The Missing Invariance Principle Found – the Reciprocal Twin of Invariant Risk Minimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Machine learning models often generalize poorly to out-of-distribution (OOD) data as a result of relying on features that are spuriously correlated with the label during training. Recently, the technique of Invariant Risk Minimization (IRM) was proposed to learn predictors that only use invariant features by conserving the feature-conditioned label expectation $\mathbb{E}_e[y|f(x)]$ across environments. However, more recent studies have demonstrated that IRM-v1, a practical version of IRM, can fail in various task settings. Here, we identify a fundamental flaw of IRM formulation that causes the failure. We then introduce a complementary notion of invariance, MRI, based on conserving the label-conditioned feature expectation $\mathbb{E}_e[f(x)|y]$ across environments, which is free of this flaw. Further, we introduce a simplified, practical version of the MRI formulation called MRI-v1. We note that this constraint is convex which confers it with an advantage over IRM-v1, which imposes non-convex constraints. We prove that in a general linear problem setting, MRI-v1 can guarantee invariant predictors given sufficient environments. We also empirically demonstrate that MRI strongly out-performs IRM and consistently achieves near-optimal OOD generalization in image-based nonlinear problems.

1 Introduction

Deep learning models have shown tremendous success over the past decade. These models show great generalization properties when tested on the same distribution as the training dataset (in-distribution generalization). However, these models often show catastrophic failure when tested on out-of-distribution dataset, revealing that they learned to use easier-to-fit features that are spuriously correlated to the label in the given training domains but do not generalize to the testing domains. For example, deep networks trained on pictures of cow with only grassy backgrounds in the training domain will use the background color as the predictive feature which is easier to learn and generalize poorly to pictures of cow, for example, with a dessert background.

Recently, there has been a growing interest in developing models that generalize well across multiple domains. In particular, there has been a recent body of works that focus on developing algorithms that attempt to learn invariant predictors (Arjovsky et al., 2019; Ahuja et al., 2021; Peters et al., 2016; Rojas-Carulla et al., 2018; Heinze-Deml et al., 2018). Invariant Risk Minimization (IRM) and its practical version, IRM-v1, in particular, have attracted much attention as one of the first methods that are compatible with deep learning techniques. However, several follow-up studies have empirically demonstrated that IRM-v1 is unreliable at learning invariant representations (Kamath et al., 2021; Rosenfeld et al., 2020; Gulrajani and Lopez-Paz, 2020; Ahuja et al., 2020, 2021). Here, we identify a fundamental flaw of IRM formulation that causes this limitation and propose a new method that is free of this flaw.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Related Works There has been considerable work in the field of learning invariant representations. They vary from learning domain-invariant feature representations conserving $P(f(x))$ using kernel methods (Muandet et al., 2013; Ghifary et al., 2016; Hu et al., 2020), variational autoencoder (Ilse et al., 2020), and adversarial networks Ganin et al. (2016); Long et al. (2018); Akuzawa et al. (2019); Albuquerque et al. (2019) to learning invariant class-conditional features $P(f(x)|y)$ (Gong et al., 2016; Li et al., 2018b) in the context of domain adaptation, which assumes access to the test distribution for adaptation. There is also a large body of work to learn invariant representations in the field of domain generalization that doesn't assume access to test distributions. This includes imposing invariance of $\mathbb{E}_e[y|f(x)]$ (Arjovsky et al., 2019) with information bottleneck constraint (Ahuja et al. (2021)), imposing object-invariant condition (Mahajan et al. (2021)), using domain inference (Creager et al., 2021), model calibration (Wald et al., 2021), and others (Krueger et al., 2021; Li et al., 2018a; Shankar et al., 2018).

Our Contributions We introduce a simpler formulation of IRM that is more generalizable, and show that it can be modified to yield a new complementary notion of invariance, called MRI. We also show that IRM has a fundamental flaw due to the indirect way it imposes invariance which leads to the failure of IRM-v1. MRI is shown to be free of this flaw. We prove that in a general linear problem setting, MRI-v1 can guarantee invariant predictors given sufficient environments. We then empirically demonstrate that MRI strongly out-performs IRM and consistently achieves near-optimal OOD generalization in image-based nonlinear problems.[1]

2 Problem Formulation

Consider a set of environments $\mathcal{E} = \{e\}$, each of which defines a distribution $P_e(x, y)$ over inputs and labels, from which the dataset of the environment is drawn $\mathcal{D}_e \equiv \{(x_j^e \in \mathbb{R}^d, y_j^e \in \mathbb{R})\}$. The distribution $P_e(x, y)$ is assumed to be generated according to the causal graph in Fig 1 as introduced in Rosenfeld et al. (2020), which includes latent features that are invariantly z_i or spuriously z_s correlated with the label. The observation x is generated as a function of the latent features: $x = g(z_i, z_s)$ is assumed to be injective, which makes it possible to recover the latent features from the observation.

Note that while this work focuses on the causal graph in Figure 1, our results are also applicable to the graph in which the invariant feature z_i causes the label y ($Z_i \to Y$). This is the graph considered by Arjovsky et al. (2019) in Figure 3 with the difference being that they allow both the invariant feature and label distribution to also vary across environments. Also, while Arjovsky et al. (2019) focused on the causal graph with $Z_i \to Y$, the IRM method is also applicable to the causal graph considered in Figure 1 (Rosenfeld et al., 2020).

The risk of a predictor $f : X \to O$ in environment e is defined as the population average

$$\begin{aligned}\mathcal{L}_e(f) &\equiv \mathbb{E}_{P_e(x,y)}[l(f(x), y)] && (1)\\ &= \mathbb{E}_{P_e(o)}[\mathbb{E}_{P_e(y|o)}[l(o, y)]] && (2)\\ &= \mathbb{E}_{P(y)}[\mathbb{E}_{P_e(o|y)}[l(o, y)]] && (3)\end{aligned}$$

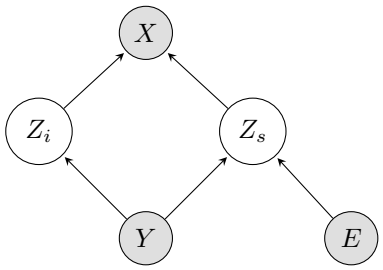


Figure 1: Causal graph depicting the data generating process. Shading indicates the variable is observed.

where $o = f(x)$ is the predictor's output. Here, we consider standard convex loss functions $l : O \times Y \to \mathbb{R}_{\geq 0}$, including the square loss $l_{\text{sq}}(o, y) = \frac{1}{2}(o - y)^2$ for regression ($O, Y \subseteq \mathbb{R}$) and the binary-cross-entropy (BCE) loss $l_{\log}(o, y) = -(1 + y)\log(\eta(o)) - (1 - y)\log(1 - \eta(o))$ for binary classification ($O \subseteq \mathbb{R}$, $Y \subseteq [-1, 1]$), where $\eta(o) \equiv 1/(1 + e^{-o})$ is the sigmoid function.

[1] Code available at https://anonymous.4open.science/r/MRI-2608

3 IRM vs MRI Invariance

3.1 IRM Paradigm

3.1.1 Original formulation

Definition 3.1. (Arjovsky et al., 2019) Given a predictor $\psi \circ f$, the feature representation $f : X \to Z$ is *IRM invariant* over a set of environments $\mathcal{E}$ if there exists $\psi : Z \to O$ such that ψ is simultaneously optimal on f for all environments in $\mathcal{E}$, *i.e.*

$$\forall e \in \mathcal{E}, \qquad \psi \in \arg\min_{\bar{\psi}} \mathcal{L}_e(\bar{\psi} \circ f) \tag{4}$$

where ψ is assumed to be unrestricted in the space of all measurable functions.

Lemma 3.2. *(Kamath et al., 2021) For standard loss functions*[2]*, Definition 3.1 is equivalent to*

$$\exists \psi,\ \forall e \in \mathcal{E}, \qquad \mathbb{E}_e[y|f(x)] = \sigma(\psi \circ f(x)). \tag{5}$$

where $\mathbb{E}_e[y|f(x)] \equiv \mathbb{E}_{P_e(y|f(x))}[y]$ is the feature-conditioned label expectation, and σ is a monotonic function that depends on the loss function.

3.1.2 Variational formulation

The original formulation above is overly complex due to the composite predictor $\psi \circ f$ and the optimality condition on ψ. For further analysis, we introduce a simplified formulation of IRM defined by the following variational principle.

