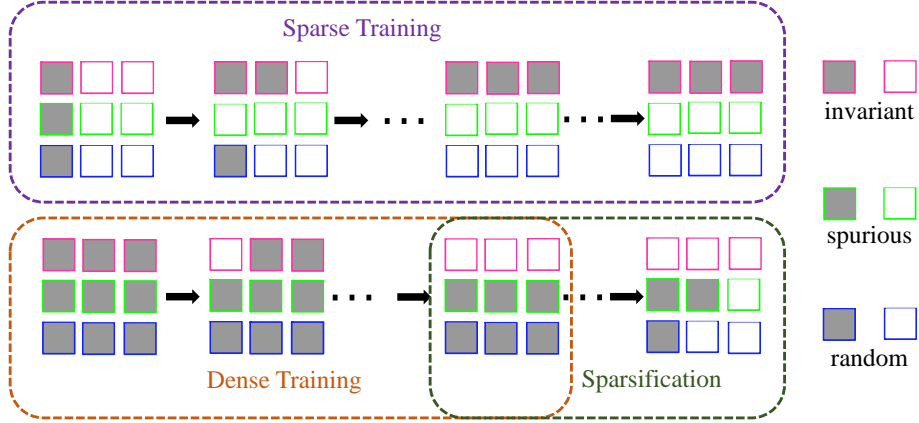


Figure 2. The flow charts of SparseIRM (top line) and the sparsify-after-training method MRM (Zhang et al., 2021b) (bottom line). The block filled with grey indicates the selected feature and the unfilled block indicates the unselected feature. Because the subnetwork we work on is too small to include all the spurious and random features due to sparsity constraint, SparseIRM is enforced to select the invariant features to minimize the loss function. MRM first performs dense training and little invariant features selected, making it difficult to recall invariant features back into the subnetwork.

ticket hypothesis on IRM and preliminarily verified the success of pruning-after-training in improving the generalization ability of IRM in some cases, while earlier discussions in Proposition 1 stresses the catastrophic pitfalls under the overparameterized settings. That is, $\Phi = [\mathbf{0}^{d_{inv}}, \mathbf{1}^{d_s+d_r}]$ after fully trained. In this scenario, the pruning methods (e.g., weight magnitude based rules (Han et al., 2016)) would discard the component $\mathbf{0}^{d_{inv}}$ in Φ permanently and will continually work on the model consisting of only spurious or random features, which is illustrated in the bottom line of Fig 2. Finding lottery tickets over such structure will only lead to failure in achieving any performance boost in generalization.

To avoid such catastrophic pitfalls, we promote our SparseIRM framework. Its key idea is to employ a sparsity constraint during the whole training process as a defense to prevent the spurious and random features from leaking into the subnetwork we work on. That is, in our framework, we concurrently perform invariant risk minimization and sparse training at the same time (Fig 2). Intuitively, during the training process, because of the sparsity constraint, the subnetwork we work on is too small to include all the spurious and random features, as the number of these features is always significantly larger than invariant features. Therefore, to achieve smaller loss, the network has to identify and focus on the invariant features. We adopt the latest state-of-the-art sparse training method to solve our sparse invariant risk minimization problem. In fact, our SparseIRM framework can be integrated with most sparse training methods flexibly, and the specific choice is not the main contribution of this work.

To be precise, we first formulate our sparse invariant risk

minimization problem as follows:

$$\min_{\mathbf{w}, \mathbf{m}} \mathcal{L}(\{\mathbf{v}, \mathbf{m} \circ \Phi\}) \quad (6)$$

$$s.t. \mathbf{w} \in \mathbb{R}^{d_w}, \mathbf{m} \in \{0, 1\}^{d_\Phi}, \|\mathbf{m}\|_1 \leq K,$$

where we associate each Φ_i with a binary mask m_i , K is used to control the total model size. d_w and d_Φ are the dimensions of \mathbf{w} and Φ , respectively. Due to the discrete nature of variable \mathbf{m} , the problem (6) is hard to solve. Following (Zhou et al., 2021b), by reparameterizing m_i to be an independent Bernoulli random variables with s_i to be 1 and $1 - s_i$ to be 0, problem (6) can be relaxed into:

$$\min_{\mathbf{w}, \mathbf{s}} \mathbb{E}_{p(\mathbf{m}|\mathbf{s})} \mathcal{L}(\{\mathbf{v}, \mathbf{m} \circ \Phi\}) \quad (7)$$

$$s.t. \mathbf{w} \in \mathbb{R}^{d_w}, \mathbf{s} \in \mathcal{S} := \{\mathbf{s} \in [0, 1]^{d_\Phi} : \mathbf{1}^\top \mathbf{s} \leq K\}.$$

We adopt the projected SGD with Gumbel-Softmax (Zhou et al., 2021b) to solve this minimization problem. Detailed algorithm description is placed in the appendix due to space limitation.

4.2. Understanding the Benefits of SparseIRM through Theoretical Analysis

In this section, we present our understanding on the working mechanism of SparseIRM in the same linear model in Section 3.2. We first impose a sparsity constraint on the problem:

$$\min_{\mathbf{v}, \Phi} \mathcal{L}(\{\mathbf{v}, \Phi\}) \quad (8)$$

$$s.t. \mathbf{v} \in \mathbb{R}^{d_v}, \Phi \in \{0, 1\}^{d_\Phi}, \|\Phi\|_1 \leq K.$$

In fact, the problem above is exactly a specific instance of problem (6). The detailed reason can be found in the appendix.

For problem (8), we have the following theorem:

Sparse Invariant Risk Minimization

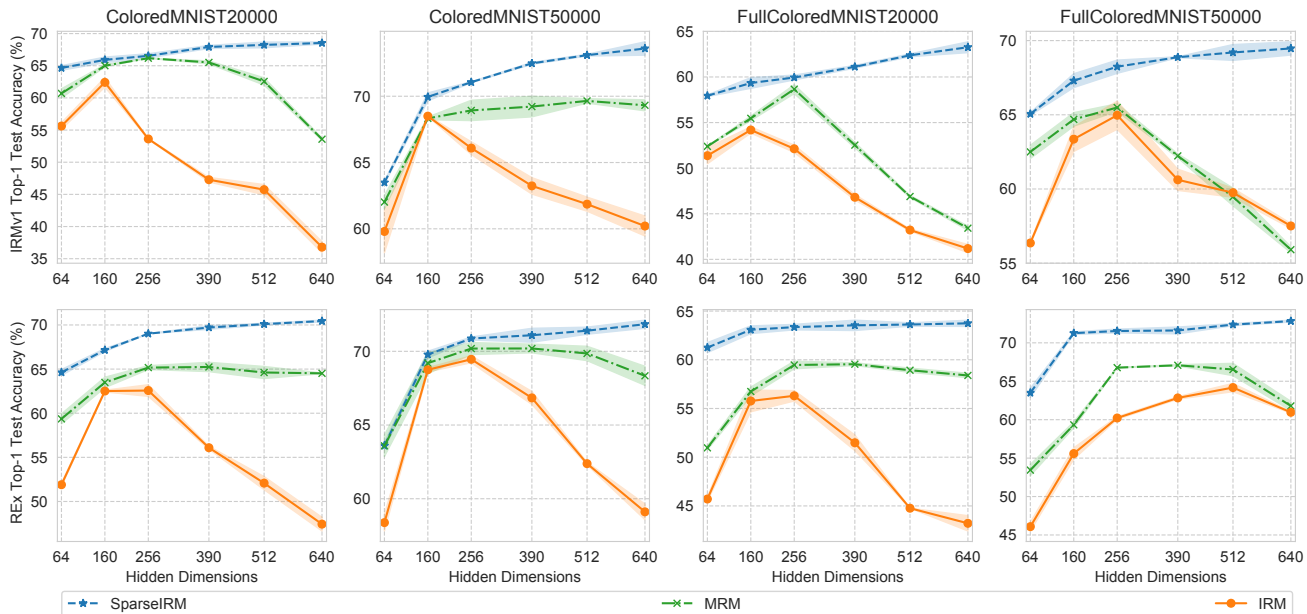


Figure 3. Comparison of MLP on MNIST and MNISTFull with varying hidden dimensions and dataset sizes. The first row indicates comparisons on IRMv1 (Arjovsky et al., 2019) and the second row indicates comparisons on REX (Krueger et al., 2021). SparseIRM successfully deals with the serious overfitting problem in IRM and MRM, leading to a gap of up to 40%.

Theorem 1. Under assumptions specified in Appendix B.6.2, assume $n_{e_1} = n_{e_2} = n$, if $n > Q_1 + Q_2 \ln(d/\delta)$ and choosing $K = d_{inv}$, then with probability at least $1 - \delta$ the following inequality holds:

$$\hat{\mathcal{L}}(\Phi_{inv}) < \hat{\mathcal{L}}(\Phi), \forall \Phi \neq \Phi_{inv} \text{ and } \|\Phi\|_1 \leq K, \quad (9)$$

where Q_1 and Q_2 are constants specified in the appendix.

Theorem 1 indicates that in the linear case, SparseIRM can provably find the invariant features as long as the number of the data samples is larger than a logarithmic term of spurious and random features. The intuition behind Theorem 1 is that, the sparsity constraint limits the number of features to be selected, in which way any combinations of spurious or random features not exceeding the constraint will only lead to a larger loss. In this way, only a feature mask with focus only on invariant features will lead to the minimal loss.

