Provable Domain Generalization via Invariant-Feature Subspace Recovery

Haoxiang Wang [1] **Haozhe Si** [1] **Bo Li** [1] **Han Zhao** [1]

Abstract

Domain generalization asks for models trained over a set of training environments to perform well in unseen test environments. Recently, a series of algorithms such as Invariant Risk Minimization (IRM) has been proposed for domain generalization. However, Rosenfeld et al. (2021) shows that in a simple linear data model, even if non-convexity issues are ignored, IRM and its extensions cannot generalize to unseen environments with less than d_s+1 training environments, where d_s is the dimension of the spurious-feature subspace. In this paper, we propose to achieve domain generalization with **I**nvariant-feature **S**ubspace **R**ecovery (ISR). Our first algorithm, ISR-Mean, can identify the subspace spanned by invariant features from the first-order moments of the class-conditional distributions, and achieve provable domain generalization with d_s+1 training environments under the data model of Rosenfeld et al. (2021). Our second algorithm, ISR-Cov, further reduces the required number of training environments to $\mathcal{O}(1)$ using the information of second-order moments. Notably, unlike IRM, our algorithms bypass non-convexity issues and enjoy global convergence guarantees. Empirically, our ISRs can obtain superior performance compared with IRM on synthetic benchmarks. In addition, on three real-world image and text datasets, we show that both ISRs can be used as simple yet effective post-processing methods to improve the worst-case accuracy of (pre-)trained models against spurious correlations and group shifts. The code is released at `https://github.com/Haoxiang-Wang/ISR`.

1 Introduction

Domain generalization, i.e., out-of-distribution (OOD) generalization, aims to obtain models that can generalize to unseen (OOD) test domains after being trained on a limited number of training domains (Blanchard et al., 2011; Wang et al., 2021b; Zhou et al., 2021; Shen et al., 2021). A series of works try to tackle this challenge by learning the so-called domain-invariant features (i.e., features whose distributions do not change across domains) (Long et al., 2015; Ganin et al., 2016; Hoffman et al., 2018; Zhao et al., 2018; 2019). On the other hand, Invariant Risk Minimization (IRM) (Arjovsky et al., 2019), represents another approach that aims to learn features that induce invariant optimal predictors over training environments. Throughout this work, we shall use the term *invariant features* to denote such features. There is a stream of follow-up works of IRM (Javed et al., 2020; Krueger et al., 2021; Shi et al., 2020; Ahuja et al., 2020; Khezeli et al., 2021), which propose alternative objectives or extends IRM to different settings.

Recently, some theoretical works demonstrate that IRM and its variants fail to generalize to unseen environments, or cannot outperform empirical risk minimization (ERM), in various simple data models (Rosenfeld et al., 2021; Kamath et al., 2021; Ahuja et al., 2021b). For instance, Rosenfeld et al. (2021) considers a simple Gaussian linear data model such that the class-conditional distribution of *invariant features* remains the same across domains, while that of *spurious features* changes across domains. Intuitively, a successful domain generalization algorithm is expected to learn an *optimal invariant predictor*, which relies on only the invariant features and is optimal over the invariant features. To remove the noise introduced by finite samples, these theoretical works generally assume that infinite samples are available per training environment to disregard finite-sample effects, and the main evaluation metric for domain generalization algorithms is the *number of training environments* needed to learn an optimal invariant predictor – this metric is also referred to as *environment complexity* in the literature (Chen et al., 2021). In the case of linear predictors, Rosenfeld et al. (2021) shows that IRM and REx (an alternative objective of IRM proposed in (Krueger et al., 2021)) need $E > d_s$ to learn optimal invariant predictors, where E is the number of training environments, and d_s is the dimension of spurious features. In the case of non-linear predictors, they both fail to learn invariant predictors. Notice that the $E > d_s$ condition of IRM can be interpreted as a *linear environment complexity* (i.e., $O(d_s)$ complexity),

[1]University of Illinois at Urbana-Champaign, Urbana, IL, USA. Correspondence to: Haoxiang Wang <hwang264@illinois.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

which is also observed in other recent works (Kamath et al., 2021; Ahuja et al., 2021b; Chen et al., 2021).

In this work, we propose a novel approach for domain generalization, Invariant-feature Subspace Recovery (ISR), that recovers the subspace spanned by the invariant features, and then fits predictors in this subspace. More concretely, we present two algorithms to realize this approach, ISR-Mean and ISR-Cov, which utilize the first-order and second-order moments (i.e., mean and covariance) of class-conditional distributions, respectively. Under the linear data model of Rosenfeld et al. (2021) with linear predictors, we prove that a) ISR-Mean is guaranteed to learn the optimal invariant predictor with $E \geqslant d_s + 1$ environment, matching the environment complexity of IRM, and b) ISR-Cov reduces the requirement to $E \geqslant 2$, achieving a constant $O(1)$ environment complexity. Notably, both of ISR-Mean and ISR-Cov require fewer assumptions on the data model than IRM, and they both enjoy global convergence guarantees, while IRM does not because of its non-convex formulation of the objective function. Notably, the ISRs are also more computationally efficient than algorithms such as IRM, since the computation of ISRs involves basically only the ERM with one additional call of an eigen-decomposition solver.

Empirically, we conduct studies on a set of challenging synthetic linear benchmarks designed by (Aubin et al., 2021) and a set of real-world datasets (two image datasets and one text dataset) used in Sagawa et al. (2019). Our empirical results on the synthetic benchmarks validate the claimed environment complexities, and also demonstrate its superior performance when compared with IRM and its variant. Since the real-world data are highly complex and non-linear, over which the ISR approach cannot be directly applied, we apply ISR on top of the features extracted by the hidden layers of trained neural nets as a post-processing procedure. Experiments show that ISR-Mean can consistently increase the worse-case accuracy of the trained models against spurious correlations and group shifts, and this includes models trained by ERM, reweighting and GroupDRO (Sagawa et al., 2019).

2 Related Work

Domain Generalization. Domain generalization (DG), also known as OOD generalization, aims at leveraging the labeled data from a limited number of training environments to improve the performance of learning models in unseen test environments (Blanchard et al., 2011). The simplest approach for DG is empirical risk minimization (Vapnik, 1992), which minimizes the sum of empirical risks over all training environments. Distributionally robust optimization is another approach (Sagawa et al., 2019; Volpi et al., 2018), which optimizes models over a worst-case distribution that is perturbed around the original distribution. Besides, there are two popular approaches, domain-invariant representation learning and invariant risk minimization, which we will discuss in detail below. In addition to algorithms, there are works that propose theoretical frameworks for DG (Zhang et al., 2021; Ye et al., 2021), or empirically examine DG algorithms over various benchmarks (Gulrajani & Lopez-Paz, 2021; Koh et al., 2021; Wiles et al., 2021). Notably, some recent works consider DG with temporarily shifted environments (Koh et al., 2021; Ye et al., 2022), which is a novel and challenging setting. Besides DG, there are other learning paradigms that involve multiple environments, such as multi-task learning (Caruana, 1997; Wang et al., 2021a) and meta-learning (Finn et al., 2017; Wang et al., 2022), which do not aim at generalization to OOD environments.

Domain-Invariant Representation Learning. Domain-Invariant representation learning is a learning paradigm widely applied in various tasks. In particular, in domain adaptation (DA), many works aim to learn a representation of data that has an invariant distribution over the source and target domains, adopting methods including adversarial training (Ganin et al., 2016; Tzeng et al., 2017; Zhao et al., 2018) and distribution matching (Ben-David et al., 2007; Long et al., 2015; Sun & Saenko, 2016). The domain-invariant representation approach for DA enjoys theoretical guarantees (Ben-David et al., 2010), but it is also pointed out that issues such as conditional shift should be carefully addressed (Zhao, 2019). In domain generalization (Blanchard et al., 2011), since there is no test data (even unlabelled ones) available, models are optimized to learn representations invariant over training environments (Albuquerque et al., 2020; Chen et al., 2021). Notice that many domain-invariant representation learning methods for DA can be easily applied to DG as well (Gulrajani & Lopez-Paz, 2021).

Invariant Risk Minimization. Arjovsky et al. (2019) proposes invariant risk minimization (IRM) that aims to learn invariant predictors over training environments by optimizing a highly non-convex bi-level objective. The authors also reduce the optimization difficulty of IRM by proposing a practical version, IRMv1, with a penalty regularized objective instead of a bi-level one. Alternatives of IRM have also been studied (Ahuja et al., 2020; Li et al., 2021). However, Rosenfeld et al. (2021); Kamath et al. (2021); Ahuja et al. (2021b) theoretically show that these algorithms fail even in simple data models.

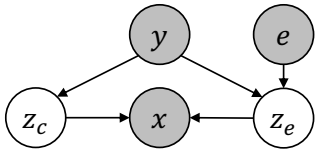

Figure 1. The causal graph of the data model in Rosenfeld et al. (2021). Shading represents that the variable is observed.

3 Problem Setup

Notations Each labeled example can be represented as a (x, y, e) tuple, where $x \in \mathbb{R}^d$ is the input, $y \in \{\pm 1\}$ is the label, and $e \in \mathbb{Z}_+$ is the index of the environment that provides (x, y). In addition, we assume x is generated by a latent feature $z \in \mathbb{R}^d$, which generates x and is correlated with y and e (e.g., see the example in Fig. 1). Besides, we use $X, Y, \mathscr{E}, Z$ to refer to random variables w.r.t. x, y, e, z.

