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

Dongsung Huh*
MIT-IBM Watson AI Lab
Cambridge, MA 02142
huh@ibm.com

Avinash Baidya*
Department of Physics and Astronomy
University of California
Davis, CA 95616
aavinash@ucdavis.edu

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 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 $\mathbb{E}_e[f(x)|y]$, 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 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

*Equal contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

fundamental flaw of IRM formulation that causes this limitation and propose a new method that is free of this flaw.

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.[2]

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 (Rosenfeld et al., 2020), which includes latent features that are invariantly z_i or spuriously z_s correlated with the label, from which the observation x is generated.[3]

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

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

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.

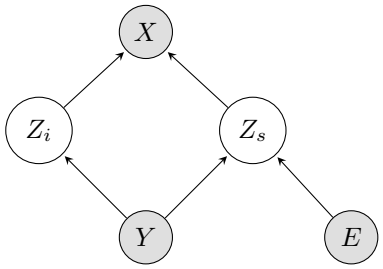


Figure 1: Causal graph depicting the data generating process. Shading indicates the variable is observed.

3 IRM vs MRI Invariance

3.1 IRM Paradigm

3.1.1 Original formulation

Definition 3.1. (Arjovsky et al., 2019) Given a predictor $\psi \circ f : X \to O$, 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

[2]Code available at `https://github.com/IBM/MRI`.

[3]While Fig 1 considers the causal direction $Y \to Z_i$, our results on IRM and MRI are also consistent with the causal direction $Z_i \to Y$.

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[4], 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 the following variational formulation.

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[4], 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 $\mathbb{E}_e[\partial_o l(o, y)|o] = -\mathbb{E}_e[y|o] + \sigma(o) = 0$. □

Definition 3.3 describes the first-order condition for the predictor f to be simultaneously optimal over all environments. This is indeed a sufficient condition for optimality given a convex loss function l. This yields a simpler formulation of IRM without requiring a composite form for the predictor $\psi \circ f$.

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 imposes that value to be also equal to the RHS term $\sigma(o)$. This extra constraint reflects the indirect way IRM attains the conservation, *i.e.* by equating $\mathbb{E}_e[y|o]$ to an intermediary term $\sigma(o)$, imposing 1 constraint per environment.

[4] 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).[5] 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 widely 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 $\partial_o l(o, y) \cdot o = o(\sigma(o) - y)$ for standard loss functions[4], eq (9) is equivalent to

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

Unlike in eq (7), however, RHS of eq (10) is not 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 $\partial_y l(o, y) = -o + \rho(y)$[6]. 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 attains invariance in a direct manner without involving any intermediary terms, thus imposing the correct number of constraints in eq (11) as described by the conservation law eq (13).

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

[5] IRM-s imposes $\partial_\psi \mathbb{E}_e[\,l(\psi \cdot o, y)] = 0$, where ψ is a scalar factor. Note that $\partial_\psi l(\psi \cdot o, y) = \partial_o l(o, y) \cdot o$.

[6] 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	$\delta\vec{\mathcal{L}}(f)$	$\mathbb{E}_e[\,\partial_o l(o,y)\cdot o\,]$
MRI-v1	$Q\,\delta\vec{\mathcal{L}}(f)$	$\mathbb{E}_e[\,\partial_y l(o,y)\cdot y\,]$

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

4 Methods

4.1 Constrained optimization problem

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

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

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 w is the parameters of the predictor f. The constraint functions are $\vec{c} = \delta\vec{\mathcal{L}} \in \mathbb{R}^{|\mathcal{E}_{\text{tr}}|}$ for IRM-v1 and $\vec{c} = Q\,\delta\vec{\mathcal{L}} \in \mathbb{R}^{|\mathcal{E}_{\text{tr}}|-1}$ for MRI-v1, where $\delta\vec{\mathcal{L}} \in \mathbb{R}^{|\mathcal{E}_{\text{tr}}|}$ is a vector of perturbed risks $\delta\mathcal{L}_e$ for $e \in \mathcal{E}_{\text{tr}}$. $Q \in \mathbb{R}^{(|\mathcal{E}_{\text{tr}}|-1)\times|\mathcal{E}_{\text{tr}}|}$ is an orthonormal matrix that satisfies $Q\,\vec{1} = \vec{0}$, such that $Q\,\delta\vec{\mathcal{L}}$ computes the differences of $\delta\mathcal{L}_e$ between environments. For example, for a training set of two environments $\mathcal{E}_{\text{tr}} = \{e_1, e_2\}$, $Q\,\delta\vec{\mathcal{L}} = (\delta\mathcal{L}_{e_1} - \delta\mathcal{L}_{e_2})/\sqrt{2}$, 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 $\vec{c}(f) = \vec{0}$ 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,

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

However, this method requires increasing $\mu^t \to \infty$ over training iteration 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)

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

where $\vec{\lambda}$ is typically initialized at $\vec{0}$ and updated at each training iteration t to accumulate the residual constraints $\vec{c}(f(w^t))$. 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 of a linear predictor given a sufficient number of environments. Consider a data generating process according to Fig 1, in which the observation $x = g(z_i, z_s)$ is an injective linear function of the latent features $z_i \in \mathbb{R}^{d_i}$, $z_s \in \mathbb{R}^{d_s}$. Note that this Structural Equation Model (SEM) (Pearl, 2009) does not require any assumptions on the generation process $Y \to Z_i, Z_s$, which generalizes the SEM of Rosenfeld et al. (2020), which additionally assumed binary labels and additive Gaussian noise for generating the latent features.