We would like to point out that although in Theorem 1 we choose K as d_{inv} , we find that our algorithm is not sensitive to K in the empirical evaluation.

5. Experiment

In this section, we conduct a series of experiments on benchmarks which are widely-used in latest studies (Arjovsky et al., 2019; Ahmed et al., 2020) to justify the superiority of our SparseIRM. We divide the experiments into three parts. In part one, we conduct detailed experiments on multi-layer-perceptrons (MLP) with varied hidden dimensions on two

datasets ColoredMNIST (Arjovsky et al., 2019) and Full-ColoredMNIST (Ahmed et al., 2020) with varying dataset sizes. In part two, we conduct more overparameterized experiments on large-sized model ResNet-18 (He et al., 2016) on CIFARMNIST and ColoredObject datasets. In part three, we conduct ablation studies to verify the effectiveness of SparseIRM in removing spurious features. Detailed experimental configurations are placed in appendix due to space limitation.

Table 1. Illustration of each dataset. C/FCMNIST stands for ColoredMNIST and FullColoredMNIST. Invariant and Spurious stand for the invariant and spurious features, respectively. The spurious features have strong correlations with the labels, as shown in Training samples. The correlations are reversed in the Testing samples to simulate the distributional shift.

Dataset	Invariant	Spurious	Training	Testing
C/FCMNIST	Digit	Color		
ColoredObject	Object	Background		
CIFARMNIST	CIFAR	MNIST		

The datasets and experimental settings adopted in our experiments align with common practice in previous works on IRM (Arjovsky et al., 2019; Krueger et al., 2021). In all the four datasets, the labels are generated from the invariant features. The spurious features have strong correlations with the labels in the training set but the correlation reverses

Table 2. Comparison of MLP on ColoredMNIST50000 and FullColoredMNIST50000 with varying hidden dimensions.

Dataset	ColoredMNIST50000						FullColoredMNIST50000						
Dim	64	160	256	390	512	640	64	160	256	390	512	640	
Oracle	72.62	72.94	73.00	73.01	74.17	74.52	76.38	76.61	76.12	75.80	75.64	75.72	
ERM	14.87	20.16	21.27	22.44	22.85	22.57	25.25	26.27	27.50	27.67	27.93	28.10	
SparseERM	15.03	20.22	22.51	23.42	22.97	23.51	26.59	28.92	29.15	28.71	28.43	30.11	
IRMV1	IRM	59.80	68.50	66.09	63.24	61.86	60.21	56.36	63.36	64.97	60.62	59.75	57.51
	MRM	62.02	68.33	68.93	69.22	69.65	69.32	62.5	64.69	65.49	62.23	59.47	55.91
	SparseIRM	63.49	69.95	71.06	72.48	73.10	73.61	65.06	67.30	68.24	68.88	69.20	69.47
REx	IRM	58.38	68.75	69.46	68.84	62.38	59.11	46.08	55.59	60.22	62.83	64.18	60.96
	MRM	63.59	69.19	70.19	70.19	69.85	68.34	53.44	59.33	66.77	67.08	66.53	61.79
	SparseIRM	63.60	69.78	70.87	71.09	71.40	71.84	63.50	71.26	71.54	71.61	72.37	72.83

in the testing set. In each dataset there exist two training environments and one testing environment with different correlations. We combine the correlations of two training environments and one testing environment into a tuple. Label noise is added to the datasets to make the task more challenging (Arjovsky et al., 2019; Zhang et al., 2021b).

To demonstrate the superiority of our **SparseIRM** method, we compare with standard empirical risk minimization (**ERM**), sparse empirical risk minimization without IRM loss penalty (**SparseERM**), two classic invariant risk minimization methods **IRMv1** (Arjovsky et al., 2019) and **REx** (Krueger et al., 2021), the sparsify-after-training method **MRM** (Zhang et al., 2021b) and state-of-the-art method **BayesianIRM** (Lin et al., 2022a) which introduces Bayesian Inference into IRM. We include ERM trained on datasets without spurious features to serve as an upper bound (**Oracle**).

5.1. MLP on ColoredMNIST/FullColoredMNIST

The MLP consists of three hidden layers and the details of the structure is given in appendix. The hidden dimensions vary in the range [64, 640] and training dataset sizes vary in {20000, 50000}. We add a number to the end of the dataset name to indicate the training set size. Intuitively, the larger hidden dimensions and smaller training set sizes, the more overparameterized the setting is. Table 2 and Figure 3 presents the Top-1 testing accuracy of Oracle, ERM, SparseERM, IRM, MRM and SparseIRM with varying hidden dimensions and dataset sizes. We achieve the following observations:

1. SparseIRM largely surpasses MRM and IRM with an evident threshold, leading to a gap of up to 40%. The gap becomes larger when the setting is more overparameterized, i.e., when the hidden dimension goes to

640 and dataset number is 20000.

2. IRM suffers from the overfitting problem caused by overparameterization seriously. Take the setting of IRMv1 on ColoredMNIST20000 for example, the performance gap incurred by overparameterization comes up to 27% when comparing the accuracy at 160 and 640 hidden dimensions.
3. MRM preliminarily deals with the overfitting problem caused by overparameterization while still cannot achieve satisfactory performance when the setting becomes more overparameterized. Take the setting of IRMv1 on FullColoredMNIST20000 for example, the performance gap incurred by overparameterization comes up to 15% when comparing the accuracy at 256 and 640 hidden dimensions.
4. SparseIRM solves the overfitting problem caused by overparameterization effectively. SparseIRM is resistant to the changes in hidden dimensions. Specifically, the Top-1 test accuracy even goes up steadily with the increment of hidden dimensions. In the experiment, we find that we effectively solves the the contradiction between model trainability and generalization ability.

5.2. ResNet18 on ColoredObject/CIFARMNIST

In this section, we evaluate the performance of SparseIRM in extremely overparameterized settings. Table 3 reports the detailed Top-1 testing accuracy on ResNet-18 on ColoredObject and CIFARMNIST. We find that MRM collapsed at the ColoredObject dataset and achieves little performance boost in CIFARMNIST. The collapse of MRM can be expected, as from the previous experiments on relatively small-sized MLP, the performance of MRM drops quickly when hidden dimension increases. When the parameter size

Table 3. Comparison of Top-1 Test Accuracy on ResNet-18 on ColoredObject and CIFARMNIST.

Dataset		ColoredObject	CIFARMNIST
Oracle		87.9 ± 0.3	83.7 ± 1.5
ERM		51.6 ± 0.5	39.5 ± 0.4
SparseERM		54.4 ± 0.4	40.1 ± 0.8
BayesianIRM		78.1 ± 0.6	59.3 ± 0.8
IRMv1	IRM	72.5 ± 2.3	51.3 ± 3.0
	MRM	58.4 ± 0.9	56.7 ± 2.3
	SparseIRM	87.4 ± 0.6	63.9 ± 0.4
REx	IRM	73.8 ± 1.3	50.1 ± 2.2
	MRM	55.7 ± 2.9	52.6 ± 1.5
	SparseIRM	80.3 ± 1.1	62.7 ± 0.6

comes up to over ten million parameters in ResNet-18, the setting becomes extremely overparameterized. Therefore MRM selects massive spurious and random features during the first dense training process and then mistakenly discards invariant features misled by the spurious and random features, finally making the model collapsed.

SparseIRM generally beats the baselines by a large margin and achieves striking results approaching the Oracle in IRMv1 in ColoredObject dataset. These results validate the effectiveness of our SparseIRM method in extremely overparameterized ResNet-18 settings.

5.3. Ablation Studies

In this section, we would like to explicitly verify whether the spurious features are removed by our SparseIRM in previous experiments through two experiments. In the first experiment, the idea is to predict the spurious features from the representation. If it cannot predict the spurious features well, it means that the extracted feature representations contain no information about spurious features. Therefore our claim is verified. In the second experiment, we visualize the difference of learned feature representations by merely flipping the value of spurious features, in order to demonstrate the little influence of spurious features to our learned representations. The MLP model is learned in the setting with 640 hidden dimensions and ColoredMNIST50000 and we take the IRMv1 for example.

[Predicting the Spurious Features from the Representations] We generate a ColoredMNIST dataset with 50% correlation across training and testing environments and then obtain the extracted feature representations by feeding images into the learned MLP model. We train a new two layer perceptron network to predict the color based on

the extracted representations. Table 4 presents the training and testing accuracy of the color prediction problem and we find that the IRM and MRM method can still predict the color well on the training set and generalize well on testing set, with accuracy approaching 90% percent. This implies that there still exists information about the color in their learned representations. In strikingly contrast, our SparseIRM achieves nearly optimal performance (both 50% accuracy in training and testing set) in the color prediction task, that is, no meaningful information concerning spurious features is preserved in the learned MLP model. This is consistent with our claim that our SparseIRM can remove the spurious features successfully through the sparsify-during-training process.

Table 4. Training and testing accuracy of the color prediction problem. SparseIRM achieves nearly optimal performance demonstrating that no information about spurious features exists in learned representations.

Dataset	Training	Testing
IRM	89.1 ± 0.3	89.2 ± 1.9
MRM	82.8 ± 1.1	83.4 ± 0.4
SparseIRM	50.3 ± 0.8	50.0 ± 0.7

[Visualize Difference of Feature Representations by Flipping Spurious Features] We randomly sample a image I from the ColoredMNIST testing set. We then flip the spurious feature color of I and denote it as I' . We feed I and I' into the learned MLP model and then plot absolute value of difference of feature representations. From Figure 4, we find that the difference of extracted representations of images with different colors is largely suppressed through our SparseIRM network. In contrast, for the baselines, the difference of extracted feature representations of images with different colors is still very large considering the relatively brighter color of the matrix. This means that the models learned by IRM and MRM can still be affected by spurious features, while our SparseIRM method is invariant to spurious features, verifying its success in removing the spurious features through the sparsify-during-training process.