Data Model In this paper, we adopt the linear Gaussian data model of Rosenfeld et al. (2021), which assumes that training data are drawn from E training environments, $\mathcal{E} = \{1, ..., E\}$. For arbitrary training environment $e \in \mathcal{E}$, each sample in this environment is generated by the following mechanism (see Fig. 1 for an illustration): first, a label $y \in \{\pm 1\}$ is sampled,

$$y = \begin{cases} 1, & \text{with probability } \eta \\ -1, & \text{otherwise} \end{cases} \tag{1}$$

Then, both invariant latent features z_c and spurious latent features z_e of this sample are drawn from the following Gaussian distributions:

$$z_c \sim \mathcal{N}(y\mu_c, \sigma_c^2 I) \in \mathbb{R}^{d_c}, z_e \sim \mathcal{N}(y\mu_e, \sigma_e^2 I) \in \mathbb{R}^{d_s} \tag{2}$$

where $\mu_c \in \mathbb{R}^{d_c}, \mu_e \in \mathbb{R}^{d_s}$ and $\sigma_c, \sigma_e \in \mathbb{R}_+$. The constants d_c and d_s refer to the dimension of invariant features and spurious features, respectively. The total number of feature attributes is then $d = d_c + d_s$. Notice that μ_c, σ_c are invariant across environments, while μ_e, σ_e are dependent on the environment index e. Following Rosenfeld et al. (2021), we name $\{\mu_e\}$ and $\{\sigma_e\}$ as *environmental* means and variances.

Rosenfeld et al. (2021) adopts a mild non-degeneracy assumption[1] on the environmental mean from the IRM paper (Arjovsky et al., 2019), stated as Assumption 1 below. In addition, the authors also make another non-degeneracy assumption[2] on the environmental variances, which we relax to the following Assumption 2.

Assumption 1. *For the set of environmental means, $\{\mu_e\}_{e=1}^E$, we assume that each element of the set cannot be expressed as an affine combination of the rest elements.*

Assumption 2. *Assume there exists a pair of distinct training environments $e, e' \in [E]$ such that $\sigma_e \neq \sigma_{e'}$.*

With the latent feature z as a concatenation of z_c and z_e, the observed sample x is generated by a linear transformation on this latent feature. For simplicity, we consider that x has the same dimension as z.

$$z = \begin{bmatrix} z_c \\ z_e \end{bmatrix} \in \mathbb{R}^d, \quad x = Rz = Az_c + Bz_e \in \mathbb{R}^d \tag{3}$$

where $d = d_c + d_s$, and $A = \mathbb{R}^{d \times d_c}, B = \mathbb{R}^{d \times d_s}$ are fixed transformation matrices with concatenation as $R = [A, B] \in \mathbb{R}^{d \times d}$. Then, each observed sample x is effectively a sample drawn from

$$\mathcal{N}(y(A\mu_c + B\mu_e), \sigma_c^2 AA^\mathsf{T} + \sigma_e^2 BB^\mathsf{T}) \tag{4}$$

The following assumption is also imposed on the transformation matrix in Rosenfeld et al. (2021):

Assumption 3. *R is injective.*

Since $R \in \mathbb{R}^{d \times d}$, Assumption 3 leads to the fact $\mathrm{rank}(R) = d$, indicating that R is full-rank.

Denote the data of any training domain e as $\mathcal{D}_e$. During training, learners have access to the environment index e for each training sample, i.e., learners observe samples in the form of (x, y, e).

Optimal Invariant Predictors The quest of IRM is to find the optimal invariant predictors, i.e., classifiers that use only invariant features and are optimal w.r.t. invariant features over the training data. In the data model of Rosenfeld et al. (2021), because of the linear nature of the data generation process, the optimal invariant predictors are contained in the linear function class. Since the task of consideration is binary classification, Rosenfeld et al. (2021) chooses the logistic loss as the loss function for optimization[3], which we also adopt in this work. Then, the goal of domain generalization in this data model is to learn a linear featurizer (feature extractor) Φ and a linear classifier β that minimizes the risk (population loss) on any unseen environment e with data distribution p_e satisfying Assumptions (1)-(3):

$$\mathcal{R}^e(\Phi, \beta) := \mathbb{E}_{(x,y)\sim p^e}\left[\ell\left(w^\mathsf{T}\Phi(x) + b,\ y\right)\right] \tag{5}$$

where ℓ is the logistic loss function, and $\beta = (w, b)$ with weight w and bias b.

To be complete, we present the optimal invariant predictor derived by Rosenfeld et al. (2021) as follows.

Proposition 1 (Optimal Invariant Predictor)**.** *Under the data model considered in Eq. (1)-(3), the optimal invariant predictor h^* w.r.t. logistic loss is unique, which can be expressed as a composition of i) a featurizer Φ^* that recovers the invariant features and ii) the classifier $\beta^* = (w^*, b^*)$ that is optimal w.r.t. the extracted features:*

$$h^*(x) = {w^*}^\mathsf{T}\Phi^*(x) + b^* \tag{6}$$

$$\Phi^*(x) := \begin{bmatrix} I_{d_c} & 0 \\ 0 & 0 \end{bmatrix} R^{-1}x = \begin{bmatrix} z_c \\ 0 \end{bmatrix} \in \mathbb{R}^{d \times d} \tag{7}$$

$$w^* := \begin{bmatrix} 2\mu_c/\sigma_c^2 \\ 0 \end{bmatrix} \in \mathbb{R}^d, \quad b^* := \log\frac{\eta}{1-\eta} \in \mathbb{R} \tag{8}$$

Notice that even though the optimal invariant predictor h^* is unique, its components (the featurizer and classifier) are only unique up to invertible transformations. For instance,

[1] It was stated as (9) in Rosenfeld et al. (2021).

[2] It is stated as Eq. (10) in Rosenfeld et al. (2021), which is a sufficient (not necessary) condition for our Assumption 2.

[3] Rosenfeld et al. (2021) proves that logistic loss over linear models can attain Bayes optimal classifiers in this data model.

Algorithm 1 ISR-Mean

Input: Data of all training environments, $\{\mathcal{D}_e\}_{e\in[E]}$.
for $e = 1, 2, \ldots, E$ **do**
  Estimate the sample mean of $\{x|(x, y) \in \mathcal{D}_e, y = 1\}$ as $\bar{x}_e \in \mathbb{R}^d$
end for
I. Construct a matrix $\mathcal{M} \in \mathbb{R}^{E\times d}$ with the e-th row as $\bar{x}_e^\mathsf{T}$ for $e \in [E]$
II. Apply PCA to $\mathcal{M}$ to obtain eigenvectors $\{P_1, ..., P_d\}$ with eigenvalues $\{\lambda_1, ..., \lambda_d\}$
III. Stack d_c eigenvectors with the lowest eigenvalues to obtain a transformation matrix $P' \in \mathbb{R}^{d_c\times d}$
IV. Fit a linear classifier (with $w \in \mathbb{R}^{d_c}$, $b \in \mathbb{R}$) by ERM over all training data with transformation $x \mapsto P'x$
Obtain a predictor $f(x) = w^\mathsf{T}P'x + b$

$(w^{*\mathsf{T}}U^{-1})(U\Phi) = w^{*\mathsf{T}}\Phi$ for any invertible $U \in \mathbb{R}^{d\times d}$.

Invariant Risk Minimization IRM optimizes a bi-level objective over a featurizer Φ and a classifier β,

$$\text{IRM}: \min_{\Phi,\beta} \sum_{e\in[E]} \mathcal{R}^e(\Phi, \beta) \quad (9)$$
$$\text{s.t. } \beta \in \arg\min_{\beta} \mathcal{R}^e(\Phi, \beta) \ \ \forall e \in [E]$$

This objective is non-convex and difficult to optimize. Thus, Arjovsky et al. (2019) proposes a Langrangian form to find an approximate solution,

$$\text{IRMv1}: \min_{\Phi,\hat{\beta}} \sum_{e\in[E]} \mathcal{R}^e(\Phi, \hat{\beta}) + \lambda \left\|\nabla_{\hat{\beta}}\mathcal{R}^e(\Phi, \hat{\beta})\right\|_2^2 \quad (10)$$

where $\lambda > 0$ controls the regularization strength. Notice that the IRMv1(10) is still non-convex, and it becomes equivalent to the original IRM (9) as $\lambda \to \infty$.

Environment Complexity To study the dependency of domain generalization algorithms on environments, recent theoretical works (Rosenfeld et al., 2021; Kamath et al., 2021; Ahuja et al., 2021a; Chen et al., 2021) consider the ideal setting of infinite data per training environment to remove the finite-sample effects. In this infinite-sample setting, a core measure of domain generalization algorithms is *environment complexity*: the number of training environments needed to learn an invariant optimal predictor. For this data model, Rosenfeld et al. (2021) proves that with the linear Φ and β, the environment complexity of IRM is $d_s + 1$, assuming the highly non-convex objective (9) is optimized to reach the global optimum. This linear environment complexity (i.e., $O(d_s)$) of IRM is also proved in (Kamath et al., 2021; Ahuja et al., 2021a) under different simple data models.