Definition 3.3. A predictor $f : X \to O$ is *IRM invariant* over a set of environments $\mathcal{E}$, if the risk $\mathcal{L}_e(f)$ remains stationary under arbitrary infinitesimal perturbations on the predictor's output $o = f(x)$ for all environments in $\mathcal{E}$, *i.e.*

$$\begin{aligned} \forall e \in \mathcal{E}, \quad \delta\mathcal{L}_e(f) &= \lim_{\epsilon \to 0} \mathbb{E}_{P_e(o,y)}[l(o + \epsilon\delta\psi(o), y) - l(o, y)]/\epsilon \\ &= \mathbb{E}_{P_e(o)}[\mathbb{E}_{P_e(y|o)}[\partial_o l(o, y)] \cdot \delta\psi(o)] = 0 \end{aligned} \tag{6}$$

where $\delta\psi : O \to O$ is an arbitrary perturbation that is unrestricted in the space of all measurable functions, and $\delta\mathcal{L}_e(f)$ denotes the resulting change in risk.

Lemma 3.4. *For standard loss functions*[2]*, Definition 3.3 is equivalent to*

$$\forall e \in \mathcal{E}, \qquad \mathbb{E}_e[y|o] = \sigma(o), \tag{7}$$

which is equivalent to Lemma 3.2 with f replacing the composite predictor $\psi \circ f$.

Proof. Eq (6) is satisfied if and only if $\mathbb{E}_{P_e(y|o)}[\partial_o l(o, y)] = 0$. For standard loss functions, the loss derivative has the form $\partial_o l(o, y) = -y + \sigma(o)$, which yields $\mathbb{E}_e[\partial_o l(o, y)|o] = -\mathbb{E}_e[y|o] + \sigma(o) = 0$. □

3.1.3 Conservation law of IRM

It was noted in Arjovsky et al. (2019); Kamath et al. (2021) that the essense of IRM's invariance is the conservation of the *feature-conditioned label expectation*, i.e.

$$\forall e_1, e_2 \in \mathcal{E}, \qquad \mathbb{E}_{e_1}[y|f(x)] = \mathbb{E}_{e_2}[y|f(x)]. \tag{8}$$

IRM eq (7) (also eq (5)) indeed implies the conservation law, since its RHS is constant with respect to the environment index e. However, there is a critical difference: Eq (8) only requires $\mathbb{E}_e[y|o]$ to share the same value across environments, thus imposing $|\mathcal{E}| - 1$ constraints, whereas eq (7) (also eq (5)) additionally requires that value to be also equal to the RHS term $\sigma(o)$. This extra constraint is due to the indirect way IRM attains the conservation law, *i.e.* by equaling $\mathbb{E}_e[y|o]$ to an intermediary term $\sigma(o)$, which imposes 1 constraint per environment.

[2] Square loss and BCE loss are considered: $\sigma(o) = o$ for square loss, and $\sigma(o) = \tanh(o/2)$ for BCE loss.

3.1.4 IRM-v1

Due to the impracticality of considering the unrestricted function space of ψ, Arjovsky et al. (2019) suggested restricting ψ to the space of linear functions. In the variational formulation, this corresponds to restricting the output perturbations $\delta\psi$ to the space of linear functions, which, for scalar outputs, is equivalent to the identity function, $\delta\psi(o) = o$. This reduces eq (6) to

$$\forall e \in \mathcal{E}, \quad \delta\mathcal{L}_e(f) = \mathbb{E}_e[\,\partial_o l(o, y) \cdot o\,] = 0. \tag{9}$$

The reduced constraints in eq (9) are identical to IRM-s in Kamath et al. (2021), which imposes $\forall e \in \mathcal{E}, \partial_\psi \mathbb{E}_e[\,l(\psi \cdot o, y)] = 0$, where ψ is simply a scalar multiplicative factor[3]. In Arjovsky et al. (2019), this reduced formulation is termed IRM-v1 when the constraints are imposed in a *soft manner*, *i.e.* as squared penalty terms (See eq (16)). In the literature, the term IRM-v1 is widly used for the reduced formulation regardless of whether hard or soft constraints are used, which we adopt here.

Despite the practical applicability, it has been empirically shown that IRM-v1 can behave quite differently from IRM (Kamath et al., 2021). This failure mode can be analytically understood here. Since $\mathbb{E}_e[\,\partial_o l(o, y) \cdot o\,] = \mathbb{E}_e[o(\sigma(o) - y)]$ for standard loss functions[2], eq (9) is equivalent to

$$\forall e \in \mathcal{E}, \quad \mathbb{E}_e[oy] = \mathbb{E}_e[o\sigma(o)]. \tag{10}$$

Unlike eq (7), however, RHS of eq (10) does not necessarily remain constant across environments, and hence does not prescribe a meaningful invariance. Therefore, restricting the function space of $\delta\psi$ (or ψ) invalidates the indirect way of imposing invariance in IRM involving an intermediary term.

3.2 MRI Paradigm

We now introduce a complementary notion of invariance by considering infinitesimal perturbations on label, which we call *Mirror Reflected IRM*, or MRI.

Definition 3.5. A predictor $f : X \to O$ is MRI invariant over a set of environments $\mathcal{E}$, if the change in risk due to arbitrary infinitesimal label perturbations is *shared* across environments, *i.e.*

$$\forall e_1, e_2 \in \mathcal{E}, \quad \delta\mathcal{L}_{e_1}(f) = \delta\mathcal{L}_{e_2}(f) \tag{11}$$

$$\text{where} \quad \delta\mathcal{L}_e(f) \equiv \lim_{\epsilon \to 0} \mathbb{E}_{P_e(x,y)}[\,l(o, y + \epsilon\delta\psi(y)) - l(o, y)\,]/\epsilon$$

$$= \mathbb{E}_{P(y)}[\,\mathbb{E}_e[\,\partial_y l(o, y)|y\,] \cdot \delta\psi(y)\,] \tag{12}$$

where $\delta\psi : Y \to Y$ is an arbitrary perturbation that is unrestricted in the space of all measurable functions, and $\delta\mathcal{L}_e(f)$ denotes the resulting change in risk. $\mathbb{E}_e[\partial_y l(o, y)|y] \equiv \mathbb{E}_{P_e(o|y)}[\partial_y l(o, y)]$.

Lemma 3.6. *For standard loss functions, Definition 3.5 is equivalent to*

$$\forall e_1, e_2 \in \mathcal{E}, \qquad \mathbb{E}_{e_1}[o|y] = \mathbb{E}_{e_2}[o|y], \tag{13}$$

which conserves the label-conditioned feature expectation $\mathbb{E}_e[o|y] \equiv \mathbb{E}_{P_e(o|y)}[o]$ *across environments.*

Proof. Eq (11) is satisfied if and only if $\mathbb{E}_e[\partial_y l(o, y)|y]$ is conserved. For standard loss functions, the loss derivative has the form $\partial_y l(o, y) = -o + \rho(y)$[4]. Therefore, $\mathbb{E}_{e_1}[\partial_y l(o, y)|y] - \mathbb{E}_{e_2}[\partial_y l(o, y)|y] = \mathbb{E}_{e_1}[o|y] - \mathbb{E}_{e_2}[o|y] = 0$. □

Note that MRI eq (11) imposes the correct number of constraints as needed by the conservation law eq (13), as it attains invariance in a direct manner without involving any intermediary terms.

3.2.1 MRI-v1

Restricting the label perturbations to the space of linear functions, or equivalently, an identity function $\delta\psi(y) = y$, reduces eq (12) to $\delta\mathcal{L}_e(f) = \mathbb{E}_e[\,\partial_y l(o, y) \cdot y\,]$. This reduces the conservation law eq (13) to

$$\forall e_1, e_2 \in \mathcal{E}, \qquad \mathbb{E}_{e_1}[oy] = \mathbb{E}_{e_2}[oy]. \tag{14}$$

which is in fact a necessary condition for the MRI invariance eq (13). Therefore, the direct way of MRI's constraints imposing invariance continues to hold even when $\delta\psi$ is restricted to the space of linear functions.

[3] Note that $\partial_\psi l(\psi \cdot o, y) = \partial_o l(o, y) \cdot o$.

[4] Square loss and BCE loss are considered: $\rho(y) = y$ for square loss, and $\rho(y) = 0$ for BCE loss.

Algorithm	Constraint $\vec{c}(f)$	$\delta\mathcal{L}_e(f)$
IRM-v1	$\delta\vec{\mathcal{L}}_{\text{tr}}(f)$	$\mathbb{E}_e[\,\partial_o l(o,y)\cdot o\,]$
MRI-v1	$Q\,\delta\vec{\mathcal{L}}_{\text{tr}}(f)$	$\mathbb{E}_e[\,\partial_y l(o,y)\cdot y\,]$

Table 1: Invariance constraints of IRM-v1 and MRI-v1. $o = f(x)$.