6. Limitations

In our method, we adopt the weight-level sparsity, which is difficult to be efficiently implemented in the general deep neural network training platforms, such as TensorFlow and PyTorch, to accelerate the training process. In the current stage of IRM study, the training speed is not a main issue as the dataset size is not too large. We will explore efficient training methods (Yuan et al., 2020b; Zhou et al., 2021a; Chen et al., 2021a) in the future .

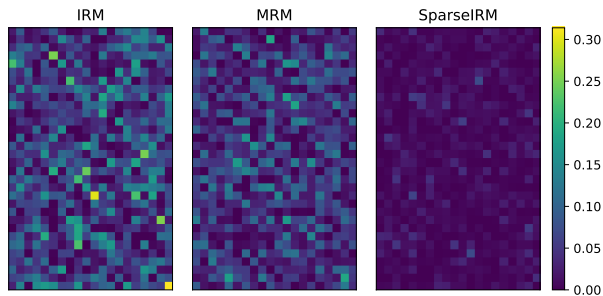


Figure 4. Comparison of absolute value of difference of feature representations by flipping spurious features. The dimension of feature representation 640 and we reshape it into 32×20 matrix for better visualization.

7. Conclusion

In this paper, we propose an effective sparse invariant risk minimization method named SparseIRM to address the overfitting problem in IRM originated from overparameterization. We provide some theoretical results to demonstrate the appealing properties of SparseIRM over the existing methods. Empirically we achieve surprisingly high performance on various datasets and vividly verify the effectiveness of our SparseIRM in removing spurious features in ablation studies.

Acknowledgements

This work is supported by GRF 16201320.

References

- Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.
- Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020a.
- Ahuja, K., Wang, J., Dhurandhar, A., Shanmugam, K., and Varshney, K. R. Empirical or invariant risk minimization? a sample complexity perspective. *arXiv preprint arXiv:2010.16412*, 2020b.
- Ahuja, K., Caballero, E., Zhang, D., Bengio, Y., Mitliagkas, I., and Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. *arXiv preprint arXiv:2106.06607*, 2021.
- Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.
- Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- Bai, H., Sun, R., Hong, L., Zhou, F., Ye, N., Ye, H.-J., Chan, S.-H. G., and Li, Z. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6705–6713, 2021a.
- Bai, H., Zhou, F., Hong, L., Ye, N., Chan, S.-H. G., and Li, Z. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8320–8329, 2021b.
- Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.
- Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. Invariant rationalization. In *International Conference on Machine Learning*, pp. 1448–1458. PMLR, 2020.
- Chen, C., Shen, L., Huang, H., and Liu, W. Quantized adam with error feedback. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–26, 2021a.
- Chen, K., Hong, L., Xu, H., Li, Z., and Yeung, D.-Y. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7546–7554, 2021b.
- Chen, Y., Rosenfeld, E., Sellke, M., Ma, T., and Risteski, A. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *arXiv preprint arXiv:2106.09913*, 2021c.
- Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning. In *International Conference on Machine Learning*, pp. 2189–2200. PMLR, 2021.
- DeGrave, A. J., Janizek, J. D., and Lee, S.-I. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, pp. 1–10, 2021.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

- Diao, S., Bai, J., Song, Y., Zhang, T., and Wang, Y. Zen: Pre-training chinese text encoder enhanced by n-gram representations. *arXiv preprint arXiv:1911.00720*, 2019.
- Diao, S., Xu, R., Su, H., Jiang, Y., Song, Y., and Zhang, T. Taming pre-trained language models with n-gram representations for low-resource domain adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3336–3349, 2021.
- Gao, J., Zhou, Y., Yu, P. L., Joty, S., and Gu, J. Unison: Unpaired cross-lingual image captioning. 2022.
- Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.
- Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., and others. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv:2006.07733*, 2020.
- Gu, J., Cai, J., Joty, S. R., Niu, L., and Wang, G. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7181–7189, 2018.
- Gu, Y., Zhang, W., Fang, C., Lee, J. D., and Zhang, T. How to characterize the landscape of overparameterized convolutional neural networks. *Advances in Neural Information Processing Systems*, 2020.
- Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.
- Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations*, 2016.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.
- Hsu, D., Kakade, S. M., and Zhang, T. Random design analysis of ridge regression. In *Conference on learning theory*, pp. 9–1. JMLR Workshop and Conference Proceedings, 2012.
- Huang, M., Huang, Z., Li, C., Chen, X., Xu, H., Li, Z., and Liang, X. Arch-graph: Acyclic architecture relation predictor for task-transferable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11881–11891, 2022.
- Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.
- Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: convergence and generalization in neural networks. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 6–6, 2021.
- Jin, W., Barzilay, R., and Jaakkola, T. Domain extrapolation via regret minimization. *arXiv preprint arXiv:2006.03908*, 2020.
- Kamath, P., Tangella, A., Sutherland, D., and Srebro, N. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pp. 4069–4077. PMLR, 2021.
- Kawaguchi, K., Huang, J., and Kaelbling, L. P. Effect of depth and width on local minima in deep learning. *Neural computation*, 31(7):1462–1498, 2019.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.
- Kusupati, A., Ramanujan, V., Somani, R., Wortsman, M., Jain, P., Kakade, S., and Farhadi, A. Soft threshold weight reparameterization for learnable sparsity. In *Proceedings of the International Conference on Machine Learning*, July 2020.
- Lin, Y., Lian, Q., and Zhang, T. An empirical study of invariant risk minimization on deep models. *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*, 2021.
- Lin, Y., Dong, H., Wang, H., and Zhang, T. Bayesian invariant risk minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16021–16030, 2022a.
- Lin, Y., Zhu, S., and Cui, P. Zin: When and how to learn invariance by environment inference? *arXiv preprint arXiv:2203.05818*, 2022b.

- Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. Heterogeneous risk minimization. *arXiv preprint arXiv:2105.03818*, 2021a.
- Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. Kernelized heterogeneous risk minimization. *arXiv preprint arXiv:2110.12425*, 2021b.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- Liu, Z., Han, J., Chen, K., Hong, L., Xu, H., Xu, C., and Li, Z. Task-customized self-supervised pre-training with scalable dynamic routing. In *AAAI*, 2022.
- Luo, P., Wang, X., Shao, W., and Peng, Z. Towards understanding regularization in batch normalization. *arXiv preprint arXiv:1809.00846*, 2018.
- Luong, T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.
- Lym, S., Choukse, E., Zangeneh, S., Wen, W., Sanghavi, S., and Erez, M. Prunetrain: fast neural network training by dynamic sparse model reconfiguration. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–13, 2019.
- Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.
- Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Rosenfeld, E., Ravikumar, P., and Risteski, A. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.
- Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.
- Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.
- Shao, W., Meng, T., Li, J., Zhang, R., Li, Y., Wang, X., and Luo, P. Ssn: Learning sparse switchable normalization via sparsestmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.
- Sun, Y., Wang, X., and Tang, X. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1891–1898, 2014.
- Wang, H., Wu, Z., and He, J. Training fair deep neural networks by balancing influence. *arXiv preprint arXiv:2201.05759*, 2022.
- Xie, C., Chen, F., Liu, Y., and Li, Z. Risk variance penalization: From distributional robustness to causality. *arXiv e-prints*, pp. arXiv–2006, 2020.
- Xu, R., Cui, P., Kuang, K., Li, B., Zhou, L., Shen, Z., and Cui, W. Algorithmic decision making with conditional fairness. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2125–2135, 2020.
- Xu, R., Cui, P., Shen, Z., Zhang, X., and Zhang, T. Why stable learning works? a theory of covariate shift generalization. *arXiv preprint arXiv:2111.02355*, 2021.
- Xu, R., Zhang, X., Cui, P., Li, B., Shen, Z., and Xu, J. Regulatory instruments for fair personalized pricing. In *Proceedings of the ACM Web Conference 2022*, pp. 4–15, 2022.
- Xu, Y. and Jaakkola, T. Learning representations that support robust transfer of predictors. *arXiv preprint arXiv:2110.09940*, 2021.
- Ye, N., Li, K., Bai, H., Yu, R., Hong, L., Zhou, F., Li, Z., and Zhu, J. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7947–7958, 2022.
- Yuan, G., Shen, L., and Zheng, W.-S. A block decomposition algorithm for sparse optimization. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 275–285, 2020a.