4 Invariant-Feature Subspace Recovery

In this section, we introduce two algorithms, ISR-Mean and ISR-Cov, which recover the invariant-feature subspace with the first-order and second-order moments of class-conditional data distributions, respectively.

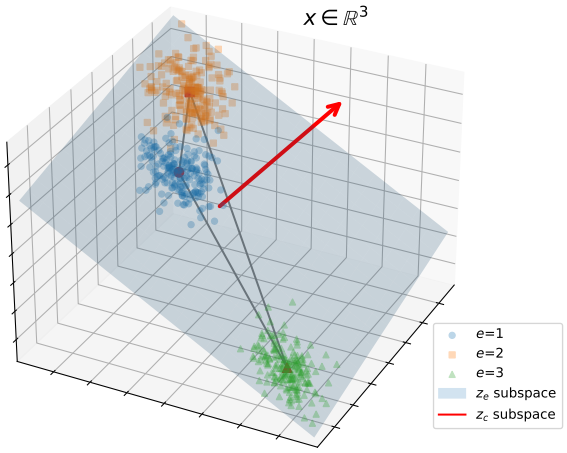

Figure 2. An example for ISR-Mean with d_c=1, d_s=2, E=3. In this $\mathbb{R}^3$ input space, the blue 2D plane is determined by sample means of positive-class samples of the 3 training environments.

4.1 ISR-Mean

Algorithm 1 shows the pseudo-code of ISR-Mean, and we explain its four main steps in detail below. In the setup of Section 3, ISR-Mean enjoys a linear environment complexity that matches that of IRM, while requiring fewer assumptions (no need for Assumption 2).

I. Estimate Sample Means across Environments In any training environment e, each observed sample $x \in \mathbb{R}^d$ is effectively drawn i.i.d. from $\mathcal{N}(y(A\mu_c + B\mu_e), AA^\mathsf{T}\sigma_c^2 + BB^\mathsf{T}\sigma_e^2)$, as stated in (4). In the infinite-sample setting considered in Section 3, the mean of the positive-class data in environment e can be expressed as $\mathbb{E}[X|Y = 1, \mathscr{E} = e]$, which is exactly the value of $\bar{x}_e$ in Algorithm 1. Thus, we know $\bar{x}_e$ satisfies $\bar{x}_e = A\mu_c + B\mu_e$, and the matrix $\mathcal{M}$ can be expressed as

$$\mathcal{M} := \begin{bmatrix} \bar{x}_1^\mathsf{T} \\ \vdots \\ \bar{x}_E^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \mu_c^\mathsf{T}A^\mathsf{T} + \mu_1^\mathsf{T}B^\mathsf{T} \\ \vdots \\ \mu_c^\mathsf{T}A^\mathsf{T} + \mu_E^\mathsf{T}B^\mathsf{T} \end{bmatrix} = \overbrace{\begin{bmatrix} \mu_c^\mathsf{T} & \mu_1^\mathsf{T} \\ \vdots & \vdots \\ \mu_c^\mathsf{T} & \mu_E^\mathsf{T} \end{bmatrix}}^{\mathcal{U}^\mathsf{T} :=} R^\mathsf{T} \quad (11)$$

II. PCA on $\mathcal{M}$. In this step, we apply principal component analysis (PCA) (Pearson, 1901) to the matrix $\mathcal{M}$. First, PCA performs mean-substraction on $\mathcal{M}$ to shift the sample mean of each column to zero, and we denote the shifted matrix as $\widetilde{\mathcal{M}}$. Then, PCA eigen-decompose $\widehat{\Sigma}_\mathcal{M} := \frac{1}{E}\widetilde{\mathcal{M}}^\mathsf{T}\widetilde{\mathcal{M}}$, the sample covariance matrix of $\widetilde{\mathcal{M}}$, such that $\widehat{\Sigma}_\mathcal{M} = P^\mathsf{T}SP$, where $P = [P_1, \ldots, P_d] \in \mathbb{R}^{d\times d}$ is a stack of eigenvectors $\{P_i\}_{i\in d}^d$, and $S \in \mathbb{R}^{d\times d}$ is a diagonal square matrix with diagonal entries as eigenvalues $\{\lambda_i\}_{i=1}^d$ of $\widehat{\Sigma}_\mathcal{M}$. We consider the eigenvalues $\{\lambda_i\}_{i=1}^d$ are sorted in ascending order.

III. Recover the Invariant-Feature Subspace As we shall formally prove in Theorem 1, in the infinite-sample setting, a) the eigenvalues $\{\lambda_i\}_{i\in 1}^d$ should exhibit a "phase

transition" phenomenon such that the first d_c eigenvalues all are *zeros* while the rest are all *positive*, b) the d_c eigenvectors corresponding to zero eigenvalues, $\{P_1, \ldots, P_{d_c}\}$, are guaranteed to recover the d_c-dimensional subspace spanned by invariant latent feature dimensions, i.e., the subspace of z_c defined in (2). We stack these eigenvectors as a matrix P'

$$P' := [P_1, \ldots, P_{d_c}]^\mathsf{T} \in \mathbb{R}^{d_c \times d} \tag{12}$$

IV. Train a Classifier in the Invariant-Feature Subspace In this final step, we just transform all training data by the transformation $x \mapsto P'x$, and fit a linear classifier with ERM to the transformed data to obtain an predictor,

$$f(x) = w^\mathsf{T} P' x + b \tag{13}$$

which is guaranteed to be the optimal invariant predictor h^* defined in Proposition 1, i.e., $f \equiv h^*$.

Global Convergence Guarantee ISR-Mean is guaranteed to converge to a global optimum since a) the step I and III are optimization-free, b) PCA can be efficiently optimized to global the optimum by various methods (Arora et al., 2012; Vu et al., 2013; Hauser et al., 2018; Eftekhari & Hauser, 2020), c) the ERM objective of linear classifiers with logistic loss is convex, enjoying global convergence.

Geometric Interpretation. We provide an geometric interpretation of ISR-Mean with a 3D example in Fig. 2, where d_c=1, d_s=2, E=3. For each environment e, the sample mean of its positive-class data, $\bar{x}_e$, must lie in a d_s-dimensional spurious-feature subspace in the infinite-sample setting, as proved by Theorem 1. ISR-Mean aims to identify this spurious-feature subspace, and take its tangent subspace as the invariant-feature subspace.

Linear Environment Complexity In the infinite-sample setting, we prove below that with more than d_s training environments, ISR-Mean is guaranteed to learn the invariant optimal predictor (Theorem 1). Notice that even though this linear environment complexity is identical to that of IRM (proved in Theorem 5.1 of Rosenfeld et al. (2021)), our ISR-Mean has two additional advantages: (a) Unlike IRM, ISR-Mean does not require any assumption on the covariance[4] such as Assumption 2; (b) ISR-Mean enjoys the global convergence guarantee, while IRM does not due to its non-convex formulation. The proof is in Appendix A.

Theorem 1 (ISR-Mean). *Suppose $E > d_s$ and the data size of each environment is infinite, i.e., $|\mathcal{D}_e| \to \infty$ for e=1, ..., E. For PCA on the $\mathcal{M}$ defined in (11), the obtained eigenvectors $\{P_1, \ldots, P_d\}$ with corresponding ascendingly ordered eigenvalues $\{\lambda_1, \ldots, \lambda_d\}$ satisfy*

$$\forall 1 \leqslant i \leqslant d_c,\ \lambda_i = 0 \quad \textit{and} \quad \forall d_c < i \leqslant d,\ \lambda_i > 0$$

The eigenvectors corresponding to these zero eigenvalues, i.e., $\{P_1, \ldots, P_{d_c}\}$, can recover the subspace spanned by the invariant latent feature dimensions, i.e.,

$$\mathrm{Span}(\{P_1^\mathsf{T} R, \ldots, P_{d_c}^\mathsf{T} R\}) = \mathrm{Span}(\{\hat{\mathbf{1}}, \ldots, \hat{\mathbf{d}}_\mathbf{c}\}) \tag{14}$$

where $\hat{\mathbf{i}}$ is the unit-vector along the i-th coordinate in the latent feature space for $i = 1, \ldots, d$. Then, the classifier f fitted with ERM to training data transformed by $x \mapsto [P_1, \ldots, P_{d_c}]^\mathsf{T} x$ is guaranteed to be the invariant optimal predictor, i.e., $f = h^*$, where h^* is defined in (6).

4.2 ISR-Cov

The pseudo-code of ISR-Cov[5] is presented in Algorithm 2, with a detailed explanation below. In the setup of Section 3, ISR-Cov attains an $O(1)$ environment complexity, the optimal complexity any algorithm can hope for, while requiring fewer assumptions than IRM (no need for Assumption 1).