4 Training methods

4.1 Constrained optimization problem

The full methods of IRM-v1 and MRI-v1 can be formalized as a constrained optimization problem

$$\min_{\theta} \mathcal{L}_{\text{tr}}(f(\theta)) \qquad \text{subject to } \vec{c}(f(\theta)) = \vec{0}, \tag{15}$$

where $\mathcal{L}_{\text{tr}} = \frac{1}{|\mathcal{E}_{\text{tr}}|}\sum_{e\in\mathcal{E}_{\text{tr}}}\mathcal{L}_e$ is the average risk over the set of training environments $\mathcal{E}_{\text{tr}} \subset \mathcal{E}$, and θ is the parameters of the predictor f. The constraint functions are $\vec{c} = \delta\vec{\mathcal{L}}_{\text{tr}} \in \mathbb{R}^{|\mathcal{E}_{\text{tr}}|}$ for IRM-v1 and $\vec{c} = Q\,\delta\vec{\mathcal{L}}_{\text{tr}} \in \mathbb{R}^{|\mathcal{E}_{\text{tr}}|-1}$ for MRI-v1, where $\delta\vec{\mathcal{L}}_{\text{tr}} \in \mathbb{R}^{|\mathcal{E}_{\text{tr}}|}$ is a vector of perturbed risks $\delta\mathcal{L}_e$ for $e \in \mathcal{E}_{\text{tr}}$, and $Q \in \mathbb{R}^{(|\mathcal{E}_{\text{tr}}|-1)\times|\mathcal{E}_{\text{tr}}|}$ is an orthonormal matrix that satisfies $Q\,\vec{1} = \vec{0}$. For example, in the case of a training set of two environments $\mathcal{E}_{\text{tr}} = \{e_1, e_2\}$, $Q\,\delta\vec{\mathcal{L}}_{\text{tr}}$ is simply the difference of perturbed risks $(\delta\mathcal{L}_{e_1} - \delta\mathcal{L}_{e_2})/\sqrt{2}$, since $Q = [1, -1]/\sqrt{2}$. The constraints are summarized in Table 1.

4.2 Soft-constraint methods

Numerically solving IRM-v1 and MRI-v1 requires converting the hard constraints $\vec{c}(f(\theta)) = \vec{0}$ to soft constraints, which allows the use of off-the-shelf gradient-based optimization algorithms.

Penalty Method (PM) Penalty method is the most commonly used approach, including in Arjovsky et al. (2019), which adds the squared residual constraints as a penalty term to the objective,

$$\min_{\theta}\ \mathcal{L}_{\text{tr}}(f(\theta)) + \mu\,\|\vec{c}(f(\theta))\|^2 \tag{16}$$

However, this method requires increasing $\mu^t \to \infty$ over training iterations t in order to approximate the exact hard-constraint, which leads to training instability and slow convergence (Bertsekas, 1976).

Augmented Lagrangian Method (ALM) ALM was introduced to overcome the limitations of penalty method (Bertsekas, 1976), which adds a Lagrange multiplier term to (16)

$$\min_{\theta}\ \mathcal{L}_{\text{tr}}(f(\theta)) + \mu\,\|\vec{c}(f(\theta))\|^2 + \vec{\lambda}^{\intercal}\cdot\vec{c}(f(\theta)), \tag{17}$$

where $\vec{\lambda}$ is typically initialized at $\vec{0}$ and gets updated as $\vec{\lambda}^{t+1} = \vec{\lambda}^t + \mu\vec{c}(f(\theta^t))$ to accumulate the residual constraints at each training iteration t. In practice, ALM can operate with moderate values of μ (~ 10) without fine tuning, and thus exhibits fast and stable convergence (Bertsekas, 1976).

5 Analytic Results

5.1 General Linear SEM

In this section, we show that MRI-v1 can correctly remove all spurious features given a sufficient number of environments.

We consider a data generating process according to Figure 1, in which $x = g(z_i, z_s)$ is a linear function. Note that this Structural Equation Model (SEM, Pearl (2009)) is a generalization of the linear Gaussian model considered in Rosenfeld et al. (2020), which additionally assumed binary y and a specific noise model (*i.e.* linear additive Gaussian) for the latent features. In comparison, we don't put any assumptions on the noise generating process or the label.

We consider a linear predictor $f : X \to O$. Since g is injective and has an inverse over its range, without loss of generality, we can define f as a linear function directly over the latents as

$$o = f(x; \theta) = w_i \cdot z_i + w_s \cdot z_s \tag{18}$$

with $\theta = \{w_i \in \mathbb{R}^{d_i}, w_s \in \mathbb{R}^{d_s}\}$, where d_i is the dimension of invariant features, d_s is the dimension of spurious features.

Theorem 5.1. *Given $|\mathcal{E}_{tr}| > d_s$ training environments, and that $\mathbb{E}_e[z_s y]$ are in general linear position, MRI-v1 will eliminate all spurious feature dimensions.*

Proof. MRI-v1 constraint eq (14) is

$$\forall e \in \mathcal{E}_{\text{tr}}, \quad \mathbb{E}_e[oy] = w_i \cdot \mathbb{E}_e[z_i y] + w_s \cdot \mathbb{E}_e[z_s y] = const$$

This can be expressed in a matrix form as

$$w_s \cdot M' = 0, \tag{19}$$

where $M' = M - \bar{M} \cdot 1$ with $M \equiv [\mathbb{E}_e[z_s y]]_{e \in \mathcal{E}_{\text{tr}}} \in \mathbb{R}^{d_s \times |\mathcal{E}_{\text{tr}}|}$ and $\bar{M} = \frac{1}{|\mathcal{E}_{\text{tr}}|} \sum_{e \in \mathcal{E}_{\text{tr}}} \mathbb{E}_e[z_s y]$. Since $rank(M') = d_s$, eq (19) is equivalent to $w_s = 0$.

□

Rosenfeld et al. (2020) derives a similar result that IRM-v1 can eliminate all spurious feature dimensions given $|\mathcal{E}_{\text{tr}}| > d_s$. However, unlike our results for MRI-v1, their result is restricted to a specialized linear family of environments with binary labels and additive gaussian noise. In fact, IRM-v1 has been shown to fail to eliminate spurious features in a more general linear problems in which the noise is not additive Gaussian (Arjovsky et al., 2019; Kamath et al., 2021).

5.2 Minimal Example: $d_i = 1,\ d_s = 1,\ |\mathcal{E}_{\text{tr}}| = 2$

Here, we demonstrate the result for the general linear SEM above with a minimal linear regression problem that involves one invariant feature, one spurious feature and two training environments. This toy example shares the same structure as the Shape-texture dataset (Section 6.1). It is analogous to the two-bit environment task from Kamath et al. (2021). The detailed experimental setting and the analytical solutions are presented in the Supplementary Materials.

IRM-v1 IRM-v1 has two quadratic equality constraints ($\delta\mathcal{L}_{e_1}(f) = 0, \delta\mathcal{L}_{e_2}(f) = 0$), shown as two elliptic curves. The intersection between the non-convex constraints on a 2-D feature space yields a disjoint set of 0-D points, all of which are local constrained optima, including the true invariant optimum, a zero-predictor solution, and two non-invariant solutions. Note that one of the non-invariant solutions exhibits lower training loss than the true invariant optimum solution.

IRM-relaxed We also consider a relaxed version of IRM-v1, whose constraint function is $c = Q\,\delta\vec{\mathcal{L}} = (\delta\mathcal{L}_{e_1} - \delta\mathcal{L}_{e_2})/\sqrt{2}$ with $\delta\mathcal{L}_e = \mathbb{E}_e[\partial_o l(o, y) \cdot o]$, such that it has a single constriant $\delta\mathcal{L}_{e_1}(f) = \delta\mathcal{L}_{e_2}(f)$. This constraint is a set of hyperbolic curves, which appears to be two straight lines in Figure 2a. Due to the non-convex constraint, this problem exhibits two local constrained optima, including the true invariant optimum and a non-invariant solution, with the non-invariant solution exhibiting lower training loss. Therefore, simply reducing the number of constraints of IRM-v1 does not resolve its limitation.