- Yuan, X., Savarese, P. H. P., and Maire, M. Growing efficient deep networks by structured continuous sparsification. In *International Conference on Learning Representations*, 2020b.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021a.
- Zhang, D., Ahuja, K., Xu, Y., Wang, Y., and Courville, A. Can subnetwork structure be the key to out-of-distribution generalization? *arXiv preprint arXiv:2106.02890*, 2021b.
- Zhang, G., Zhao, H., Yu, Y., and Poupart, P. Quantifying and improving transferability in domain generalization. *arXiv preprint arXiv:2106.03632*, 2021c.
- Zhang, X., Xu, Z., Xu, R., Liu, J., Cui, P., Wan, W., Sun, C., and Li, C. Towards domain generalization in object detection. *arXiv preprint arXiv:2203.14387*, 2022a.
- Zhang, X., Zhou, L., Xu, R., Cui, P., Shen, Z., and Liu, H. Nico++: Towards better benchmarking for domain generalization. *arXiv preprint arXiv:2204.08040*, 2022b.
- Zhou, W., Zeng, Y., Diao, S., and Zhang, X. Vlue: A multi-task benchmark for evaluating vision-language models, 2022a. URL <https://arxiv.org/abs/2205.15237>.
- Zhou, X., Zhang, W., Chen, Z., Diao, S., and Zhang, T. Efficient neural network training via forward and backward propagation sparsification. *Advances in Neural Information Processing Systems*, 2021a.
- Zhou, X., Zhang, W., Xu, H., and Zhang, T. Effective sparsification of neural networks with global sparsity constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3599–3608, 2021b.
- Zhou, X., Lin, Y., Pi, R., Zhang, W., Xu, R., Peng, C., and Zhang, T. Model agnostic sample reweighting for out-of-distribution learning. In *International Conference on Machine Learning*. PMLR, 2022b.
- Zou, F., Shen, L., Jie, Z., Zhang, W., and Liu, W. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11127–11135, 2019.

Supplemental Material: Sparse Invariant Risk Minimization

This appendix can be divided into 6 parts. To be precise,

- In Section A, we give detailed descriptions of four datasets.
- In Section B.1, we give more discussion about our theoretical analysis in this work.
- In Section B.2, we verify our claim in Line 512 that Problem (8) is a specific instance of Problem (6).
- In Sections B.3 to B.6, we present the proofs of the three theorems in the main text.
- In section C, we present our detailed algorithm for solving SparseIRM.
- In Section D, we give the experimental configurations.
- In Section E, we provide some additional experimental results .
- In Section F, we present discussions on future works.

A. Dataset Details



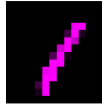






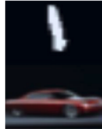
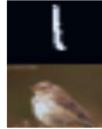

ColoredMNIST (Arjovsky et al., 2019). It contains images from MNIST and the images are labeled as 0 or 1. Each image is attached with a color as the spurious feature. Correlation tuple is (0.9, 0.8, 0.1). Noise ratio is 25%.

FullColoredMNIST (Ahmed et al., 2020). It extends ColoredMNIST to 10 classes. Correlation tuple is (0.999, 0.7, 0.1). Noise ratio is 20%.

ColoredObject (Ahmed et al., 2020; Zhang et al., 2021b). It is constructed by extracting 8 classes of object from MSCOCO and put them onto colored backgrounds. Correlation tuple is (0.999, 0.7, 0.1). Noise ratio is 5%.

CIFARMNIST (Shah et al., 2020; Lin et al., 2021). It is constructed by concatenating images of CIFAR10 with MNIST. The CIFAR images are the invariant features and the MNIST images are the spurious features. Correlation tuple is (0.999, 0.7, 0.1). Noise ratio is 10%.

Table 5. Illustration of each dataset. C/FCMNIST stands for ColoredMNIST and FullColoredMNIST. Invariant and Spurious stand for the invariant and spurious features, respectively. The spurious features have strong correlations with the labels, as shown in Training samples. The correlations are reversed in the Testing samples to simulate the distributional shift.

Dataset	Invariant	Spurious	Training		Testing	
C/FCMNIST	Digit	Color				
ColoredObject	Object	Background				
CIFARMNIST	CIFAR	MNIST				

B. Proofs

B.1. More Discussions on Our Theoretical Analysis

In this paper, we analyze the theoretical properties of our method through a linear model for the following considerations:

- Our main motivation is to show the superiority of our method over the baseline MRM (Zhang et al., 2021b). We show that, even for the simple linear case, MRM can be demolished by overfitting caused by overparameterization. Thus, the superiority of our method over MRM in addressing overfitting is verified.
- We would like to show some intuitive insights of our method instead of presenting thorough rigorous analysis of SparseIRM.
- Although applying IRM to deep neural networks is possible, it is hard to provide theoretical guarantees in this scenario (Rosenfeld et al., 2020). The reason is that we need to find and analyze the global optimum in the analysis of IRM, while theory on the global optimum of deep neural networks is still quite limited, making it not applicable to analyze IRM directly in the complicated setting.

B.2. The Relationship between Problem (6) and Problem (8)

Below, we will show that Problem (8) is a specific instance of Problem (6) as we claimed in Line 512.

Firstly, recall that Problem (6) takes the form of

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{m}} \mathcal{L}(\{\mathbf{v}, \mathbf{m} \circ \Phi\}) \\ \text{s.t. } \mathbf{w} \in \mathbb{R}^{d_w}, \mathbf{m} \in \{0, 1\}^{d_\Phi}, \|\mathbf{m}\|_1 \leq K, \end{aligned} \quad (10)$$

By freezing $\Phi = \mathbf{1}^{d_\Phi}$, it becomes

$$\begin{aligned} \min_{\mathbf{v}, \mathbf{m}} \mathcal{L}(\{\mathbf{v}, \mathbf{m}\}) \\ \text{s.t. } \mathbf{v} \in \mathbb{R}^{d_v}, \mathbf{m} \in \{0, 1\}^{d_\Phi}, \|\mathbf{m}\|_1 \leq K, \end{aligned} \quad (11)$$

Then, we change the notation of \mathbf{m} to be Φ , the problem becomes

$$\begin{aligned} \min_{\mathbf{v}, \Phi} \mathcal{L}(\{\mathbf{v}, \Phi\}) \\ \text{s.t. } \mathbf{v} \in \mathbb{R}^{d_v}, \Phi \in \{0, 1\}^{d_\Phi}, \|\Phi\|_1 \leq K, \end{aligned} \quad (12)$$

which is actually Problem (8). Thus our claim is verified.

B.3. Basics

In real applications, we always have massive spurious and random features while the invariant features are usually low dimensional (Sagawa et al., 2020). So it is reasonable to let $d_s + d_r \gg d_{inv}$. We also assume the sample size in each environment is the same, i.e., $n^e = n, \forall e \in \mathcal{E}$. For any vector v and linear operator M , let the vector norm be $\|v\|_M := \sqrt{v^\top M v}$. When M is omitted, it is assumed to be the identity I , so $\|v\| = \sqrt{v^\top v}$. For a linear operator, let $\|M\|$ be the spectral(operator) norm, i.e., $\|M\| := \sup_v \|Mv\|/\|v\|$. Let $\lambda_{max}[M]$ and $\lambda_{min}[M]$ be the largest and smallest eigenvalue of M , respectively. We use \mathbb{E}^e and $\hat{\mathbb{E}}^e$ to denote the expectation and empirical mean in environment e , e.g.,

$$\mathbb{E}^e[y] = \int y^e d\mathbb{P}(y^e), \quad \hat{\mathbb{E}}^e[y] = \frac{1}{n} \sum_{i=1}^n y_i^e.$$

Further, we use \mathbb{E} and $\hat{\mathbb{E}}$ to denote expectation and empirical mean from *all* environments, e.g.,

$$\mathbb{E}[y] = \frac{1}{|\mathcal{E}|} \sum_{e=1}^{|\mathcal{E}|} \mathbb{E}^e[y], \quad \hat{\mathbb{E}}[y] = \frac{1}{|\mathcal{E}|} \sum_{e=1}^{|\mathcal{E}|} \hat{\mathbb{E}}^e[y].$$

Given the feature mask Φ , let Σ_Φ denote the design matrix for the environment mixture, i.e.,

$$\Sigma_\Phi = \mathbb{E}[\Phi(\mathbf{x})\Phi(\mathbf{x})^\top]$$

We further let Σ_{Φ}^e be the design matrix in environment e and $\hat{\Sigma}_{\Phi}$ be the empirical design matrix. When it is clear from the context, we drop the subscript Φ for simplicity. Let β achieve the minimum mean square error over all linear function, i.e.,

$$\beta = \arg \min_{\mathbf{v}} \mathbb{E}[(y - \mathbf{v}^{\top} \Phi(\mathbf{x}))^2] \quad (13)$$

We also use β^e and $\hat{\beta}$ to denote the counterparts of β similarly as defined above.