I. Estimate and Select Sample Covariances across Environments As (4) indicates, in any environment e, each observed sample $x \in \mathbb{R}^d$ with $y = 1$ is effectively drawn i.i.d. from $\mathcal{N}(A\mu_c + B\mu_e, AA^\mathsf{T}\sigma_c^2 + BB^\mathsf{T}\sigma_e^2)$. Thus, the covariance of the positive-class data in environment e can be expressed as $\mathrm{Cov}[X|Y=1, \mathscr{E}=e] = AA^\mathsf{T}\sigma_c^2 + BB^\mathsf{T}\sigma_e^2$, which is the value that Σ_e in the step I of Algorithm 2 estimates. The estimation is exact in the infinite-sample setting of consideration, so we have $\Sigma_e = AA^\mathsf{T}\sigma_c^2 + BB^\mathsf{T}\sigma_e^2$. Assumption 2 guarantees that we can select a pair of environments e_1, e_2 with $\Sigma_1 \neq \Sigma_2$. Then, we have

$$\Delta\Sigma := \Sigma_{e_1} - \Sigma_{e_2} = (\sigma_{e_1}^2 - \sigma_{e_2}^2) BB^\mathsf{T} \in \mathbb{R}^{d \times d} \tag{15}$$

II. Eigen-decompose $\Delta\Sigma$ Similar to the step II of Algorithm 1 explained in Sec. 4.1, we eigen-decompose $\Delta\Sigma$ to obtain eigenvectors $\{P_i\}_{i\in d}^d$ corresponding to eigenvalues $\{\lambda_i\}_{i=1}^d$. We consider the eigenvalues are sorted in ascending order by their *absolute values*.

III. Recover the Invariant-Feature Subspace As we shall formally prove in Theorem 2, in the infinite-sample setting, a) the eigenvalues $\{\lambda_i\}_{i\in 1}^d$ should exhibit a "phase transition" phenomenon such that the first d_c eigenvalues all are *zeros* while the rest are all *non-zero*, b) the d_c eigenvectors corresponding to zero eigenvalues, $\{P_1, \ldots, P_{d_c}\}$, are guaranteed to recover the d_c-dimensional invariant-feature subspace. We stack these eigenvectors as a matrix P'

$$P' := [P_1, \ldots, P_{d_c}]^\mathsf{T} \in \mathbb{R}^{d_c \times d} \tag{16}$$

IV. Train a Classifier in the Invariant-Feature Subspace This final step is the same as the step IV of Algorithm 1 described in Sec. 4.1.

[4] IRM needs a covariance assumption stronger than our Assumption 2, as pointed out in Sec. 3.

[5] In the final stage of this paper preparation, we notice a concurrent work, the v2 of Chen et al. (2021) (uploaded to arXiv on Nov 22, 2021), appends a new algorithm in its Appendix C that is similar to our ISR-Cov, under data model assumptions stricter than ours. That algorithm does not exist in their v1.

Algorithm 2 ISR-Cov

Input: Data of all training environments, $\{\mathcal{D}_e\}_{e\in[E]}$.
for $e = 1, 2, \ldots, E$ **do**
  Estimate the sample covriance of $\{x|(x, y) \in \mathcal{D}_e, y = 1\}$ as $\Sigma_e \in \mathbb{R}^{d\times d}$
end for
I. Select a pair of environments e_1, e_2 such that $\Sigma_1 \neq \Sigma_2$, and compute their difference, $\Delta\Sigma := \Sigma_{e_1} - \Sigma_{e_2}$
II. Eigen-decompose $\Delta\Sigma$ to obtain eigenvectors $\{P_1, ..., P_d\}$ with eigenvalues $\{\lambda_1, ..., \lambda_d\}$
III. Stack d_c eigenvectors of eigenvalues with lowest absolute values to obtain a matrix $P' \in \mathbb{R}^{d_c\times d}$
IV. Fit a linear classifier (with $w \in \mathbb{R}^{d_c}$, $b \in \mathbb{R}$) by ERM over all training data with transformation $x \mapsto P'x$
Obtain a predictor $f(x) = w^\mathsf{T} P'x + b$

Global Convergence Applying the same argument in Sec. 4.1, it is clear that ISR-Cov also enjoys the global convergence guarantee: the eigen-decomposition and ERM can both be globally optimized.

Improving the Robustness of ISR-Cov In practice with finite data, Algorithm 2 may be non-robust as σ_e and $\sigma_{e'}$ become close to each other. The noise due to finite samples could obfuscate the non-zero eigenvalues so that they are indistinguishable from the zero eigenvalues. To mitigate such issues, we propose a robust version of ISR-Cov that utilizes more pairs of the given environments. Briefly speaking, the robust version is to **a)** run the step I to III of ISR-Cov over $N \leqslant \binom{E}{2}$ pairs of environments with distinct sample covariances, leading to N d_c-dimensional subspaces obtained through Algorithm 2, **b)** we find the stable d_c-dimensional subspace, and use it as the invariant-feature subspace that we train the following classifier. Specifically, we achieve b) by computing the flag-mean (Marrinan et al., 2014) over the set of P' (defined in (16)) obtained from the N selected pairs of environments. Compared with the original ISR-Cov, this robust version makes use of training data more efficiently (e.g., it uses more than one pair of training environments) and is more robust in the finite-data case. We implement this robust version of ISR-Cov in our experiments in Sec. 5.

Geometric Interpretation We provide an geometric interpretation of ISR-Cov with a 3D example in Fig. 3, where d_c=1, d_s=2, E=2. For either environment $e \in \{1, 2\}$, the covariance of its class-conditional latent-feature distribution, $\begin{bmatrix}\sigma_c^2 I_{d_c} & 0\\ 0 & \sigma_e^2 I_{d_s}\end{bmatrix}$, is *anisotropic*: the variance σ_c along invariant-feature dimensions is constant, while σ_e along the spurious-feature dimensions is various across $e \in \{1, 2\}$ (ensured by Assumption 2). Though the transformation R is applied to latent features, ISR-Cov still can identify the subspace spanned by invariant-feature dimensions in the latent-feature space by utilizing this anisotropy property.

$\mathcal{O}(1)$ **Environment Complexity** In the infinite-sample setting, we prove below that as long as there are at least two training environment that satisfies Assumption 2 and 3, ISR-Cov is guaranteed to learn the invariant optimal predictor. This is the minimal possible environment complexity, since spurious and invariant features are indistinguishable with only one environment. Notably, unlike IRM, a) ISR-Cov does not require Assumption 1, and b) ISR-Cov has a global convergence guarantee. The proof is in Appendix A.

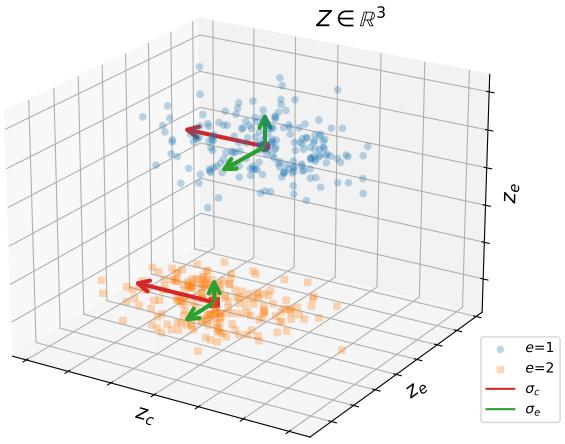

Figure 3. An example for ISR-Cov, where d_c=1, d_s=2, E=2. In this latent feature space of $z \in \mathbb{R}^3$, there is one dimension of z_c and the rest two of z_e.

Theorem 2 (ISR-Cov). *Suppose $E \geqslant 2$ and the data size of each environment is infinite, i.e., $|\mathcal{D}_e|\to\infty$ for e=1, ..., E. Eigen-decomposing $\Delta\Sigma$ defined in (15), the obtained eigenvectors $\{P_1, \ldots, P_d\}$ with corresponding eigenvalues $\{\lambda_1, \ldots, \lambda_d\}$ (ascendingly ordered by absolute values) satisfy*

$$\forall 1 \leqslant i \leqslant d_c,\ \lambda_i = 0 \quad \textit{and} \quad \forall d_c < i \leqslant d,\ \lambda_i \neq 0$$

The eigenvectors corresponding to these zero eigenvalues, i.e., $\{P_1, \ldots, P_{d_c}\}$, can recover the subspace spanned by the invariant latent feature dimensions, i.e.,

$$\mathrm{Span}(\{P_1^\mathsf{T} R, \ldots, P_{d_c}^\mathsf{T} R\}) = \mathrm{Span}(\{\hat{\mathbf{1}}, \ldots, \hat{\mathbf{d}}_\mathbf{c}\}) \quad (17)$$

where $\hat{\mathbf{i}}$ is the unit-vector along the i-th coordinate in the latent feature space for $i = 1, \ldots, d$. Then, the classifier f fitted with ERM to training data transformed by $x \mapsto [P_1, \ldots, P_{d_c}]^\mathsf{T} x$ is guaranteed be the invariant optimal predictor, i.e., $f = h^*$, where h^* is defined in (6).

5 Experiments

We conduct experiments on both synthetic and real datasets to examine our proposed algorithms.