MRI-v1 MRI-v1 has a single linear equality constraint, which is exactly the set of invariant solutions ($w_s = 0$): That is, any point on the constraint is a invariant predictor. This is a convex problem, since both the objective and the constraint are convex, and thus features a unique optimum, which is the true invariant optimum solution.

In Figure 2b,c, we show the learning trajectory under full-batch gradient descent (penalty method eq (16) with $\mu = 5 \times 10^4$ used). The convergence pattern for IRM-v1 is highly dependent on the initialization (shown with different trajectory colors), due to the presence of multiple local minima. In this case, the invariant optimum is especially extremely difficult to converge to as it has higher training loss than the nearby non-invariant minima. In contrast, MRI-v1 always converges to the invariant

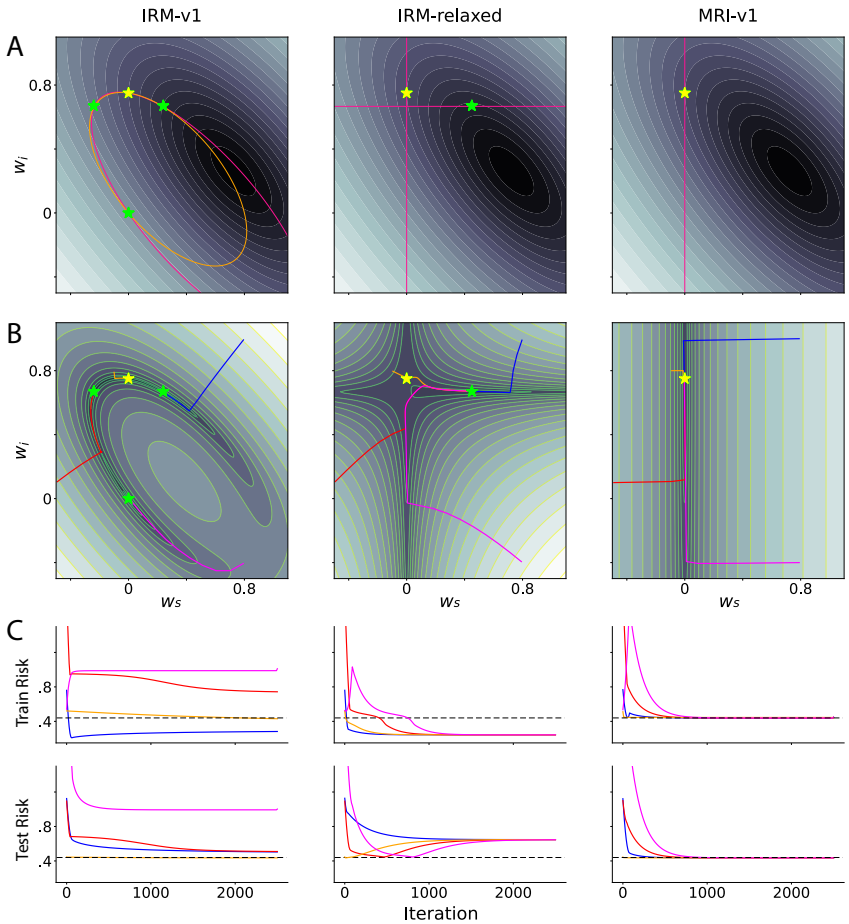


Figure 2: Minimal example in Section 5.2. (A) Hard-constraint case. Background: Average train risk $\mathcal{L}_{\text{tr}}$. Lines: Invariance constraints $\vec{c}$. Stars: Local constrained optima. (B) Soft-constraint case. Background: $\mathcal{L}_{\text{tr}} + \mu \|\vec{c}\|^2$. Overlaid are examplary learning trajectories of different intializations. (C) Loss profiles of examplary learning trajectories. (Top) Average train risk $\mathcal{L}_{\text{tr}}$. (Bottom) Test domain risk $\mathcal{L}_{\text{test}}$. Trajectories are color-matched with panel B. Dotted line: Optimal invariant solution.

optimum regardless of initialization since it is a unique minimum. For IRM-relaxed, the invariant optimum is lost at this value of μ, which exists in the limit $\mu \to \infty$. As a result, all trajectories converge to the non-invariant optimum.

Minimal examples for binary classification, based on the shape-texture classification and the toy-CMNISTa/b datasets, are also analyzed in Supplementary Materials (Figure 3).

6 Nonlinear Image-based Problems

6.1 Datasets

Shape-Texture Dataset We develop a new versatile dataset designed for domain generalization, called the Shape-Texture dataset, based on the data generating model in Figure 1. This dataset is designed to test algorithms on linear and non-linear regression and classification problems in multiple adjustable settings.

The dataset involves an invariant latent feature $\theta_i \in \mathbb{R}$, a spurious latent feature $\theta_s \in \mathbb{R}$, and the latent feature $\theta_y \in \mathbb{R}$ for the label. Each of them represents an angle of orientation, and therefore is naturally represented on a unit circle in the complex plane as $e^{i\theta} \in \mathcal{S}^1$. In the linear regression setting, the observed input is simply the concatenated features $x = [e^{i\theta_i}, e^{i\theta_s}]$, and the label is $y = e^{i\theta_y}$. Whereas in the linear classification setting, the input is given by $x = [\sin(\theta_i), \sin(\theta_s)]$, and the label

is $y = H(\sin(\theta_y))$ where H denotes the Heaviside step function. In the non-linear regression and classification setting, the label remains the same as their respective linear versions but the input is an image (Figure 4) composed of two planar waves of differing frequencies; The orientation of the low frequency wave (a.k.a shape) is given by $\theta_i/2$, and the orientation of the high frequency wave (a.k.a texture) is given by $\theta_s/2$. For all task settings, the latent angles and the label angle are linearly related as $\theta_i \leftarrow \theta_y + \xi_i, \theta_s \leftarrow \theta_y + \xi_s$, where $\xi_* \sim \mathcal{N}(0, \sigma_*)$ with probability p_*, or $\xi_* \sim \mathcal{U}(-\pi, \pi)$ probability $1 - p_*$ for $* \in \{i, s\}$. p_i, σ_i are fixed invariant parameters whereas p_s, σ_s can vary from one environment to another. In this work, for all task settings, we use two training environments and one testing environment with $p_i = 0.75$ for all environments, $p_s = 0.8, 1$ for the two training environments, $p_s = 0$ for the testing environment, and $\sigma_i = \sigma_s = 0$ for all environments.

Colored MNIST The Colored MNIST (CMNIST) dataset was introduced by Arjovsky et al. (2019) to analyze IRM-v1's ability to learn non-linear invariant predictors. This is a synthetic binary classification task derived from MNIST. In this dataset, the color of the image is determined in correlation with the label whose correlation probability p_s changes between environments. The correlation probability p_i between digit and label remains the same across all environments. Therefore, in this dataset color is the spurious feature and digit is the invariant feature. This dataset follows the data generating model in Figure 1.

We use two versions of CMNIST, CMNISTa and CMNISTb, that are generated with two different sets of correlation probabilities: CMNISTa uses the same correlation probabilities as Arjovsky et al. (2019) with $p_i = 0.75$ for all environments, $p_s = 0.8, 0.9$ for the two training environments, and $p_s = 0.1$ for the testing environment; CMNISTb uses $p_i = 0.95$ for all environments, $p_s = 0.8, 1$ for the two training environments, and $p_s = 0.1$ for the testing environment.

We also use an abstract version of CMNIST introduced by Kamath et al. (2021), called toy-CMNISTa/b, in which the image data is replaced by a two-bit data $x = [x_d, x_c]$, where $x_d \in \{-1, 1\}$ corresponds to the label of the digit (without label noise) and $x_c \in \{-1, 1\}$ corresponds to the color.

6.2 Result

In addition to IRM-v1 and MRI-v1, we report the performance of the ERM algorithm, which minimizes the average risk across environments without any invariance constraints (Table 2). For synthetic datasets, we also report the performance of the *Oracle* (Table 2), which is trained using ERM on the dataset in which the spurious features are rendered uncorrelated with the label. For the shape-texture datasets, we use a small convolutional network (single convolution layer), while for the other image-based datasets, we use the LeNet architecture (LeCun et al., 1998).

Shape-Texture Tasks For both regression and classification tasks, we find that IRM-v1 trained using PM always converges to the zero-predictor (Table 2-3; Figure 5). In comparison, MRI-v1 trained using the penalty method achieved near-optimal OOD generalization and performed close to Oracle (Table 2-3; Figure 5). This demonstrates that MRI-v1 is far superior to IRM-v1 in terms of both loss and accuracy when trained using the penalty method.