Before providing the proofs, we first introduce the extension of IRMv1 to the minimax formulation:

$$\min_{\mathbf{v}, \Phi} \max_{\mathbf{v}^e} \sum_e \mathcal{R}^e(\mathbf{v}, \Phi) + \lambda[\mathcal{R}^e(\mathbf{v}, \Phi) - \mathcal{R}^e(\mathbf{v}^e, \Phi)] \quad (14)$$

Then the loss for a feature representation Φ is

$$\mathcal{L}(\Phi) = \sum_e \mathcal{R}^e(\beta, \Phi) + \lambda(\mathcal{R}^e(\beta, \Phi) - \mathcal{R}^e(\beta^e, \Phi)), \quad (15)$$

where β and β^e are defined in Eqn. (13). Further, we have:

$$\nabla_{\mathbf{v}} \mathcal{R}^e(\mathbf{v}, \Phi) \Big|_{\mathbf{v}=\beta^e} = \mathbb{E}^e[\Phi(\mathbf{x})(y - \Phi(\mathbf{x}))] = \mathbf{0}.$$

We then have into the following:

$$\begin{aligned} & \mathcal{R}^e(\mathbf{v}, \Phi) \\ &= \mathbb{E}^e (y - \Phi(x))^{\top} (\mathbf{v} - \beta^e + \beta^e)^2 \\ &= \mathbb{E}^e (\Phi(x)^{\top} (\mathbf{v} - \beta^e))^2 + \mathbb{E}^e ((\mathbf{v} - \beta^e)^{\top} \Phi(x)) (y - \Phi(x)^{\top} \beta^e) + \mathbb{E}^e (y - \Phi(x)^{\top} \beta^e)^2 \\ &= \mathbb{E}^e (\Phi(x)^{\top} (\mathbf{v} - \beta^e))^2 + \mathcal{R}^e(\beta^e, \Phi) \end{aligned}$$

So the penalty term in Eq. (14) translates into the following:

$$\mathcal{R}^e(\mathbf{v}, \Phi) - \mathcal{R}^e(\beta^e, \Phi) = \mathbb{E}^e (\Phi(x)^{\top} (\mathbf{v} - \beta^e))^2 \quad (16)$$

Then IRM loss in Eq. (14) is

$$\begin{aligned} & \sum_e \mathcal{R}^e(\beta, \Phi) + \lambda(\mathcal{R}^e(\beta, \Phi) - \mathcal{R}^e(\beta^e, \Phi)) \\ &= \sum_e \mathbb{E}^e (y - \Phi(\mathbf{x})^{\top} \beta)^2 + \lambda(\Phi(\mathbf{x})^{\top} (\beta - \beta^e))^2. \end{aligned} \quad (17)$$

Comparing IRMv1 defined in Eq. (J-IRMv1) and Eq. (14), we can see that Eq. (J-IRMv1) is the first order approximation of Eq. (14) by initializing \mathbf{v}^e with β and then taking a gradient step on $\mathcal{R}^e(\mathbf{v}, \Phi)$ by $\mathbf{v}^e \leftarrow \beta - \eta \nabla_{\mathbf{v}^e} \mathcal{R}^e(\mathbf{v}^e, \Phi) \Big|_{\mathbf{v}^e=\mathbf{v}}$. Then

$$\mathcal{R}^e(\mathbf{v}, \Phi) - \mathcal{R}^e(\mathbf{v} - \eta \nabla_{\mathbf{v}^e} \mathcal{R}^e(\mathbf{v}, \Phi), \Phi) = \eta \|\nabla_{\mathbf{v}^e} \mathcal{R}^e(\mathbf{v}, \Phi)\|^2 + O(\eta^2)$$

In the later part, we provide the proof of our results based on the formulation in Eq. (14).

B.4. Proof for Proposition 1

Proof. Because of the condition $d_{inv} + d_s + d_r > n_{e_1} + n_{e_2}$, we have

$$\hat{\mathcal{R}}^e(\hat{\beta}, \Phi_{all}) = 0, \hat{\mathcal{R}}^e(\hat{\beta}^e, \Phi_{all}) = 0, \forall e \in \{e_1, e_2\}.$$

Putting these into Eq. (15), we have

$$\hat{\mathcal{L}}(\Phi_{all}) = 0. \quad (18)$$

On the other hand, we have the following,

$$\begin{aligned}
 \hat{\mathcal{L}}(\Phi_{inv}) &= \sum_e \hat{\mathcal{R}}^e(\Phi_{inv}) + \frac{1}{n_e} \sum_{i=1}^{n_e} (\Phi(\mathbf{x})^\top (\hat{\beta} - \hat{\beta}^e))^2 \\
 &\geq \sum_e \hat{\mathcal{R}}^e(\Phi_{inv}) \\
 &\geq 0
 \end{aligned} \tag{19}$$

The first and second inequality are due to the non-negativity of the penalty and risk. Putting Eq. (18) and Eq. (19) together, we finish the proof. \square

B.5. Proof of Corollary 1

Proof. The proof follows directly from that of Theorem 1 but replacing Φ_{all} with Φ_{sr} . When spurious and random feature has large dimension, $d_s + d_r > n_{e_1} + n_{e_2}$, $\mathcal{L}(\Phi_{sr})$ will be already 0. \square

B.6. Proof of Theorem 1

B.6.1. PRELIMINARIES

Recall the data generation process in Section 3.2, by simple algebra, we know that $\mathbb{E}^e[\Phi(\mathbf{x})y] = \mathbb{E}[\Phi(\mathbf{x})y]$, $\mathbb{E}^e[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_{inv,r}] = \mathbb{E}[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_{inv,r}]$, $\mathbb{E}^e[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_s] = \mathbb{E}[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_s]$ and $\mathbb{E}^e[\Phi(\mathbf{x})_s^\top \Phi(\mathbf{x})_s] = \mathbb{E}[\Phi(\mathbf{x})_s^\top \Phi(\mathbf{x})_s] + \text{Diag}(\dots, (\alpha_i^e)^2 - (\alpha_i)^2, \dots)$, where $\alpha_i = \sqrt{\sum_e (\alpha_i^e)^2}$, $\forall i \in [d_s]$. Specifically, we have

$$\begin{aligned}
 \Sigma_{\Phi}^e &= \begin{pmatrix} \mathbb{E}^e[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_{inv,r}] & \mathbb{E}^e[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_s] \\ \mathbb{E}^e[\Phi(\mathbf{x})_s^\top \Phi(\mathbf{x})_{inv,r}] & \mathbb{E}^e[\Phi(\mathbf{x})_s^\top \Phi(\mathbf{x})_s] \end{pmatrix} \\
 &= \begin{pmatrix} \mathbb{E}[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_{inv,r}] & \mathbb{E}[\Phi(\mathbf{x})_{inv,r}^\top \Phi(\mathbf{x})_s] \\ \mathbb{E}[\Phi(\mathbf{x})_s^\top \Phi(\mathbf{x})_{inv,r}] & \mathbb{E}[\Phi(\mathbf{x})_s^\top \Phi(\mathbf{x})_s] + \begin{pmatrix} (\alpha_1^e)^2 - (\alpha_1)^2 & 0 & \dots & 0 \\ 0 & (\alpha_2^e)^2 - (\alpha_2)^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (\alpha_{d_s(\Phi)}^e)^2 - (\alpha_{d_s(\Phi)})^2 \end{pmatrix} \end{pmatrix} \\
 &= \Sigma_{\Phi} + \text{Diag}[\underbrace{0, \dots, 0}_{d_{inv,s}(\Phi)}, (\alpha_1^e)^2 - (\alpha_1)^2, \dots, (\alpha_{d_s(\Phi)}^e)^2 - (\alpha_{d_s(\Phi)})^2]
 \end{aligned}$$

So Σ^e and Σ only differ by some diagonal elements corresponding to the spurious feature. Let $\{\lambda_i\}_{i=1}^d$ and $\{\lambda_i^e\}_{i=1}^d$ be the eigenvalues of Σ and Σ^e , respectively. Their corresponding eigenvectors are $\{\mathbf{v}_i\}_{i=1}^d$ and $\{\mathbf{v}_i^e\}_{i=1}^d$. It is easy to see that Σ and Σ^e share the same orthonormal basis Π , indicating $\mathbf{v}_i = \mathbf{v}_i^e, \forall i \in [K]$.

$$\lambda_i^e = \begin{cases} \lambda_i, & \text{if } i \in [d_{inv}(\Phi)] \cup [d_r(\Phi)], \\ \lambda_i + (\alpha_i^e)^2 - \alpha_i^2, & \text{if } i \in [d_s(\Phi)]. \end{cases} \tag{20}$$

Assume the maximum and minimum possible eigenvalue to be λ_{max} and $\lambda_{min} > 0$, i.e.,

$$\begin{aligned}
 \forall e \in \mathcal{E}, \quad \lambda_{min} \leq \lambda_i^e \leq \lambda_{max}, \\
 \text{and} \quad \lambda_{min} \leq \lambda_i \leq \lambda_{max}.
 \end{aligned}$$

Let Φ_{inv} denote the feature mask that merely selects K invariant features, $\Phi_{inv,r}$ be the feature mask that selects random features but no spurious features (the number of spurious features can be arbitrary), $\Phi_{inv,r,s}$ be the feature mask that selects spurious features (the number of random and invariant features can be arbitrary). Our goal is to show $\hat{\mathcal{L}}(\Phi_{inv}) < \hat{\mathcal{L}}(\Phi_{inv,r,s})$ and $\hat{\mathcal{L}}(\Phi_{inv}) < \hat{\mathcal{L}}(\Phi_{inv,r})$ holds with high probability.