5.1 Synthetic Datasets: Linear Unit-Tests

We adopt a set of synthetic domain generalization benchmarks, Linear Unit-Tests (Aubin et al., 2021), which is proposed by authors of IRM and is used in multiple recent works (Koyama & Yamaguchi, 2020; Khezeli et al., 2021; Du et al., 2021). Specifically, we take four classification

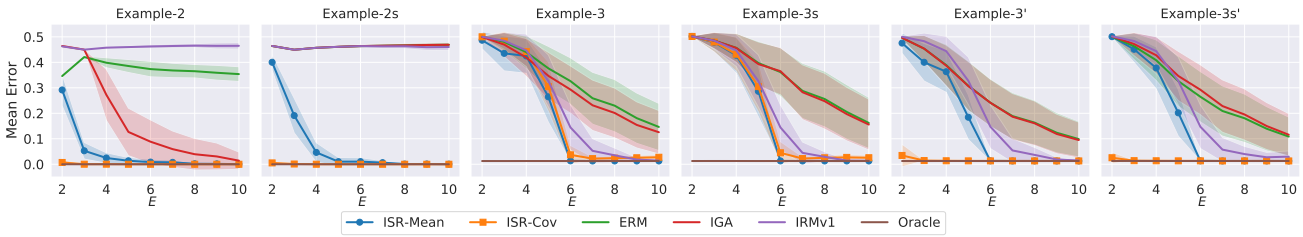

Figure 4. Test results on Linear Unit-Tests (first 4 plots) and its variants (last 2 plots), where $d_c = 5, d_s = 5$, and $E = 2, \ldots, 10$.

benchmarks[6] from the Linear Unit-Tests, which are named by Aubin et al. (2021) as Example-2/2s/3/3s. Example-2 and 3 are two binary classification tasks of Gaussian linear data generated in processes similar to the setup of Sec. 3, and they have identity transformation, $R = I$ (see the definition of R in (3)), while Example 2s/3s are their counterparts with R as a random transformation matrix. However, Example-3/3s do not satisfy Assumption 2, thus cannot properly examine our ISR-Cov. Hence, we construct variants of Example-3/3s satisfying Assumption 2, which we name as Example-3'/3s', respectively. We provide specific details of these benchmarks in Appendix B.1.

Example-2: The data generation process for Example-2 follows the Structual Causal Model (Peters et al., 2015), where $P(Y|\mu_c)$ is invariant across environments.

Example-3: It is similar to our Gaussian setup in 3, except that $\sigma_e \equiv \sigma_c = 0.1$, breaking Assumption 2. In this example, $P(\mu_c|Y)$ is invariant across environments

Example-3': We modify Example-3 slightly such that $\sigma_c = 0.1$ and $\sigma_e \sim \text{Unif}(0.1, 0.3)$. All the rest settings are identical to Example-3.

Example-2s/3s/3s': An random orthonormal projection matrix $R = [A,\ B] \in \mathbb{R}^{d\times d}$ (see the definition in (3)) is applied to the original Example-2/3/3' to scramble the invariant and spurious latent feature, leading to Example-2s/3s/3s' with observed data in the form of $x = Az_c + Bz_e$.

Implementation For baseline algorithms, we directly adopt their implementations by Aubin et al. (2021). We implement ISRs following Algorithm 1 and 2, where the last step of fitting predictors is done by the ERM implementation of Aubin et al. (2021), which optimizes the logistic loss with an Adam optimizer (Kingma & Ba, 2015). More details are provided in Appendix B.

Evaluation Procedures Following Aubin et al. (2021), we fix d=10, d_c=5, d_s=5, and increase E, the number of training environments, from 2 to 10, with 10K observed samples per environment. Each algorithm trains a linear predictor on these training data, and the predictor is evaluated in E test environments, each with 10K data. The test environments are generated analogously to the training ones, while the spurious features z_e are randomly shuffled across examples within each environment. The mean classification error of the trained predictor over E test environments is evaluated.

Empirical Comparisons We compare our ISRs with several algorithms implemented in Aubin et al. (2021), including IRMv1, IGA (an IRM variant by Koyama & Yamaguchi (2020)), ERM and Oracle (the optimal invariant predictor) on the datasets. We repeat the experiments over 50 random seeds and plot the mean errors of the algorithms. Fig. 4 shows the results of our experiment on these benchmarks: **a)** On Example-2/2s, our ISRs reach the oracle performance with a small E (number of training environments), significantly outperforming other algorithms. **b)** On Example-3/3s, ISRs reach the oracle performance as $E > 5 = d_s$, while IRM or others need more environments to match the oracle. **c)** On Example-3'/3s', ISR-Cov matches the oracle as $E \geqslant 2$, while the performance of all other algorithms does not differ much from that of Example-3/3s.

Conclusions Observing these results, we can conclude that: **a)** ISR-Mean can stably match the oracle as $E > d_s$, validating the environment complexity proved in Theorem 1. **b)** ISR-Cov matches the oracle as $E \geqslant 2$ in datasets satisfying Assumption 2 (i.e., Example-2/2s/3'/3s'), corroborating its environment complexity proved in Theorem 2.

5.2 Real Datasets

We adopt three datasets that Sagawa et al. (2019) proposes to study the robustness of models against spurious correlations and group shifts. See Fig. 5 for a demo of these datasets. Each dataset has multiple spurious attributes, and we treat each spurious attribute as a single environment.

Waterbirds (Sagawa et al., 2019): This is a image dataset built from the CUB (Wah et al., 2011) and Places (Zhou et al., 2017) datasets. The task of this dataset is the classification of waterbirds vs. landbirds. Each image is labelled with class $y \in \mathcal{Y} = \{$*waterbird, landbird*$\}$ and environment $e \in \mathcal{E} = \{$*water background, land background*$\}$. Sagawa et al. (2019) defines 4 groups[7] by $\mathcal{G} = \mathcal{Y} \times \mathcal{E}$. There are

[6] The rest are regression benchmarks, which we do not study.

[7] Notice that the definition of environment in this paper is dif-

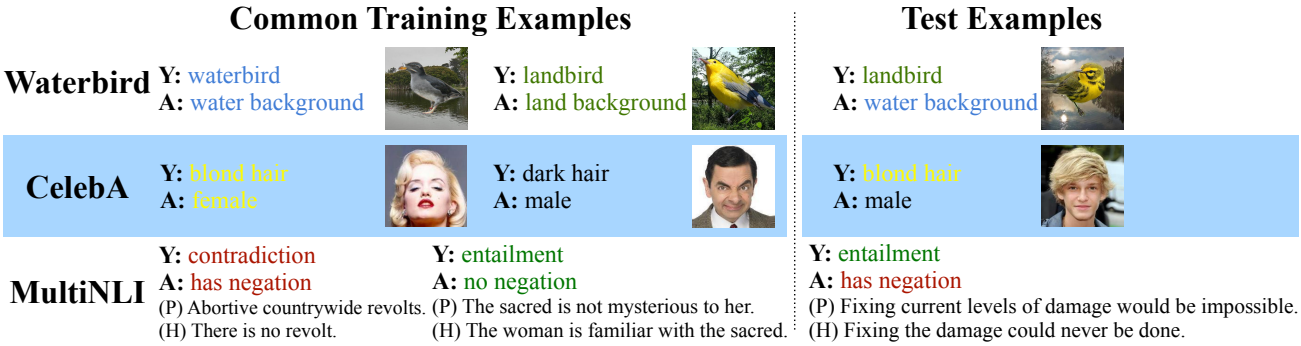

Figure 5. Representative examples of the three real datasets we use. The spurious correlation between the label (**Y**) and the attribute (**A**) in the training data does not hold in the test data.

Dataset	Backbone	Algorithm	Average Accuracy			Worst-Group Accuracy		
			Original	ISR-Mean	ISR-Cov	Original	ISR-Mean	ISR-Cov
Waterbirds	ResNet-50	ERM	86.66±0.67	87.87±0.80	**90.47±0.33**	62.93±5.37	76.10±1.11	**82.46±0.55**
		Reweighting	91.49±0.46	**91.77±0.52**	91.63±0.44	87.69±0.53	88.02±0.42	**88.67±0.55**
		GroupDRO	92.01±0.33	91.74±0.35	**92.25±0.27**	90.79±0.47	90.42±0.61	**91.00±0.45**
CelebA	ResNet-50	ERM	**95.12±0.34**	94.34±0.12	90.12±2.59	46.39±2.42	55.39±6.13	**79.73±5.00**
		Reweighting	**91.45±0.50**	91.38±0.51	91.24±0.35	84.44±1.66	**90.08±0.50**	88.84±0.57
		GroupDRO	**91.82±0.27**	91.82±0.27	91.20±0.23	88.22±1.67	**90.95±0.32**	90.38±0.42
MultiNLI	BERT	ERM	**82.48±0.40**	82.11±0.18	81.28±0.52	65.95±1.65	72.60±1.09	**74.21±2.55**
		Reweighting	**80.82±0.79**	80.53±0.88	80.73±0.90	64.73±0.32	**67.87±0.21**	66.34±2.46
		GroupDRO	**81.30±0.23**	81.21±0.24	81.20±0.24	78.43±0.87	**78.95±0.95**	78.91±0.75

Table 1. Test accuracy(%) with standard deviation of ERM, Re-weighting and GroupDRO over three datasets. We compare the accuracy of original trained classifiers vs. ISR-Mean post-processed classifiers. The average accuracy and the worst-group accuracy are both presented. Bold values mark the higher accuracy over Original vs. ISR-Mean for a given algorithm (e.g., ERM) and a specific metric (e.g., Average Acc.).

4795 training samples, and smallest group (waterbirds on land) only has 56.

CelebA (Liu et al., 2015): This is a celebrity face dataset of 162K training samples. Sagawa et al. (2019) considers a hair color classification task ($\mathcal{Y} = \{blond, dark\}$) with binary genders as spurious attributes (i.e., $\mathcal{E} = \{male, female\}$). Four groups are defined by $\mathcal{G} = \mathcal{Y} \times \mathcal{E}$, where the smallest group (blond-haired males) has only 1387 samples.