Similar to the PM case, MRI-v1 is able to perform close to Oracle when trained using ALM (Table 2-3). In comparison, while IRM was able to avoid the zero-predictor, it still achieved worse test loss and accuracy than MRI-v1. Nevertheless, by using ALM, we were able to significantly improve IRM-v1's performance by avoiding the zero-predictor, while MRI-v1 performed similarly irrespective of the training method as the the zero-predictor is not a local constrained optimum of MRI-v1 (ALM slightly improves MRI-v1's performance).

Colored MNIST Similar to the shape-texture tasks, IRM-v1 always converges to the null-predictor when trained using the penalty method while MRI-v1 achieves a much better test loss and accuracy that IRM-v1 and ERM (Table 2- 3; Figure 6). This results for MRI-v1 holds true also when trained using ALM, albeit with a slightly improved performance (Table 2- 3; Figure 7). In comparison, similar to the shape-texture tasks, IRM-v1 is able to avoid the null predictor when trained with ALM, but it achieves worse test loss than MRI-v1 on both the datasets (Table 2; Figure 7), especially for the CMNISTb datasets. In contrast to the loss results, IRM-v1 achieves a similar test accuracy to MRI-v1 for both the datasets (Table 3; Figure 7).

	S-T Reg.		S-T Cls.		CMNISTa		CMNISTb	
	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$
Oracle	0.46 ± 0.00	0.46 ± 0.00	0.47 ± 0.00	0.47 ± 0.00	0.57 ± 0.00	0.58 ± 0.00	0.22 ± 0.00	0.24 ± 0.01
ERM	0.16 ± 0.00	1.37 ± 0.01	0.24 ± 0.00	1.16 ± 0.01	0.36 ± 0.00	1.44 ± 0.01	0.13 ± 0.00	0.73 ± 0.01
IRM-v1 (PM)	1.00 ± 0.00	1.00 ± 0.00	0.69 ± 0.00	0.69 ± 0.00	0.69 ± 0.00	0.69 ± 0.00	0.69 ± 0.00	0.69 ± 0.00
IRM-v1 (ALM)	0.23 ± 0.00	0.62 ± 0.01	0.4 ± 0.01	0.51 ± 0.01	0.62 ± 0.02	0.69 ± 0.02	0.17 ± 0.01	0.47 ± 0.02
MRI-v1 (PM)	0.53 ± 0.03	0.54 ± 0.03	0.44 ± 0.01	$\mathbf{0.46} \pm 0.01$	0.62 ± 0.01	0.66 ± 0.01	0.46 ± 0.01	0.4 ± 0.01
MRI-v1 (ALM)	0.45 ± 0.01	$\mathbf{0.46} \pm 0.00$	0.47 ± 0.00	0.47 ± 0.00	0.63 ± 0.02	$\mathbf{0.64} \pm 0.01$	0.25 ± 0.01	$\mathbf{0.29} \pm 0.01$

Table 2: Average train domain risk $\mathcal{L}_{\text{tr}}$ and test domain risk $\mathcal{L}_{\text{test}}$ for image-based experiments. Oracle uses environments in which the spurious features are uncorrelated with the label. Mean and standard deviation shown up to 2 decimal places.

7 Discussion

Limitations of the accuracy metric The accuracy metric is widely used in the field of domain generalization (Gulrajani and Lopez-Paz, 2020) to compare the performance of various algorithms. While accuracy does provide some insight into the performance of an algorithm, it is not always a sensitive enough metric to discriminate between the representations learned by different algorithms. For example, we find that for the toy-CMNISTb dataset, both invariant ($(w_i, w_s) \approx (3, 0)$) and non-invariant predictors ($(w_i, w_s) \in \{(3, -2), (3, 2)\}$) achieve the same accuracy (Figure 8). In comparison, the loss metric (Figure 8) can indeed differentiate between the invariant and non-invariant representations. This discrepancy between the loss and accuracy metric holds true even for non-linear image-based datasets (Table 2-5). To our knowledge, such discrepancy between loss and accuracy has not been reported before in the domain generalization literature. Based on this, we suggest that the reliance over accuracy as the sole metric of comparison between domain generalization algorithms can be misleading, but loss and accuracy taken together can provide a meaningful comparison.

Limitations A limitation of both MRI and IRM (Rosenfeld et al., 2020; Ahuja et al., 2021) is that it requires significant overlap in the support of different domain distributions. This limits the applicability of both MRI and IRM on most DomainBed (Gulrajani and Lopez-Paz, 2020) benchmarks. Rotated-MNIST, for example, expects an algorithm to generalize to an orientation of the digit that was never seen in the training set. In this case, the support of the training and testing set clearly does not overlap. Similar arguments can be made for other datasets. For example, images with different styles across testing and training domains may not satisfy the significant support overlap condition.

IRM and MRI's domain of applicability only coincide when there is no label shift or shift invariant feature distribution. For datasets that do not follow these conditions, MRI, but not IRM, should fail. However, note that, while IRM is expected to work for such shifts, IRM-v1 fails for such domain shifts (Ahuja et al., 2020, 2021). This is because IRM-v1 and IRM are not the same, as demonstrated by us and Kamath et al. (2021).

References

Ahuja, K., Caballero, E., Zhang, D., Bengio, Y., Mitliagkas, I., and Rish, I. (2021). Invariance principle meets information bottleneck for out-of-distribution generalization. *arXiv preprint arXiv:2106.06607*.

Ahuja, K., Wang, J., Dhurandhar, A., Shanmugam, K., and Varshney, K. R. (2020). Empirical or invariant risk minimization? a sample complexity perspective. *arXiv preprint arXiv:2010.16412*.

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. (2019). Adversarial invariant feature learning with accuracy constraint for domain generalization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 315–331. Springer.

Albuquerque, I., Monteiro, J., Darvishi, M., Falk, T. H., and Mitliagkas, I. (2019). Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*.

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

Bertsekas, D. P. (1976). Multiplier methods: A survey. *Automatica*, 12(2):133–145.

Creager, E., Jacobsen, J.-H., and Zemel, R. (2021). Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR.

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

Ghifary, M., Balduzzi, D., Kleijn, W. B., and Zhang, M. (2016). Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. (2016). Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848. PMLR.

Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*.

Heinze-Deml, C., Peters, J., and Meinshausen, N. (2018). Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2).

Hu, S., Zhang, K., Chen, Z., and Chan, L. (2020). Domain generalization via multidomain discriminant analysis. In *Uncertainty in Artificial Intelligence*, pages 292–302. PMLR.

Ilse, M., Tomczak, J. M., Louizos, C., and Welling, M. (2020). Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*, pages 322–348. PMLR.

Kamath, P., Tangella, A., Sutherland, D., and Srebro, N. (2021). Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pages 4069–4077. PMLR.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2018a). Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Li, Y., Gong, M., Tian, X., Liu, T., and Tao, D. (2018b). Domain generalization via conditional invariant representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2018). Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31.

Mahajan, D., Tople, S., and Sharma, A. (2021). Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR.

Muandet, K., Balduzzi, D., and Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR.

Pearl, J. (2009). *Causality*. Cambridge university press.

Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012.

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. (2018). Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342.

Rosenfeld, E., Ravikumar, P., and Risteski, A. (2020). The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*.

Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., and Sarawagi, S. (2018). Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*.

Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. (2021). On calibration and out-of-domain generalization. *Advances in Neural Information Processing Systems*, 34.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section ??.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3 and 5.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 3 and 5.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] `https://anonymous.4open.science/r/MRI-2608/readme`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Supplementary Materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 2 and 3.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplementary Materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 6.1 and Supplementary Materials.
   (b) Did you mention the license of the assets? [Yes] See Supplementary Materials.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] `https://anonymous.4open.science/r/MRI-2608/readme`
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Analytical Solution for Toy Shape-Texture Regression

The Shape-Texture dataset involves an invariant latent feature $\theta_i \in \mathbb{R}$, a spurious latent feature $\theta_s \in \mathbb{R}$, and a latent feature $\theta_y \in \mathbb{R}$, each of which represents an angle of orientation, and therefore is naturally represented on a unit circle in the complex plane as $e^{i\theta} \in \mathcal{S}^1$. In the toy linear regression setting, the observed input is simply the concatenated features $x = \begin{bmatrix} e^{i\theta_i} \\ e^{i\theta_s} \end{bmatrix}$ and the label is $y = e^{i\theta_y}$.