B.6.2. ASSUMPTIONS AND TECHNICAL LEMMAS

In this paper, we adopt the following standard assumptions from the existing methods (Arjovsky et al., 2019; Rosenfeld et al., 2020):

Assumption 1. For each feature mask Φ , there exists ρ such that, almost surely,

$$\frac{\|(\Sigma_{\Phi}^e)^{-1/2}\Phi(\mathbf{x})\|}{\sqrt{\mathbb{E}^e[\|(\Sigma_{\Phi}^e)^{-1/2}\Phi(\mathbf{x})\|^2]}} \leq \rho, \forall e \in \mathcal{E}, \text{ and } \frac{\|(\Sigma_{\Phi})^{-1/2}\Phi(\mathbf{x})\|}{\sqrt{\mathbb{E}[\|(\Sigma_{\Phi})^{-1/2}\Phi(\mathbf{x})\|^2]}} \leq \rho.$$

Assumption 2. There exists finite $\sigma > 0$ such that, almost surely,

$$\mathbb{E}[\exp(\eta\epsilon_{inv})] \leq \exp(\eta^2\sigma^2), \forall \eta \in \mathbb{R}.$$

Assumption 3. There exists s and $\bar{\gamma}$, such that:

$$\forall i \in [d_{inv}], |\gamma_i| \geq \bar{\gamma}, \text{ and } \mathbb{V}[x_{inv,i}] \geq s.$$

Assumption 1 and 2 are a standard statistical assumption which can also be found in (Hsu et al., 2012). They can be satisfied when each $x_{inv,i}$ and ϵ_{inv} are bounded sub-Gaussian variables. Assumption 3 requires each x_{inv} should explain sufficient amount of variance of y

Assumption 4. For i th spurious feature, let $\alpha_i = \sqrt{\sum_e (\alpha_i^e)^2}, \forall i \in [d_s]$. There exists a constant $\Delta > 0$, the following inequality holds for each spurious feature,

$$\forall i \in [d_s], \exists e \in \mathcal{E}, |\alpha_i^2 - (\alpha_i^e)^2| \geq \Delta.$$

Assumption 5. For each K subset of feature \mathbf{x} that is selected by Φ , the projection of $\mathbb{E}[\Phi(\mathbf{x})^\top y]$ on each basis corresponding to spurious feature is non zero, i.e., there exists a constant C such that, if $\Phi(\mathbf{x})_i$ is a spurious feature, then

$$\exists e \in \mathcal{E}, |\mathbb{E}^e[\Phi(\mathbf{x})y]^\top \mathbf{v}_i| \geq C > 0.$$

Assumption 4 requires that the coefficient of each spurious feature exhibits a certain level of variation among the two environments. This is reasonable according to the definition of spurious feature that the conditional of target x_s on y varies in different environments, otherwise there is no way to differentiate a spurious feature from the invariant feature (Arjovsky et al., 2019; Rosenfeld et al., 2020). Assumption 5 ensures that the coefficients of a spurious feature can not be always 0, otherwise IRM can not differentiate between the spurious and invariant features, neither.

We first present some useful lemmas from (Hsu et al., 2012).

Lemma 1 (Excess mean squared error, Proposition 5 of (Hsu et al., 2012)). For any v ,

$$\begin{aligned} \mathbb{E}[(y - \mathbf{v}^\top \Phi(x))^2] - \mathbb{E}[(y - \beta^\top \Phi(x))^2] &= \mathbb{E}[(\mathbf{v} - \beta)^\top \Phi(x)]^2 = \|\mathbf{v} - \beta\|_{\Sigma}^2 \\ \hat{\mathbb{E}}[(y - \mathbf{v}^\top \Phi(x))^2] - \hat{\mathbb{E}}[(y - \hat{\beta}^\top \Phi(x))^2] &= \|\mathbf{v} - \hat{\beta}\|_{\hat{\Sigma}}^2. \end{aligned}$$

The same arguments also hold for environment e .

Lemma 2 (Effect of errors in $\hat{\Sigma}$, Lemma 2 and 3 of (Hsu et al., 2012)). With Assumption 1, for any $\delta \leq \min\{1, de^{-2.6}\}$, with probability at least $1 - \delta$,

$$\|\Sigma^{-1/2}(\hat{\Sigma} - \Sigma)\Sigma^{-1/2}\| \leq \sqrt{\frac{4\rho^2 d(\ln d + \ln(1/\delta))}{n}} + \frac{2\rho^2 d(\ln d + \ln(1/\delta))}{3n}.$$

Further, if $\|\Sigma^{-1/2}(\hat{\Sigma} - \Sigma)\Sigma^{-1/2}\| < 1$, then

$$\|\Sigma^{1/2}\hat{\Sigma}^{-1}\Sigma^{1/2}\| \leq \frac{1}{1 - \|\Sigma^{-1/2}(\hat{\Sigma} - \Sigma)\Sigma^{-1/2}\|}.$$

Lemma 3 (Regret Error of Empirical OLS Solution, Theorem 1 and Remark 9 of (Hsu et al., 2012)). Pick any $\delta \leq \min\{1, de^{-2.6}\}$, by Assumption 1 and 2, with probability at least $1 - \delta$, the following holds:

$$\|\hat{\beta} - \beta\|_{\Sigma}^2 \leq \frac{\sigma^2(d + 2\sqrt{d \ln(3/\delta)} + 2 \ln(3/\delta))}{n} + o(1/n),$$

where $o(1/n)$ means higher order term.

B.6.3. PROOF OF THE MAIN THEOREM

Proof. The Eq. (14) can be translated into

$$\begin{aligned}\hat{\mathcal{L}}(\Phi) &= \min_{\mathbf{v}} \max_{\mathbf{v}^e} \sum_e \mathcal{R}^e(\mathbf{v}, \Phi) + \lambda [\mathcal{R}^e(\mathbf{v}, \Phi) - \mathcal{R}^e(\mathbf{v}^e, \Phi)] \\ &= \min_{\mathbf{v}} \max_{\mathbf{v}^e} \sum_e \hat{\mathbb{E}}^e(y - \langle \mathbf{v}, \Phi(\mathbf{x}) \rangle)^2 + \lambda \left[\hat{\mathbb{E}}^e(y - \langle \mathbf{v}, \Phi(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \mathbf{v}^e, \Phi(\mathbf{x}) \rangle)^2 \right] \\ &= \sum_e \hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi(\mathbf{x}) \rangle)^2 + \lambda \left[\hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi(\mathbf{x}) \rangle)^2 \right],\end{aligned}$$

where $\hat{\beta}$ and $\hat{\beta}^e$ is defined as Eq. (13). The penalty term above is

$$\begin{aligned}& \sum_e \hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi(\mathbf{x}) \rangle)^2 \\ &= \underbrace{\sum_e \hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi(\mathbf{x}) \rangle)^2 - \mathbb{E}^e(y - \langle \beta, \Phi(\mathbf{x}) \rangle)^2}_{\xi_a(\Phi)} - \underbrace{\sum_e \left(\hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi(\mathbf{x}) \rangle)^2 - \mathbb{E}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2 \right)}_{\xi_b(\Phi)} \\ & \quad + \underbrace{\sum_e \mathbb{E}^e(y - \langle \beta, \Phi(\mathbf{x}) \rangle)^2 - \mathbb{E}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2}_{\xi_c(\Phi)}\end{aligned}$$

We will tackle (a) - (c) one by one. First,

$$\begin{aligned}|\xi_b(\Phi)| &= \sum_e |\hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2 + \hat{\mathbb{E}}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2 - \mathbb{E}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2| \\ &\leq \sum_e |\hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2| + |\hat{\mathbb{E}}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2 - \mathbb{E}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2|\end{aligned}$$

One on hand, by Hoeffding's Inequality, we have with probability at least $1 - \delta/3$,

$$|\hat{\mathbb{E}}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2 - \mathbb{E}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2| \leq \sqrt{\ln(3/\delta)/n}. \quad (21)$$

On the other hand, we have

$$\begin{aligned}& |\hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \beta^e, \Phi(\mathbf{x}) \rangle)^2| \\ &= \|\beta^e - \hat{\beta}^e\|_{\hat{\Sigma}^e}^2 \\ &= |(\beta^e - \hat{\beta}^e)^\top \Sigma^{e,-1/2} \hat{\Sigma}^{e,1/2} \hat{\Sigma}^{e,-1} \Sigma^{e,1/2} \Sigma^{e,-1/2} (\beta^e - \hat{\beta}^e)| \\ &\leq \|(\beta^e - \hat{\beta}^e)^\top \Sigma^{e,-1/2}\|_2^2 \|\Sigma^{e,1/2} \hat{\Sigma}^{e,-1} \Sigma^{e,1/2}\|_2\end{aligned}$$

The first equality is due to Lemma 1. Further, by Lemma 2 with probability at least $1 - \delta/3$,

$$\|\Sigma^{e,1/2} \hat{\Sigma}^{e,-1} \Sigma^{e,1/2}\|_2 \leq \frac{1}{1 - \kappa}, \quad (22)$$

where $\kappa = \sqrt{\frac{4\rho^2 K(\ln K + \ln(3/\delta))}{n}} + \frac{2\rho^2 K(\ln K + \ln(3/\delta))}{3n}$. Further, if we have

$$n > 8/3\rho^2 K(\ln K + \ln(3/\delta)), \quad (23)$$

then

$$\kappa = \sqrt{\frac{2\rho^2 K(\ln K + \ln(3/\delta))}{3n}} + \frac{2\rho^2 K(\ln K + \ln(3/\delta))}{3n} \leq 1/2 + 1/2 * 1/6 < 3/4$$

By Lemma 1, we have with probability at least $1 - \delta/3$,

$$\|(\beta^e - \hat{\beta}^e)^\top \Sigma^{e,-1/2}\|_2^2 \quad (24)$$

$$= \|\beta^e - \hat{\beta}^e\|_{\Sigma^e} \quad (25)$$

$$\begin{aligned} &\leq \frac{\sigma^2(K + 2\sqrt{K \ln(9/\delta)} + 2 \ln(9/\delta))}{n^e} + o(1/n) \\ &\leq \frac{\sigma^2(2K + 3 \ln(9/\delta))}{n} + o(1/n). \end{aligned} \quad (26)$$

Putting Eq. (21), (22) and (24) together, we have with probability at least $1 - \delta$, for all $e \in \mathcal{E}$,

$$|\xi_b(\Phi)| \leq \underbrace{\frac{1}{1-\kappa} \frac{\sigma^2 |\mathcal{E}| (2K + 3 \ln(9|\mathcal{E}|/\delta))}{n}}_{\zeta} + |\mathcal{E}| \sqrt{\ln(3|\mathcal{E}|/\delta)/n}. \quad (27)$$

Similarly, we have with probability at least $1 - \delta$,

$$|\xi_a(\Phi)| \leq \frac{1}{1-\kappa} \frac{\sigma^2(2K + 3 \ln(9/\delta))}{n|\mathcal{E}|} + \sqrt{\ln(3/\delta)/n|\mathcal{E}|} < \zeta. \quad (28)$$

by noticing that the mixture of environments can be regarded as one environment with $n|\mathcal{E}|$ samples.