MultiNLI (Williams et al., 2017): This is a text dataset for natural language inference. Each sample includes two sentences, a hypothesis and a premise. The task is to identify if the hypothesis is contradictory to, entailed by, or neutral with the premise ($\mathcal{Y} = \{contradiction, neutral, entailment\}$). Gururangan et al. (2018) observes a spurious correlation between y=*contradiction* and negation words such as nobody, no, never, and nothing. Thus $\mathcal{E}$=$\{no\ negation, negation\}$ are spurious attributes (also environments), and 6 groups are defined by $\mathcal{G}$=$\mathcal{Y}\times\mathcal{E}$. There are 20K training data, while the smallest group (entailment with negations) has only 1521.

ferent from the definition of group in Sagawa et al. (2019).

Implementation We take three algorithms implemented by Sagawa et al. (2019): ERM, Reweighting, and GroupDRO. First, for each dataset, we train neural nets with these algorithms using the code and optimal hyper-parameters provided by Sagawa et al. (2019) implementation, and early stop models at the epoch with the highest worst-group validation accuracy. Then, we use the hidden-layers of the trained models to extract features of training data, and fit ISR-Mean/Cov to the extracted features. Finally, we replace the original last linear layer with the linear classifier provided by ISR-Mean/Cov, and evaluate it in the test set. More details are provided in Appendix B.

Empirical Comparisons We compare trained models with the original classifier vs. ISR-Mean/Cov post-processed classifiers over three datasets. Each experiment is repeated over 10 random seeds. From the results in Table 1, we can observe that: **a)** ISRs can improve the worst-group accuracy of trained models across all dataset-algorithm choices. **b)** Meanwhile, the average accuracy of ISR-Mean/Cov is maintained around the same level as the original classifier.

Dataset	Backbone	Algorithm	Average Accuracy			Worst-Group Accuracy		
			Linear Probing	ISR-Mean	ISR-Cov	Linear Probing	ISR-Mean	ISR-Cov
Waterbirds	CLIP (ViT-B/32)	ERM	76.42±0.00	**90.27±0.09**	76.80±0.01	52.96±0.00	**71.75±0.39**	55.76±0.00
		Reweighting	87.38±0.09	**88.23±0.12**	88.07±0.05	82.51±0.27	**85.13±0.22**	83.33±0.00

Table 2. Evaluation with CLIP-pretrained vision transformers. We compare ISR-Mean/ISR-Cov vs. linear probing in the Waterbird dataset, and report the test accuracy (%) with standard deviation.

5.2.1 Reduced Requirement of Environment Labels

Algorithms such as GroupDRO are successful, but they require each training sample to be presented in the form (x, y, e), where the environment label e is usually not available in many real-world datasets. Recent works such as Liu et al. (2021) try to relieve this requirement. To this end, we conduct another experiment on Waterbirds to show that ISRs can be used in cases where only a part of training samples are provided with environment labels. Adopting the same hyperparameter as that of Table 1, we reduce the available environment labels from 100% to 10% (randomly sampled), and apply ISR-Mean/Cov on top of ERM-trained models with the limited environment labels. We repeat the experiment over 10 runs for each of 10 ERM-trained models, and plot the mean accuracy in Fig. 6. We can observe that **a)** even with only 10% environment labels, the worst-group accuracy of ISR-Mean attains 73.4%, outperforming the original ERM-trained classifier by a large margin of 10.5%, and **b)** with 50% environment labels, the worst-group accuracy of ISR-Cov becomes 80.9%, surpassing the original classifier by 18.0%. The compelling results demonstrate another advantage of our ISRs, the *efficient utilization of environment labels*, which indicates that ISRs can be useful to many real-world datasets with only partial environment labels.

5.2.2 Applying ISRs to Pretrained Feature Extractors

It is recently observed that CLIP-pretrained models (Radford et al., 2021) have impressive OOD generalization ability across various scenarios (Miller et al., 2021; Wortsman et al., 2022; Kumar et al., 2022). Also, Kumar et al. (2022) shows that over a wide range of OOD benchmarks, linear probing (i.e., re-training the last linear layer only) could obtain better OOD generalization performance than finetuning all parameters for CLIP-pretrained models. Notice that ISR-Mean & ISR-Cov also re-train last linear layers on top of provided feature extractors, thus our ISRs can be used as substitutes for linear probing on CLIP-pretrained models. We empirically compare ISR-Mean/Cov vs. linear probing for a CLIP-pretrained vision transformer (ViT-B/32) in the Waterbirds dataset. As Table 2 shows, ISRs outperform linear probing in terms of both average and worst-group accuracy, and the improvement that ISR-Mean obtains is more significant than that of ISR-Cov. This experiment indicates that our ISRs could be useful post-processing tools for deep learning practitioners who frequently use modern pre-trained (foundation) models (Bommasani et al., 2021).

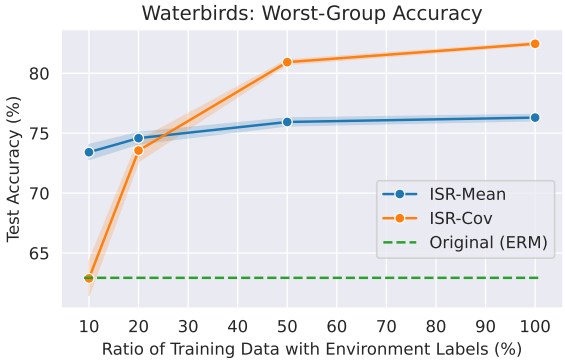

Figure 6. Applying ISR-Mean/ISR-Cov to ERM-trained models with partially available environment labels in the Waterbirds dataset. The shading area indicates the 95% confidence interval for mean accuracy.

6 Conclusion

In this paper, under a common data generative model in the literature, we propose two algorithms, ISR-Mean and ISR-Cov, to achieve domain generalization by recovering the invariant-feature subspace. We prove that ISR-Mean admits an $\mathcal{O}(d_s)$ environment complexity and ISR-Cov obtains an $\mathcal{O}(1)$ environment complexity, the minimum environment complexity that any algorithm can hope for. Furthermore, both algorithms are computationally efficient, free of local minima, and can be used off-the-shelf as a post-processing method over features learned from existing models. Empirically, we test our algorithms on synthetic benchmarks and demonstrate their superior performance when compared with other domain generalization algorithms. We also show that our proposed algorithms can be used as post-processing methods to increase the worst-case accuracy of (pre-)trained models by testing them on three real-world image and text datasets.

Acknowledgements

This work is partially supported by NSF grant No.1910100, NSF CNS No.2046726, C3 AI, and the Alfred P. Sloan Foundation. BL and HZ would like to thank the support from a Facebook research award.

References

Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.-C., Bengio, Y., Mitliagkas, I., and Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021a.

Ahuja, K., Wang, J., Dhurandhar, A., Shanmugam, K., and Varshney, K. R. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=jrA5GAccy_.

Albuquerque, I., Monteiro, J., Darvishi, M., Falk, T. H., and Mitliagkas, I. Generalizing to unseen domains via distribution matching, 2020.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Arora, R., Cotter, A., Livescu, K., and Srebro, N. Stochastic optimization for pca and pls. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 861–868. IEEE, 2012.

Aubin, B., Słowik, A., Arjovsky, M., Bottou, L., and Lopez-Paz, D. Linear unit-tests for invariance discovery. *arXiv preprint arXiv:2102.10867*, 2021.

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.

Ben-David, S., Blitzer, J., Crammer, K., Pereira, F., et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24: 2178–2186, 2011.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Caruana, R. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.

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*, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Du, X., Ramamoorthy, S., Duivesteijn, W., Tian, J., and Pechenizkiy, M. Beyond discriminant patterns: On the robustness of decision rule ensembles. *arXiv preprint arXiv:2109.10432*, 2021.

Eftekhari, A. and Hauser, R. A. Principal component analysis by optimization of symmetric functions has no spurious local optima. *SIAM Journal on Optimization*, 30(1): 439–463, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017.

Hauser, R. A., Eftekhari, A., and Matzinger, H. F. Pca by determinant optimisation has no spurious local optima. In *KDD*, pp. 1504–1511, 2018.

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.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. PMLR, 2018.

Javed, K., White, M., and Bengio, Y. Learning causal models online. *arXiv preprint arXiv:2006.07461*, 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.

Khezeli, K., Blaas, A., Soboczenski, F., Chia, N., and Kalantari, J. On invariance penalties for risk minimization. *arXiv preprint arXiv:2106.09777*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Koyama, M. and Yamaguchi, S. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint arXiv:2008.01883*, 2020.

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.

Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Li, B., Shen, Y., Wang, Y., Zhu, W., Reed, C. J., Zhang, J., Li, D., Keutzer, K., and Zhao, H. Invariant information bottleneck for domain generalization. *arXiv preprint arXiv:2106.06333*, 2021.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Marrinan, T., Ross Beveridge, J., Draper, B., Kirby, M., and Peterson, C. Finding the subspace mean or median to fit your need. In *CVPR*, pp. 1082–1089, 2014.

Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y., and Schmidt, L. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pp. 7721–7735. PMLR, 2021.

Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., and Schölkopf, B. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

Pearson, K. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference using invariant prediction: identification and confidence intervals, 2015.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Rosenfeld, E., Ravikumar, P. K., and Risteski, A. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=BbNIbVPJ-42.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Shi, C., Veitch, V., and Blei, D. Invariant representation learning for treatment effect estimation. *arXiv preprint arXiv:2011.12379*, 2020.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.