For all task settings, the latent angles and the label angle are linearly related as $\theta_i \leftarrow \theta_y + \xi_i, \theta_s \leftarrow \theta_y + \xi_s$, where $\xi_* \sim \mathcal{N}(0, \sigma_*)$ with probability p_*, or $\xi_* \sim \mathcal{U}(-\pi, \pi)$ probability $1 - p_*$ for $* \in \{i, s\}$. We used $p_i = 0.75$ for all environments, $p_{s_1} = 0.8, p_{s_2} = 1$ for the two training environments, $p_{s_{\text{test}}} = 0$ for the testing environment, and $\sigma_i = \sigma_{s_e} = 0$ for all environments.

Note that

$$\mathbb{E}_e[e^{i(\theta_i - \theta_y)}] = \begin{cases} 1, & \text{with prob. } p_i \\ 0, & \text{with prob. } 1 - p_i \end{cases}$$

$$\mathbb{E}_e[e^{i(\theta_s - \theta_y)}] = \begin{cases} 1, & \text{with prob. } p_s \\ 0, & \text{with prob. } 1 - p_s \end{cases}$$

$$\mathbb{E}_e[e^{i(\theta_i - \theta_s)}] = \begin{cases} 1, & \text{with prob. } p_i p_s \\ 0, & \text{with prob. } 1 - p_i p_s \end{cases}$$

which yields

$$\mathbb{E}_e[xx^\dagger] = \begin{bmatrix} 1 & \mathbb{E}_e[e^{i(\theta_i - \theta_s)}] \\ \mathbb{E}_e[e^{-i(\theta_i - \theta_s)}] & 1 \end{bmatrix} = \begin{bmatrix} 1 & p_i p_s \\ p_i p_s & 1 \end{bmatrix} \tag{20}$$

$$\mathbb{E}_e[xy^\dagger] = \begin{bmatrix} \mathbb{E}_e[e^{i(\theta_i - \theta_y)}] \\ \mathbb{E}_e[e^{i(\theta_s - \theta_y)}] \end{bmatrix} = \begin{bmatrix} p_i \\ p_s \end{bmatrix} \tag{21}$$

We consider real weights $w = [w_i \quad w_s]$. Since $o = wx$,

$$\mathbb{E}_e[oo^\dagger] = w\mathbb{E}_e[xx^\dagger]w^\intercal = w_i^2 + w_s^2 + 2w_i p_i w_s p_{s_e},$$
$$\mathbb{E}_e[oy^\dagger] = w\mathbb{E}_e[xy^\dagger] = w_i p_i + w_s p_{s_e}$$

The risk of environment e is

$$\mathcal{L}_e(f) = \mathbb{E}_e[\|o - y\|^2] = \mathbb{E}_e[oo^\dagger] - (\mathbb{E}_e[oy^\dagger] + \mathbb{E}_e[oy^\dagger]^\dagger) + \mathbb{E}_e[yy^\dagger].$$

Therefore, the average risk for training is

$$\bar{\mathcal{L}}_{tr}(f) = \frac{1}{2} \sum_{e \in \{1,2\}} (w_i^2 + w_s^2 + 2w_i p_i w_s p_{s_e} - 2(w_i p_i + w_s p_{s_e}) + 1) \tag{22}$$

MRI constraint

$$\begin{aligned} &\mathbb{E}_{e_1}[oy] = \mathbb{E}_{e_2}[oy] \\ &\quad \implies w_i p_i + w_s p_{s_1} = w_i p_i + w_s p_{s_2} \\ &\quad \implies w_s = 0 \end{aligned} \tag{23}$$

Minimizing $\bar{\mathcal{L}}_{tr}(f)$ subject to the above constraint yields $(w_i^*, w_s^*) = (p_i, 0)$ as the constrained optimum.

IRM-relaxed constraint

$$\begin{aligned} \mathbb{E}_{e_1}[oo] - \mathbb{E}_{e_1}[oy] &= \mathbb{E}_{e_2}[oo] - \mathbb{E}_{e_2}[oy] \\ &\implies (2w_i - p_i) w_s p_{s_1} = (2w_i - p_i) w_s p_{s_2} \\ &\implies \begin{cases} w_s = 0 \\ w_i = \frac{1}{2p_i} \end{cases} \end{aligned} \tag{24}$$

Minimizing $\bar{\mathcal{L}}_{tr}(f)$ subject to the above constraint yields

$$(w_i^*, w_s^*) = \begin{cases} (p_i, 0) \\ (\frac{1}{2p_i}, \frac{p_{s_1}+p_{s_2}}{4}) \end{cases}$$

as the constrained optima.

IRM constraint

$$\forall e \in \{1,2\} \qquad \mathbb{E}_e[oo] - \mathbb{E}_e[oy] = w_i^2 + w_s^2 + 2w_i p_i w_s p_{s_e} - w_i p_i - w_s p_{s_e} = 0 \tag{25}$$

which yields

$$(w_i^*, w_s^*) = \begin{cases} (0, 0) \\ (p_i, 0) \\ \left(\frac{1}{2p_i}, \frac{\sqrt{2p_i^2-1}}{2p_i}\right) \\ \left(\frac{1}{2p_i}, -\frac{\sqrt{2p_i^2-1}}{2p_i}\right) \end{cases} \tag{26}$$

as constraints, which are also the constrained optima of the problem.

B Training Procedure

PyTorch version 1.10.0 was used. For all tasks, Tesla V-100 with a memory of 16GB was used. For all our simulations, we utilized the DomainBed suite (Gulrajani and Lopez-Paz (2020); available under MIT license). For a given task, we use the same training procedure for all the algorithms unless mentioned otherwise. The details of the training are described as follows.

B.1 Linear Datasets

For Figure 2, 3, and 8, we used a linear predictor $f(x) = w_i z_i + w_s z_s$. In all cases, we used full-batch training with a batch size of 40000. We optimized using the penalty method with $\mu = 5 \times 10^4$. For Figure 2, we used the SGD optimizer with no momentum. For Figure 3 and 8, we used the Adam optimizer with betas of (0.975, 0.999). The learning rate was set to 0.005, the total number of gradient steps were set to 2000, and the gradient norms were clipped at 2 for all cases.

B.2 Non-Linear Datasets

The results for the non-linear datasets are presented in Table 2-5 and Figure 5-7.

Shape-Texture Datasets For the shape-texture datasets, we used a single convolutional layer with 12 channels, a kernel size of 5, and stride of 1. The convolutional layer was followed by a 2D adaptive average pooling layer with output size of 1x1 followed by a fully-connected linear layer. For results in Table 2 and 3, for optimization using the ALM method we use $\mu = 10$ and $\lambda = 10$ and for optimization using PM we use $\mu = 10000$. We used mini-batch training with a mini-batch size 8000 and total dataset size 50000. We used the Adam optimizer with betas of (0.9, 0.999). The learning rate was set to 0.01. The total number of gradient steps were set to 2000. Gradient clipping was only used when training using PM such that gradient norms were clipped at 2. We used a weight decay of 0.003 for the classification setting and 0.005 for the regression setting.

For Figure 5 we use the same training procedure as above for the penalty method except that the seeds, learning rates and algorithm-specific parameters were chosen at random to generate 50 random hyperparameter combinations for each algorithm. The learning rates were chosen randomly from the distribution $10^{Uniform[-3,-1.5)}$. For IRM-v1 and MRI-v1, μ was chosen randomly from the distribution $10^{Uniform[2,5)}$. We used DomainBed's (Gulrajani and Lopez-Paz, 2020) implementation of IB-IRM, GroupDRO, and MMD. For IB-IRM, λ_{IRM} and λ_{IB} were both chosen randomly from the distribution $10^{Uniform[2,5)}$. For GroupDRO, $\eta_{GroupDRO}$ was chosen from the distribution $10^{Uniform[-3,1)}$. For MMD, γ_{MMD} was chosen from the distribution $10^{Uniform[-1,1)}$

CMNIST For the CMNIST datasets, we use the LeNet architecture (LeCun et al., 1998). We used mini-batch training with a mini-batch size of 256. For optimization using the ALM method with $\lambda = 10$ for both datasets, but $\mu = 1$ for CMNISTa and $\mu = 10$ for CMNISTb datasets for results in Table 2, 3. For optimization using the PM method, we used $\mu = 10000$ for both datasets. We used the Adam optimizer with betas of (0.975, 0.999). The learning rate was set to 0.001, the total number of gradient steps were set to 10000. The weight decay was set to 0.001. Gradient clipping was only used when training using PM such that gradient norms were clipped at 2. For the CMNIST datasets, we also used data augmentation during training. This was done using Torchvision's Random affine transformation class with degrees=30, translate=(0.1, 0.1), and scale=(0.8, 1.2).