To bound $\xi_c(\Phi)$, we have

$$\begin{aligned} \xi_c(\Phi) &= \sum_e (\beta - \beta^e)^\top \Sigma^e (\beta - \beta^e) \\ &= \sum_e (\Sigma^{-1} \mathbb{E}[\Phi(\mathbf{x})y] - \Sigma^{e,-1} \mathbb{E}^e[\Phi(\mathbf{x})y])^\top \Sigma^e (\Sigma^{-1} \mathbb{E}[\Phi(\mathbf{x})y] - \Sigma^{e,-1} \mathbb{E}^e[\Phi(\mathbf{x})y]) \\ &= \sum_e \mathbb{E}^e[\Phi(\mathbf{x})y]^\top (\Sigma^{-1} - \Sigma^{e,-1})^\top \Sigma^e (\Sigma^{-1} - \Sigma^{e,-1}) \mathbb{E}^e[\Phi(\mathbf{x})y] \\ &= \sum_e \mathbb{E}^e[\Phi(\mathbf{x})y]^\top \Pi (\Lambda^{-1} - \Lambda^{e,-1})^\top \Lambda^e (\Lambda^{-1} - \Lambda^{e,-1}) \Pi^\top \mathbb{E}^e[\Phi(\mathbf{x})y] \\ &= \sum_e \sum_i^K (\mathbb{E}^e[\Phi(\mathbf{x})y]^\top \mathbf{v}_i)^2 \lambda_i \left(\frac{1}{\lambda_i^e} - \frac{1}{\lambda_i} \right)^2 \end{aligned}$$

The first equality is due to Lemma 1, the fourth equality is by the fact that $\mathbb{E}^e[\Phi(\mathbf{x})y] = \mathbb{E}[\Phi(\mathbf{x})y]$. So by Assumption 4 and Eq. 20, we have

$$\xi_c(\Phi) \begin{cases} = 0, & \text{if } d_s(\Phi) = 0, \\ \geq \frac{d_s(\Phi) C^2 \lambda_{\min} \Delta^2}{\lambda_{\max}^4} \geq \frac{C^2 \lambda_{\min} \Delta^2}{\lambda_{\max}^4}, & \text{if } d_s(\Phi) \geq 1. \end{cases} \quad (29)$$

Step 1) Comparing $\hat{\mathcal{L}}(\Phi_{inv})$ with $\hat{\mathcal{L}}(\Phi_{inv,r,s})$. We have the following:

$$\begin{aligned} &\hat{\mathcal{L}}(\Phi_{inv,r,s}) - \hat{\mathcal{L}}(\Phi_{inv}) \\ &= \sum_e \hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi_{inv,r,s}(\mathbf{x}) \rangle)^2 + \lambda \left[\hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi_{inv,r,s}(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi_{inv,r,s}(\mathbf{x}) \rangle)^2 \right] \\ &\quad - \left(\sum_e \hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi_{inv}(\mathbf{x}) \rangle)^2 + \lambda \left[\hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi_{inv}(\mathbf{x}) \rangle)^2 - \hat{\mathbb{E}}^e(y - \langle \hat{\beta}^e, \Phi_{inv}(\mathbf{x}) \rangle)^2 \right] \right) \\ &\geq \lambda [\xi_a(\Phi_{inv,r,s}) + \xi_b(\Phi_{inv,r,s}) + \xi_c(\Phi_{inv,r,s})] \\ &\quad - [\xi_a(\Phi_{inv}) + \mathbb{V}(\epsilon)] - \lambda [\xi_a(\Phi_{inv}) - \xi_b(\Phi_{inv}) + \xi_c(\Phi_{inv})] \\ &\geq - (4\lambda + 1)\zeta + \lambda \xi_c(\Phi_{inv,r,s}) - \mathbb{V}(\epsilon) \end{aligned}$$

The first inequality is due to $\hat{\mathbb{E}}^e(y - \langle \hat{\beta}, \Phi_{inv}(\mathbf{x}) \rangle)^2 \geq 0$ and the definition of ξ_a , ξ_b and ξ_c . The second inequality is due to $|\xi_a(\Phi)|, |\xi_b(\Phi)| \leq \zeta$ (see Eq. (28) and (27)) and $\xi_c(\Phi_{inv}) = 0$ (see Eq. (29)).

Then, by taking

$$\lambda = \frac{2\mathbb{V}(\epsilon)}{K_s C^2 \lambda_{min} \Delta^2 / \lambda_{max}^4} \geq \frac{2\mathbb{V}(\epsilon)}{\xi_c(\Phi_{inv,r,s})},$$

we have

$$\hat{\mathcal{L}}(\Phi_{inv,r,s}) - \hat{\mathcal{L}}(\Phi_{inv}) > -(4\lambda + 1) \left(\frac{1}{1 - \kappa} \frac{2\sigma^2(2K + 3 \ln(36/\delta))}{n} + \sqrt{4 \ln(12/\delta)/n} \right) + \mathbb{V}(\epsilon) > 0$$

when the following holds

$$n > \max \left\{ \frac{(32\lambda + 8)\sigma^2(2K + 3 \ln(36/\delta))}{\mathbb{V}(\epsilon)}, \frac{(8\lambda + 2)^2 \ln(36/\delta)}{\mathbb{V}(\epsilon)^2} \right\}. \quad (30)$$

Step 2) Comparing $\hat{\mathcal{L}}(\Phi_{inv})$ with $\hat{\mathcal{L}}(\Phi_{inv,r})$. Assume the absolute value of each element of γ is lower bounded as $|\gamma_i| \geq \bar{\gamma}, \forall i \in [d_r]$. The variance of each invariant feature is lower bounded as $\mathbb{V}(x_{inv,i}) \geq s, \forall i \in [d_r]$.

First, by simple algebra, we have

$$\sum_e \mathbb{E}^e(y - \langle \beta, \Phi_{inv,r}(\mathbf{x}) \rangle)^2 - \mathbb{E}^e(y - \langle \beta, \Phi_{inv}(\mathbf{x}) \rangle)^2 = \sum_{i \in [d_r(\Phi)]} r_i \mathbb{V}(x_{inv,i})$$

Intuitively, to replace a feature $x_{inv,i}$ by a random feature lost a explanation component of y , thus resulting in larger loss (which is equal to the variance of explanation component, $x_{inv,i}$).

So with probability at least $1 - \delta$, we have

$$\begin{aligned} & \hat{\mathcal{L}}(\Phi_{inv,r}) - \hat{\mathcal{L}}(\Phi_{inv}) \\ &= \left(\xi_a(\Phi_{inv,r}) + \mathbb{V}(\epsilon) + \sum_{i \in [d_r(\Phi)]} r_i \mathbb{V}(x_{inv,i}) \right) + \lambda \left(\xi_a(\Phi_{inv,r}) + \xi_b(\Phi_{inv,r}) + \xi_c(\Phi_{inv,r}) \right) \\ & \quad - \left(\xi_a(\Phi_{inv}) + \mathbb{V}(\epsilon) \right) - \lambda \left(\xi_a(\Phi_{inv}) - \xi_b(\Phi_{inv}) + \xi_c(\Phi_{inv}) \right) \\ &= \left((\lambda + 1)\xi_a(\Phi_{inv,r}) + \lambda\xi_b(\Phi_{inv,r}) - (\lambda + 1)\xi_a(\Phi_{inv}) - \lambda\xi_b(\Phi_{inv}) \right) + \sum_{i \in [d_r(\Phi)]} r_i \mathbb{V}(x_i) \\ &\geq - (4\lambda + 1)\zeta + \bar{\gamma}s \end{aligned}$$

The second inequality is because $\xi_c(\Phi_{inv}) = \xi_c(\Phi_{inv,r}) = 0$. The first inequality is due to the definition of $\bar{\gamma}$ and s in Assumption 3. The second inequality is due to $|\xi_a(\Phi)|, |\xi_b(\Phi)| \leq \zeta$ (see Eq. (28) and (27)) and $\xi_c(\Phi_{inv}) = 0$ (see Eq. (29)). So if

$$n > \max \left\{ \frac{(32\lambda + 16)\sigma^2(2K + 3 \ln(36/\delta))}{\bar{\gamma}s}, \frac{(8\lambda + 4)^2 \ln(36/\delta)}{\bar{\gamma}^2 s^2} \right\}, \quad (31)$$

then $\hat{\mathcal{L}}(\Phi_{inv,r}) - \hat{\mathcal{L}}(\Phi_{inv}) > 0$. Putting Eq. (23), (30) and (31) together, we have with probability at least $1 - \delta$ such if the following holds:

$$\begin{aligned} n &> \underbrace{\left(2K\sigma^2(32\lambda + 16) \left(\frac{1}{\bar{\gamma}s} + \frac{1}{\mathbb{V}[\epsilon]} \right) + 8\rho^2 K \ln K \right)}_{Q_1} + \underbrace{\left(3(32\lambda + 16)^2 \left(\frac{1}{\bar{\gamma}s} + \frac{1}{\bar{\gamma}^2 s^2} + \frac{1}{\mathbb{V}[\epsilon]} + \frac{1}{\mathbb{V}[\epsilon]^2} \right) + 8\rho^2 K \right)}_{Q_2} \ln(36/\delta) \\ &= Q_1 + \tilde{Q}_2 \ln(36/\delta) \end{aligned}$$