Vapnik, V. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pp. 831–838, 1992.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, pp. 5339–5349, 2018.

Vu, V. Q., Cho, J., Lei, J., and Rohe, K. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. *Advances in neural information processing systems*, 26, 2013.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, H., Zhao, H., and Li, B. Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. In *International Conference on Machine Learning*, pp. 10991–11002. PMLR, 2021a.

Wang, H., Wang, Y., Sun, R., and Li, B. Global convergence of maml and theory-inspired neural architecture search for few-shot learning. *CVPR*, 2022.

Wang, J., Lan, C., Liu, C., Ouyang, Y., Zeng, W., and Qin, T. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021b.

Wiles, O., Gowal, S., Stimberg, F., Alvise-Rebuffi, S., Ktena, I., Cemgil, T., et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., and Schmidt, L. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

Ye, H., Xie, C., Cai, T., Li, R., Li, Z., and Wang, L. Towards a theoretical framework of out-of-distribution generalization. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Ye, M., Jiang, R., Wang, H., Choudhary, D., Du, X., Bhushanam, B., Mokhtari, A., Kejariwal, A., and qiang liu. Future gradient descent for adapting the temporal shifting data distribution in online recommendation system. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

Zhang, G., Zhao, H., Yu, Y., and Poupart, P. Quantifying and improving transferability in domain generalization. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Zhao, H. On learning invariant representations for domain adaptation. *ICML*, 2019.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.

Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pp. 7523–7532. PMLR, 2019.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

A Proof

A.1 Proof of Theorem 1

Proof. From (11), we know

$$\mathcal{M} := \begin{bmatrix} \bar{x}_1^\mathsf{T} \\ \vdots \\ \bar{x}_E^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \mu_c^\mathsf{T} A^\mathsf{T} + \mu_1^\mathsf{T} B^\mathsf{T} \\ \vdots \\ \mu_c^\mathsf{T} A^\mathsf{T} + \mu_E^\mathsf{T} B^\mathsf{T} \end{bmatrix} = \overbrace{\begin{bmatrix} \mu_c^\mathsf{T} & \mu_1^\mathsf{T} \\ \vdots & \vdots \\ \mu_c^\mathsf{T} & \mu_E^\mathsf{T} \end{bmatrix}}^{\mathcal{U}^\mathsf{T} :=} R^\mathsf{T} = (R\mathcal{U})^\mathsf{T} \tag{18}$$

where $\mathcal{U} := \begin{bmatrix} \mu_c & \dots & \mu_c \\ \mu_1 & \dots & \mu_E \end{bmatrix} \in \mathbb{R}^{d \times E}$

If $E \leqslant d_s$, Assumption 1 guarantees that $\{\mu_1, \dots, \mu_E\}$ are linearly independent almost surely. Then, we have $\operatorname{rank}(\mathcal{U}) = E$. As $E > d_s$, since the first d_c rows of $\mathcal{U}$ are the same, the rank of $\mathcal{U}$ is capped, i.e., $\operatorname{rank}(\mathcal{U}) = d - d_c = d_s$.

The mean-subtraction step of PCA compute the sample-mean

$$\widetilde{x} = \frac{1}{E} \sum_{e=1}^{E} \bar{x}_e = A\mu_c + B \left(\frac{1}{E} \sum_{e=1}^{E} \mu_e \right) = A\mu_c + B\bar{\mu}, \tag{19}$$

where $\bar{\mu} := \frac{1}{E} \sum_{e=1}^{E} \mu_e$, and then subtracts $\widetilde{x}^\mathsf{T}$ off each row of $\mathcal{M}$ to obtain

$$\widetilde{\mathcal{M}} := \begin{bmatrix} \bar{x}_1^\mathsf{T} - \widetilde{x}^\mathsf{T} \\ \vdots \\ \bar{x}_E^\mathsf{T} - \widetilde{x}^\mathsf{T} \end{bmatrix} = \begin{bmatrix} (\mu_1 - \bar{\mu})^\mathsf{T} B^\mathsf{T} \\ \vdots \\ (\mu_E - \bar{\mu})^\mathsf{T} B^\mathsf{T} \end{bmatrix} = \overbrace{\begin{bmatrix} \mu_1^\mathsf{T} - \bar{\mu}^\mathsf{T} \\ \vdots \\ \mu_E^\mathsf{T} - \bar{\mu}^\mathsf{T} \end{bmatrix}}^{\widetilde{\mathcal{U}}^\mathsf{T} :=} B^\mathsf{T} = (B\widetilde{\mathcal{U}})^\mathsf{T} \in \mathbb{R}^{d_s \times d} \tag{20}$$

where $\widetilde{\mathcal{U}} := \begin{bmatrix} \mu_1 - \bar{\mu} & \dots & \mu_E - \bar{\mu} \end{bmatrix} \in \mathbb{R}^{d_s \times E}$

Similar to the analysis of $\mathcal{U}$ above, we can also analyze the rank of $\widetilde{\mathcal{U}}$ in the same way. However, different from $\mathcal{U}$, we have $\operatorname{rank}(\widetilde{\mathcal{U}}) = \min\{d_s, E-1\}$, where the -1 comes from the constraint $\sum_{e=1}^{E} (\mu_e - \bar{\mu}) = 0$ that is put by the mean-subtraction.

Suppose $E \geqslant d_s + 1$, then $\operatorname{rank}(\widetilde{\mathcal{U}}) = d_s$. The next step of PCA is to eigen-decompose the sample covariance matrix

$$\begin{aligned} \frac{1}{E} \widetilde{\mathcal{M}}^\mathsf{T} \widetilde{\mathcal{M}} = \frac{1}{E} (B\widetilde{\mathcal{U}})(B\widetilde{\mathcal{U}})^\mathsf{T} &= \frac{1}{E} B(\widetilde{\mathcal{U}}\widetilde{\mathcal{U}}^\mathsf{T}) B^\mathsf{T} \\ &= \frac{1}{E} \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \mathbf{0}_{d_c \times d_c} & \mathbf{0}_{d_c \times d_s} \\ \mathbf{0}_{d_s \times d_c} & \widetilde{\mathcal{U}}\widetilde{\mathcal{U}}^\mathsf{T} \end{bmatrix} \begin{bmatrix} A^\mathsf{T} \\ B^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{d \times d} \end{aligned} \tag{21}$$

where $\mathbf{0}_{n \times m}$ is a $n \times m$ matrix with all zero entries, and $\widetilde{\mathcal{U}}\widetilde{\mathcal{U}}^\mathsf{T} \in \mathbb{R}^{d_s \times d_s}$ is full-rank because $\operatorname{rank}(\widetilde{\mathcal{U}}) = d_s$.

Combining with the fact that $R = [AB]$ is full-rank (ensured by Assumption 3), we know that $\operatorname{rank}(\frac{1}{E} \widetilde{\mathcal{M}}^\mathsf{T} \widetilde{\mathcal{M}}) = d_s$. Therefore, $\frac{1}{E} \widetilde{\mathcal{M}}^\mathsf{T} \widetilde{\mathcal{M}}$ is positive-definite.

As a result, the eigen-decomposition on $\frac{1}{E} \widetilde{\mathcal{M}}^\mathsf{T} \widetilde{\mathcal{M}}$ leads to an eigen-spectrum of d_s positive values and $d_c = d - d_s$ zero eigenvalues.

Consider ascendingly ordered eigenvalues $\{\lambda_1, \dots, \lambda_d\}$, and compose a diagonal matrix S with these eigenvalues as in ascending order, i.e., $S := \operatorname{diag}(\{\lambda_1, \dots, \lambda_d\})$. Denote the eigenvectors corresponding with these eigenvalues as $\{P_1, \dots, P_{d_c}\}$, and stack their transposed matrices as

$$P := \begin{bmatrix} P_1^\mathsf{T} \\ \vdots \\ P_d^\mathsf{T} \end{bmatrix} \in \mathbb{R}^{d \times d} \tag{22}$$

Then, we have the equality

$$\frac{1}{E} B(\widetilde{\mathcal{U}}\widetilde{\mathcal{U}}^\mathsf{T}) B^\mathsf{T} = \frac{1}{E} \widetilde{\mathcal{M}}^\mathsf{T} \widetilde{\mathcal{M}} = PSP^\mathsf{T} \tag{23}$$

Since the first d_c diagonal entries of S are all zeros and the rest are all non-zero, the dimensions of P that correspond to non-zero diagonal entries of S can provide us with the subspace spanned by the d_s spurious latent feature dimensions, thus the rest dimensions of P (i.e., the ones with zero eigenvalues) correspond to the subspace spanned by d_c invariant latent feature dimensions, i.e.,

$$\mathrm{Span}(\{P_i^\mathsf{T} R : i \in [d],\ S_{ii} = 0\}) = \mathrm{Span}(\{\hat{\mathbf{1}}, \ldots, \hat{\mathbf{d}}_\mathbf{s}\}) \tag{24}$$

Since the diagonal entries of S are sorted in ascending order, we can equivalently write it as

$$\mathrm{Span}(\{P_1^\mathsf{T} R, \ldots, P_{d_c}^\mathsf{T} R\}) = \mathrm{Span}(\{\hat{\mathbf{1}}, \ldots, \hat{\mathbf{d}}_\mathbf{c}\}) \tag{25}$$