For Figure 6 we used the same training procedure as above for training with PM except μ was chosen randomly from the distribution $10^{Uniform[2,5)}$. For Figure 7, we used the same training procedure as above for training with ALM except both λ and μ were chosen independently and randomly from the distribution $Uniform[1, 10)$. For both the figures, seeds were chosen at random. The total number of gradient steps were set to 5000 for Figure 6 and 15000 for Figure 7.

VLCS/Terra-Incognita For both VLCS and Terra-Incognita datasets, we restrict the problem to a binary classification task by using only two classes for each of the two datasets, namely opossum and raccoon for the TI dataset and bird and person for VLCS dataset. The results are presented in Table 4,5. For both datasets, we use the same training procedure. We used mini-batch training with a mini-batch size of 256. We trained using the ALM method with $\mu = 10$ and $\lambda = 10$. We used the Adam optimizer with betas of (0.975, 0.999). The learning rate was set to 0.001 and the total number of gradient steps were set to 2000. The weight decay was set to 0.001. No gradient clipping was used. All the images were resized to the shape (64,64,3) and were normalized with means (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225) corresponding to each channel. We also used data augmentation during training. This was done using Torchvision's RandomHorizontalFlip transform, RandomGrayscale transform, RandomResizedCrop transform with scale = (0.7, 1.0), and ColorJitter transform brightness, contrast, saturation, and hue set to 0.3.

	S-T Cls.		CMNISTa		CMNISTb	
	Train domains	Test domain	Train domains	Test domain	Train domains	Test domain
Oracle	0.86 ± 0.0	0.86 ± 0.0	0.74 ± 0.0	0.74 ± 0.0	0.94 ± 0.0	0.94 ± 0.0
ERM	0.95 ± 0.0	0.50 ± 0.00	0.85 ± 0.0	0.10 ± 0.00	0.94 ± 0.0	0.80 ± 0.02
IRM-v1 (PM)	0.52 ± 0.02	0.52 ± 0.04	0.73 ± 0.14	0.23 ± 0.16	0.6 ± 0.03	0.52 ± 0.05
IRM-v1 (ALM)	0.79 ± 0.01	0.77 ± 0.01	0.64 ± 0.03	0.66 ± 0.03	0.93 ± 0.0	0.91 ± 0.0
MRI-v1 (PM)	0.86 ± 0.01	0.85 ± 0.01	0.68 ± 0.01	0.63 ± 0.02	0.82 ± 0.01	0.86 ± 0.01
MRI-v1 (ALM)	0.86 ± 0.01	0.86 ± 0.01	0.66 ± 0.02	0.65 ± 0.02	0.93 ± 0.0	0.9 ± 0.0

Table 3: Accuracy comparison for image-based experiments. Oracle uses environments in which the spurious features are uncorrelated with the label. Mean and standard deviation shown up to 2 decimal places.

	VLCS		Terra-Incognita	
	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$
ERM	0.12 ± 0.0	0.96 ± 0.02	0.41 ± 0.01	1.64 ± 0.16
IRM-v1 (ALM)	0.21 ± 0.02	2.02 ± 0.07	0.52 ± 0.02	2.13 ± 0.45
MRI-v1 (ALM)	0.16 ± 0.01	0.74 ± 0.02	0.43 ± 0.01	1.36 ± 0.07

Table 4: Average train domain risk $\mathcal{L}_{\text{tr}}$ and test domain risk $\mathcal{L}_{\text{test}}$ for VLCS and Terra-Incognita datasets. Mean and standard deviation shown up to 2 decimal places.

	VLCS		Terra-Incognita	
	Train domains	Test domain	Train domains	Test domain
ERM	0.96 ± 0.0	0.75 ± 0.01	0.81 ± 0.01	0.38 ± 0.02
IRM-v1 (ALM)	0.96 ± 0.0	0.75 ± 0.01	0.76 ± 0.01	0.38 ± 0.02
MRI-v1 (ALM)	0.94 ± 0.0	0.74 ± 0.01	0.79 ± 0.01	0.38 ± 0.01

Table 5: Accuracy comparison for VLCS and Terra-Incognita datasets. Mean and standard deviation shown up to 2 decimal places.

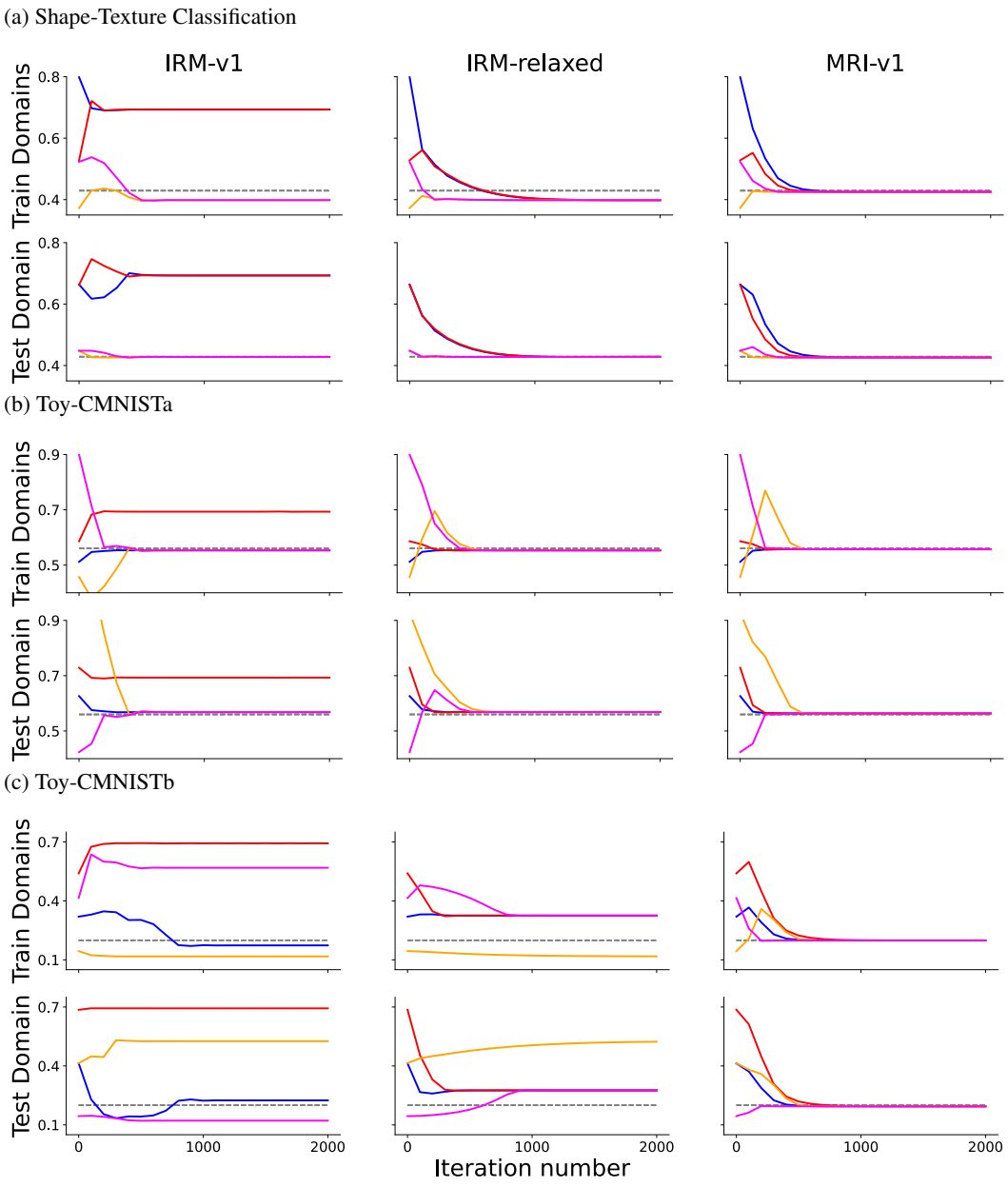
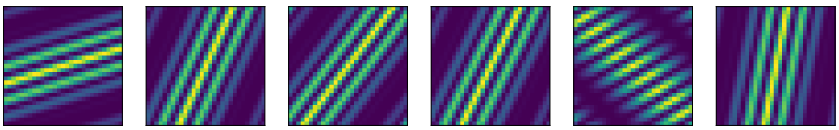
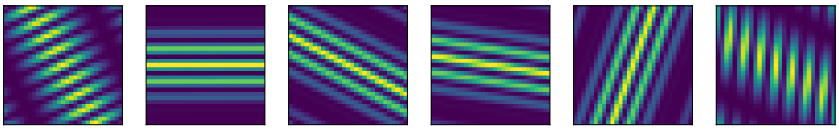


Figure 3: Linear classification tasks. Train (top) and test (bottom) domain loss corresponding to four different weight initialization for toy Shape-Texture classification dataset (a), toy-CMNISTa dataset (b), and toy-CMNISTb dataset (c). Each color corresponds to a different weight initialization. Black dotted line shows the optimal invariant predictor performance.