Finally, noticing that the choices of $\Phi_{inv,r}$ and Φ_{inv} are no more than $2d^K$, we then conclude that have that sparse IRM can uniquely identify Φ_{inv} if

$$n > \tilde{Q}_1 + \tilde{Q}_2 \ln(72d^K/\delta) > Q_1 + Q_2 \ln(d/\delta),$$

where

$$\begin{aligned}\tilde{Q}_1 &= 2K\sigma^2(32\lambda + 16)\left(\frac{1}{\gamma s} + \frac{1}{\sqrt{\mathbb{V}[\epsilon]}}\right) + 8\rho^2 K \ln K, \\ \tilde{Q}_2 &= 3(32\lambda + 16)^2\left(\frac{1}{\gamma s} + \frac{1}{\gamma^2 s^2} + \frac{1}{\sqrt{\mathbb{V}[\epsilon]}} + \frac{1}{\sqrt{\mathbb{V}[\epsilon]^2}}\right)K + 8\rho^2 K^2, \\ Q_1 &= \tilde{Q}_1 + \tilde{Q}_2 \ln(72), \\ Q_2 &= \tilde{Q}_2 K, \\ \lambda &= \frac{2\mathbb{V}(\epsilon)}{K_s C^2 \lambda_{\min} \Delta^2 / \lambda_{\max}^4}.\end{aligned}$$

□

C. Algorithm

We present our training method for solving Problem (7) in Algorithm 1, which is adapted from the algorithm in (Zhou et al., 2021b).

Algorithm 1 Sparse Invariant Risk Minimization (SparseIRM)

Input: target remaining ratio $k_f = 0.5$, a dense network w .

- 1: Initialize w , assign probabilities s to weights w , let $s = \mathbf{1}$, $K = k_f d_w$ and $\tau = 1$.
- 2: **for** training epoch $t = 1, 2 \dots T$ **do**
- 3: **for** each training iteration **do**
- 4: Sample mini batch of data $\mathcal{B} = \cup_{e \in \mathcal{E}_t} \mathcal{B}^e$, with $\mathcal{B}^e = \{(\mathbf{x}_1^e, \mathbf{y}_1^e), \dots, (\mathbf{x}_B^e, \mathbf{y}_B^e)\}$.
- 5: Generate \mathbf{g}_1 and \mathbf{g}_0 with each element sampled from Gumbel(0, 1).
- 6: $\mathbf{s} \leftarrow \text{proj}_C(z)$, with $z = \mathbf{s} - \eta \nabla_s \mathcal{L}_{\mathcal{B}}\left(\mathbf{v}, \sigma\left(\frac{\ln(\frac{\mathbf{s}}{1-\mathbf{s}}) + \mathbf{g}_1 - \mathbf{g}_0}{\tau}\right) \circ \Phi\right)$.
- 7: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_w \mathcal{L}_{\mathcal{B}}\left(\mathbf{v}, \sigma\left(\frac{\ln(\frac{\mathbf{s}}{1-\mathbf{s}}) + \mathbf{g}_1 - \mathbf{g}_0}{\tau}\right) \circ \Phi\right)$
- 8: **end for**
- 9: **end for**

output A sparse network $\{\mathbf{v}, \mathbf{m} \circ \Phi\}$ by sampling a mask \mathbf{m} from the distribution $p(\mathbf{m}|\mathbf{s})$.

D. Experimental Configurations

Dataset	C/FCM	CM	CO
GPUs	1	1	1
Epochs	1500	50	75
Weight Optimizer	Adam	SGD	SGD
Weight Learning Rate	0.0004	0.01	0.01
Weight Momentum	-	0.9	0.9
Probability Optimizer	Adam	Adam	Adam
Probability Learning Rate	6e-3	6e-3	6e-3
Penalty Weight	10000	10000	10000
Learning Rate Scheduler	Cosine	Cosine	Cosine

Table 6. C/FCM stands for ColoredMNIST and FullColoredMNIST. CM stands for CIFARMNIST. CO stands for ColoredObject.

Sparse Invariant Risk Minimization

Dataset	Training Accuracy						Testing Accuracy						
	Dim	64	160	256	390	512	640	64	160	256	390	512	640
Oracle		74.64	76.80	77.85	78.63	79.24	79.67	76.38	76.61	76.12	75.80	75.64	75.72
ERM		84.41	85.21	85.61	86.31	86.70	86.94	25.25	26.27	27.50	27.67	27.93	28.10
IRMv1	IRM	77.04	78.15	80.54	83.46	86.33	89.75	56.36	63.36	64.97	60.62	59.75	57.51
	MRM	76.56	76.57	78.19	84.52	87.5	90.04	62.5	64.69	65.49	62.23	59.47	55.91
	SparseIRM	77.27	76.74	78.13	78.43	78.09	79.16	65.06	67.30	68.24	68.88	69.20	69.47
REx	IRM	72.87	66.54	76.18	79.74	83.42	85.93	46.08	55.59	60.22	62.83	64.18	60.96
	MRM	68.77	69.49	74.62	74.86	72.71	70.22	53.44	59.33	66.77	67.08	66.53	61.79
	SparseIRM	65.83	74.69	75.31	76.74	76.27	77.28	63.50	71.26	71.54	71.61	72.37	72.83

Table 7. Comparison of Training and Testing Accuracy of MLP on FullColoredMNIST50000 with varying hidden dimensions.

[MLP and ResNet-18 Architectures] MLP consists of three linear layers, with weights of shape (392, hidden dimensions), (hidden dimensions, hidden dimensions) and (hidden dimensions, 1), respectively. The activation layer between linear layers is ReLU. Regular ResNet-18(He et al., 2016) architecture is adopted.

Dataset	ColoredMNIST20000						FullColoredMNIST20000						
	Dim	64	160	256	390	512	640	64	160	256	390	512	640
Oracle		66.89	67.26	68.96	69.97	70.31	71.62	68.34	69.07	69.43	70.50	70.97	71.27
ERM		23.50	25.56	24.32	25.38	24.76	24.83	27.84	30.29	31.09	31.43	31.16	31.45
IRMv1	IRM	55.62	62.42	53.62	47.28	45.73	36.80	51.37	54.19	52.13	46.81	43.21	41.18
	MRM	60.69	65.00	66.17	65.52	62.56	53.58	52.39	55.47	58.66	52.53	46.89	43.42
	SparseIRM	64.66	65.91	66.54	67.92	68.24	68.54	57.94	59.34	59.94	61.11	62.37	63.26
REx	IRM	51.90	62.51	62.57	56.09	52.08	47.43	45.71	55.77	56.30	51.48	44.76	43.21
	MRM	59.36	63.49	65.17	65.24	64.62	64.52	50.96	56.74	59.46	59.54	58.93	58.39
	SparseIRM	64.62	67.16	69.01	69.71	70.10	70.45	61.25	63.08	63.35	63.53	63.62	63.74

Table 8. Comparison of MLP on ColoredMNIST20000 and FullColoredMNIST20000 with varying hidden dimensions.

E. Additional Experimental Results

E.1. Training and Testing Accuracy of MLP on FullColoredMNIST50000 with varying hidden dimensions

Following we present additional experimental results on training accuracy, in order to present the superior power of SparseIRM in preventing overfitting caused by overparameterization. From Table 7, we find that besides the better testing accuracy of SparseIRM, our method also achieves lower training accuracy than Oracle, ERM and IRM. The gap between training and testing accuracy of IRM and MRM is far larger than that of our MRM in all settings. The experimental results on training and testing accuracy further demonstrate the power of SparseIRM in reducing overfitting caused by overparameterization.

E.2. MLP on ColoredMNIST20000 and FullColoredMNIST20000 with Varying Hidden Dimensions

Table 8 presents the detailed testing accuracy of MLP on ColoredMNIST20000 and FullColoredMNIST20000 with varying hidden dimensions, which are also presented in Figure 3. The results on ColoredMNIST20000 and FullColoredMNIST20000 demonstrate more superior performance compared with MRM.

F. Future Directions

SparseIRM stills needs to demonstrate its applicability to NLP tasks especially on today’s large pretraining language models (Devlin et al., 2018; Radford et al., 2019; Liu et al., 2019; Diao et al., 2019; Brown et al., 2020), cross-modal tasks (Gu et al., 2018; Gao et al., 2022; Zhou et al., 2022a), domain adaptation tasks (Diao et al., 2021; Huang et al., 2022), self-supervised

learning tasks (He et al., 2020; Grill et al., 2020; Chen et al., 2021b; Liu et al., 2022). It is also interesting to explore how SparseIRM interacts with other parallel domain generalization methods (Luo et al., 2018; Bai et al., 2021a;b), how it performs when applied with other sparse training methods (Shao et al., 2019; Kusupati et al., 2020) and how it performs on more challenging benchmarks (Ye et al., 2022).