Then, by Proposition 1 (i.e., Definition 1 of Rosenfeld et al. (2021)) and Lemma F.2 of Rosenfeld et al. (2021), for the ERM predictor fitted to all data that are projected to the recovered subspace, we know it is guaranteed to be the optimal invariant predictor (defined in Proposition 1 as Eq. (6)). □

A.2 Proof of Theorem 2

Proof. From (15), we know

$$\Delta\Sigma := \Sigma_{e_1} - \Sigma_{e_2} = (\sigma_{e_1}^2 - \sigma_{e_2}^2) BB^\mathsf{T} \in \mathbb{R}^{d \times d} \tag{26}$$

Assumption 2 guarantees that $\sigma_{e_1}^2 - \sigma_{e_2}^2 \neq 0$, and Assumption 3 ensures that $\mathrm{rank}(B) = d_s$. Thus, eigen-decomposition on $\Delta\Sigma$ leads to exactly d_c zero eigenvalues and $d_s = 1 - d_c$ non-zero eigenvalues. One just need to follow the same steps as (21)-(25) to finish the proof. □

B Experimental Details

B.1 Setups of Synthetic Datasets

Example-2 This is a binary classification task that imitate the following example inspired by Arjovsky et al. (2019); Beery et al. (2018): while most cows appear in grasslands and most camels appear in desserts, with small probability such relationship can be flipped. In this example, Aubin et al. (2021) define the animals as invariant features with mean $\pm\mu_c$ and the backgrounds as spurious features with mean $\pm\mu_e$. Aubin et al. (2021) also scale the invariant and spurious features with ν_c and ν_e respectively. To be specific, we set $\mu_c = \mathbf{1}_{d_c}$ (i.e., a d_c-dimensional vector with all elements equal to 1) , $\mu_e = \mathbf{1}_{d_e}$, $\nu_c = 0.02$ and $\nu_e = 1$. For any training environment $e \in \mathcal{E}$, Aubin et al. (2021) construct its dataset $\mathcal{D}_e$ by generating each input-label pair (x, y) in the following process:

$$j_e \sim \text{Categorical}\left(p^e s^e, (1-p^e)s^e, p^e(1-s^e), (1-p^e)(1-s^e)\right)$$

$$z_c \sim \begin{cases} +1 \cdot (\mu_c + \mathcal{N}_{d_c}(0, 0.1)) \cdot \nu_c & \text{if } j_e \in \{1, 2\}, \\ -1 \cdot (\mu_c + \mathcal{N}_{d_c}(0, 0.1)) \cdot \nu_c & \text{if } j_e \in \{3, 4\}, \end{cases}$$

$$z_e \sim \begin{cases} +1 \cdot (\mu_e + \mathcal{N}_{d_s}(0, 0.1)) \cdot \nu_e & \text{if } j_e \in \{1, 4\}, \\ -1 \cdot (\mu_e + \mathcal{N}_{d_s}(0, 0.1)) \cdot \nu_e & \text{if } j_e \in \{2, 3\}, \end{cases}$$

$$z \leftarrow \begin{bmatrix} z_c \\ z_e \end{bmatrix}, \qquad y \leftarrow \begin{cases} 1 & \text{if } 1_{d_c}^\mathsf{T} z_c > 0, \\ 0 & \text{else} \end{cases}, \qquad x = Rz \quad \text{with} \quad R = I_d,$$

where the background probabilities are $p^{e=0} = 0.95$, $p^{e=1} = 0.97$, $p^{e=2} = 0.99$ and the animal probabilities are $s^{e=0} = 0.3$, $s^{e=1} = 0.5$, $s^{e=2} = 0.7$. If there are more than three environments, the extra environment variables are drawn according to $p^e \sim \mathrm{Unif}(0.9, 1)$ and $s^e \sim \mathrm{Unif}(0.3, 0.7)$.

Example-3 This is a linear version of the spiral binary classification problem proposed by Parascandolo et al. (2020). In this example, Aubin et al. (2021) assign the first d_c dimensions of the features with an invariant, small-margin linear decision boundary, and the reset d_e dimensions have a changing, large-margin linear decision boundary. To be specific, for all environments, the d_c invariant features are sampled from a distribution with a constant mean, while the means are sampled from a Gaussian distribution for the d_e spurious features. In practice set $\gamma = 0.1 \cdot \mathbf{1}_{d_c}$, $\mu_e \sim \mathcal{N}(\mathbf{0}_{d_c}, I_{d_c})$, and $\sigma_c = \sigma_e = 0.1$, for all environments. For any training environment $e \in \mathcal{E}$, Aubin et al. (2021) construct its dataset $\mathcal{D}_e$ by

generating each input-label pair (x, y) in the following process:

$$
\begin{aligned}
y &\sim \text{Bernoulli}\left(\frac{1}{2}\right), \\
z_c &\sim \begin{cases} \mathcal{N}(+\gamma, \sigma_c I_{d_c}) & \text{if } y = 0, \\ \mathcal{N}(-\gamma, \sigma_c I_{d_c}) & \text{if } y = 1; \end{cases} \\
z_e &\sim \begin{cases} \mathcal{N}(+\mu_e, \sigma_e I_{d_s}) & \text{if } y = 0, \\ \mathcal{N}(-\mu_e, \sigma_e I_{d_s}) & \text{if } y = 1; \end{cases} \\
z &\leftarrow \begin{bmatrix} z_c \\ z_e \end{bmatrix}, \qquad x = Rz \quad \text{with} \quad R = I_d,
\end{aligned}
$$

Example-3' As explained in Section 5.1, in order to make Example-3 follow Assumption 2, we slightly modify the variance of the features in Example-3 so that $\sigma_c = 0.1$ and $\sigma_e \sim \text{Unif}(0.1, 0.3)$. All the rest settings are unchanged.

Example-2s/3s/3s' In order to increase the difficulty of the tasks, we defined the "scrambled" variations of the three problems described above. To build the scrambled variations, we no longer use the identity matrix I_d as the transformation matrix R; instead, a random orthonormal matrix $R \in \mathbb{R}^{d \times d}$ is applied to the features for all environments $e \in \mathcal{E}$. The random transformation matrix is built from a Gaussian matrix (see the code `https://github.com/facebookresearch/InvarianceUnitTests` of Aubin et al. (2021) for details).

B.2 Experiments on Synthetic Datasets

Code We adopt the codebase of Linear Unit-Tests (Aubin et al., 2021), which provide implementations of Example-2/2s/3/3s and multiple algorithms (including IRMv1, IGA, ERM, Oracle). This codebase is released at `https://github.com/facebookresearch/InvarianceUnitTests`.

Hyper-parameters Similar to Aubin et al. (2021), we perform a hyper-parameter search of 20 trials. For each trial, we train the algorithms on the training split of all environments for 10K full-batch Adam (Kingma & Ba, 2015) iterations. We run the search for ISR-Mean and ISR-Cov algorithms on all examples, and run the search for ERM, IGA (Koyama & Yamaguchi, 2020), IRMv1 (Arjovsky et al., 2019) and Oracle on Example-3' and Example-3s'. We choose the hyper-parameters that minimize the mean error over the validation split of all environments. The experiment results for ERM, IGA, IRMv1 and Oracle on Example-2, Example-2s, Example-3 and Example-3s are from (Aubin et al., 2021), thus we do not perform any search on them.

B.3 Experiments on Real Datasets

Training We directly use models, hyper-parameters and running scripts provided by authors of Sagawa et al. (2019) in `https://github.com/kohpangwei/group_DRO`. Specifically, they use ResNets (He et al., 2016) for Waterbirds and CelebA, and deploy BERT (Devlin et al., 2019) for MultiNLI. We train the neural nets following the official running scripts provided in `https://worksheets.codalab.org/worksheets/0x621811fe446b49bb818293bae2ef88c0` over 10 random seeds for Waterbirds/CelebA/MultiNLI. Each run leads to one trained neural network selected on the epoch with the highest worst-group validation accuracy.

ISR-Mean There are only $E = 2$ environments for Waterbirds, CelebA and MultiNLI and ISR-Mean can only identify a $\min\{E - 1, d_s\}$-dimensional spurious subspace. Thus we assume $d_s = 1$ for the three datasets when applying ISR-Mean.

ISR-Cov For real datasets, we do not know the d_s of the learned features, thus we have to treat d_s as a hyperparameter for Algorithm 2.

Numerical Techniques The feature space of learned models is usually of a high dimension (e.g., 2048 for ResNet-50 in Waterbirds/CelebA), while the features of training data usually live in a subspace (approximately). Thus, we typically apply dimension reduction to features through a PCA. Then, to overcome some numerical instability challenges, we modify Algorithm 1 & 2 slightly: Instead of directly identifying the invariant-feature subspace as Algorithm 1 & 2 suggest, we apply ISR-Mean/Cov in an equivalent approach: we first identify the spurious-feature subspace, and then reduces scales of features along the spurious-feature subspace. The final step of fitting linear predictors in Algorithm

1/2 is done by logistic regression solver provided in scikit-learn Pedregosa et al. (2011). But in some cases, we find that directly adapting the original predictor of the trained model also yields good performance. See more details in https://github.com/Haoxiang-Wang/ISR.