(a) Training domain 1 ($p_i = 0.75,\ p_s = 1$)



(b) Training domain 2 ($p_i = 0.75,\ p_s = 0.8$)



(c) Test domain ($p_i = 0.75,\ p_s = 0$)

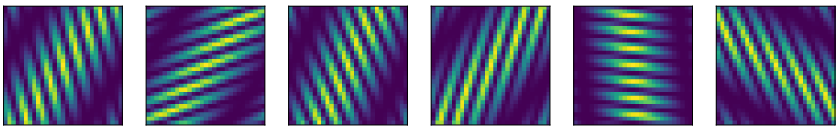
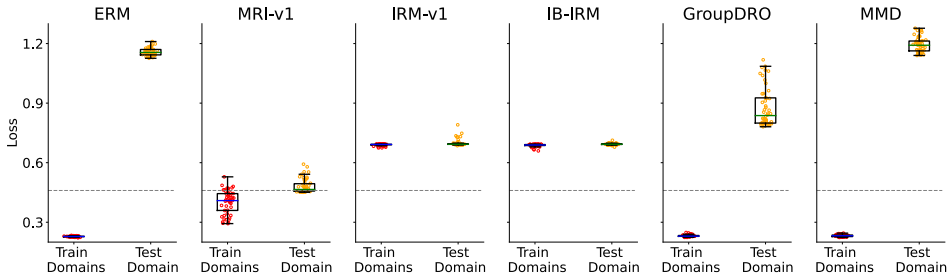


Figure 4: Sample Shape-Texture dataset images.

(a) Loss



(b) Accuracy

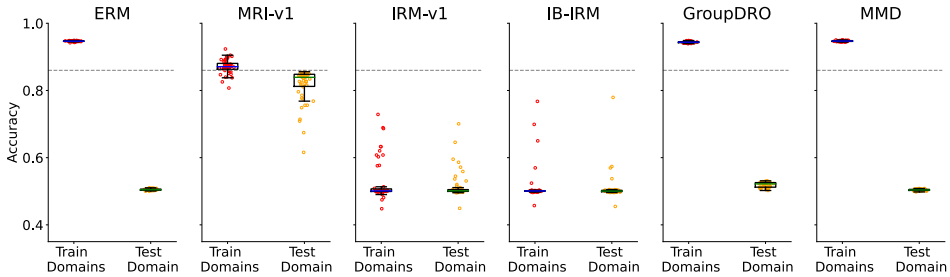


Figure 5: Train and test domain loss (a) and accuracy (b) on shape-texture classification dataset for 50 random hyperparameters for the following algorithms: ERM, MRI-v1 and IRM-v1 trained using PM, IB-IRM, GroupDRO and MMD. Black dotted line denotes the Oracle performance.

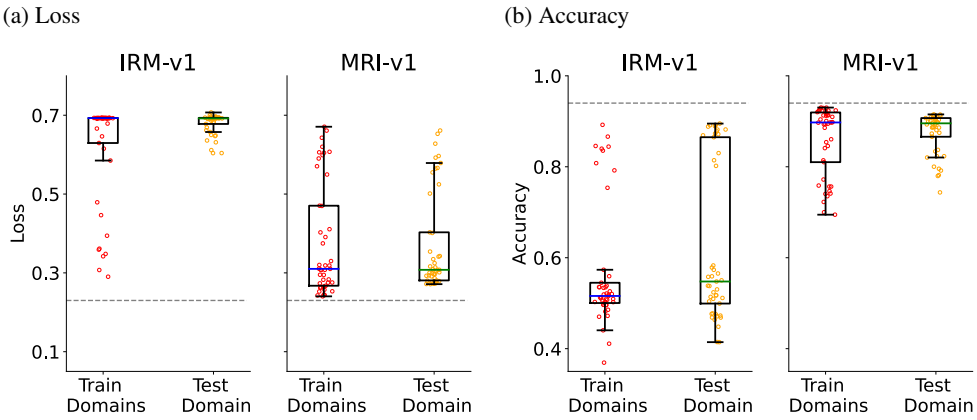


Figure 6: Train and test domain loss (a) and accuracy (b) on CMNISTb dataset for 50 random hyperparameters for IRM-v1 and MRI-v1 trained using PM. Black dotted line denotes the Oracle performance.

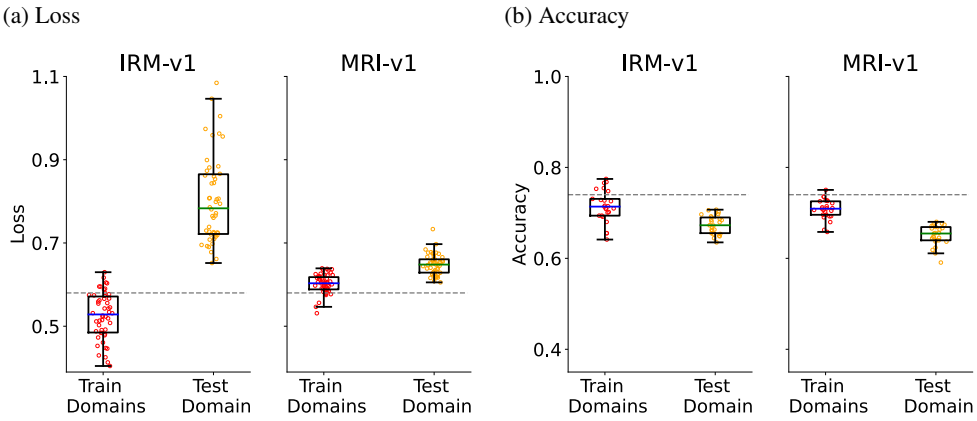


Figure 7: Train and test domain loss (a) and accuracy (b) on CMNISTa dataset for 50 random hyperparameters for IRM-v1 and MRI-v1 trained using ALM. Black dotted line denotes the Oracle performance.

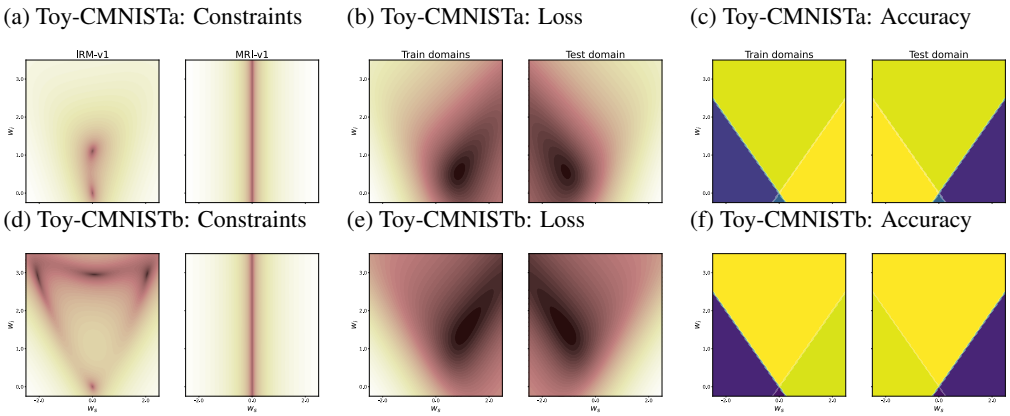


Figure 8: Toy-CMNISTa/b datasets. Results are demonstrated for a linear predictor (Eq 18). (a) Invariance constraints for IRM-v1 (left) and MRI-v1(right) on toy-CMNISTa dataset. Y-axis: weight corresponding to the invariant feature w_i (Eq 18). X-axis: weight corresponding to the invariant feature w_s (Eq 18). (b) Averaged risk over training (left) and testing (right) domains as a function of the predictor weights on toy-CMNISTa dataset. Axes: same as panel a. (c) Averaged accuracy over training (left) and testing (right) domains as a function of the predictor weights on toy-CMNISTa dataset. Axes: same as panel a. (d)-(f) Same as panels a-c except on toy-CMNISTb dataset.