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

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

with $w = \{w_i \in \mathbb{R}^{d_i}, w_s \in \mathbb{R}^{d_s}\}$.

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

Proof. MRI-v1's constraint eq (14) yields

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

which 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 $w_s = 0$.

□

A similar result was shown for IRM-v1, but under a more restricted setting, such as a specialized linear family of environments with binary labels and additive Gaussian noise (Rosenfeld et al., 2020). In fact, IRM-v1 has been shown to fail in more general linear problems with non-Gaussian noise (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 a minimal case of the above general linear problem that involves one invariant feature, one spurious feature and two training environments (Shape-Texture linear regression problem in Section 6.1). See Supplementary Materials for the detailed experimental set up and the analytical solutions. A similar minimal examples for linear binary classification are also analyzed and shown in Supplementary Materials (Fig 4, linear shape-texture classification and toy-CMNISTa/b).

Analytic solutions (Hard-constraints, Fig 2A) IRM-v1 has two quadratic equality constraints, $\mathbb{E}_{e_1}[o^2 - oy] = 0 = \mathbb{E}_{e_2}[o^2 - oy]$, 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 train loss than the true invariant optimum solution.

We also test the relaxed version of IRM-v1 by removing the extraneous constraint. The relaxed version has a single constraint $\mathbb{E}_{e_1}[o^2 - oy] = \mathbb{E}_{e_2}[o^2 - oy]$, which describes a pair of hyperbolic curves (appears as two straight lines in Fig 2A), *i.e.* a 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 a lower train loss. Therefore, simply relaxing the extraneous constraint does not resolve the fundamental problem of IRM-v1.

In contrast, MRI-v1 has one linear equality constraint, $\mathbb{E}_{e_1}[oy] = \mathbb{E}_{e_2}[oy]$, that exactly prescribes the set of all invariant solutions $w_s = 0$. That is, a solution is an invariant predictor *if and only if it satisfies this constraint.* 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.

Convergence Dynamics (Soft-constraints, Fig 2B/C) Here, we analyze the optimization dynamics under full-batch gradient descent (penalty method eq (16) with $\mu = 5 \times 10^4$). IRM-v1's convergence is highly dependent on the initialization (shown with differently colored trajectories), due to the presence of multiple local minima. Note that most trajectories do not converge to the true invariant optimum. IRM-relaxed also exhibits complex dynamics due to a saddle point near the true invariant solution: Some trajectories (red and magenta) first approach the line of invariant solutions, but all trajectories eventually converge to the non-invariant solution. Note that the true invariant optimum solution is not even a local optimum of IRM-relaxed at this value of μ, but it would exist in the limit $\mu \to \infty$. In contrast, MRI-v1 always converges to the true invariant optimum regardless of initialization since it is a unique minimum.

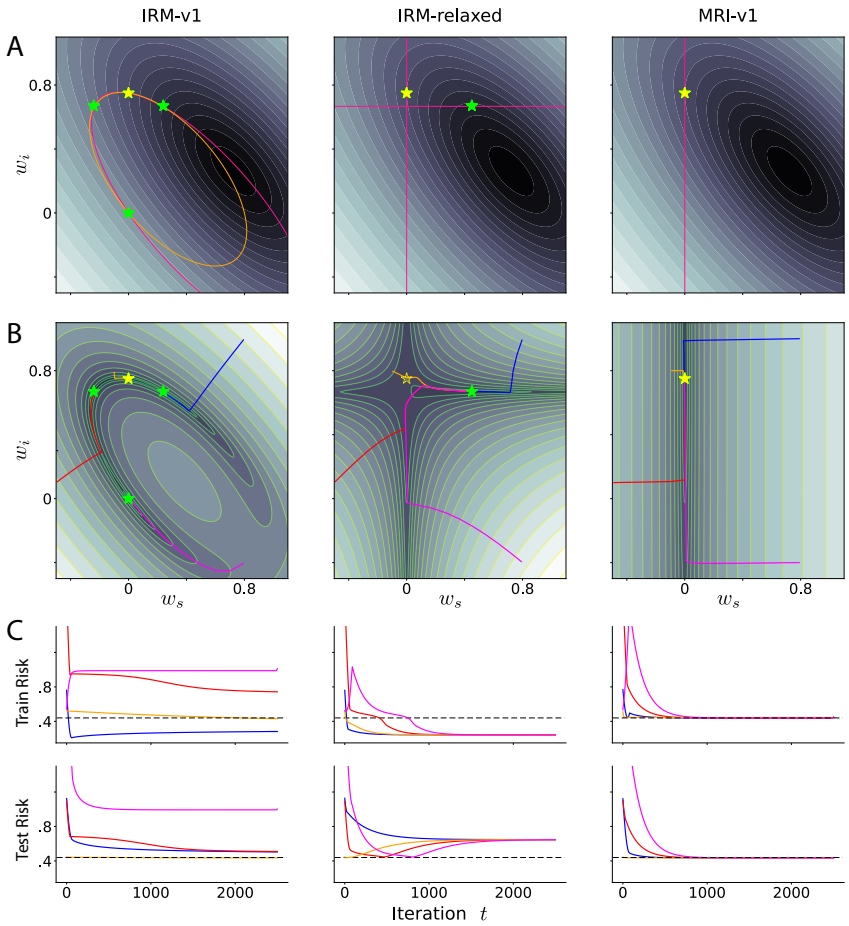


Figure 2: Minimal example in Section 5.2. (A) Hard-constraint case. Background: Average train domain risk $\mathcal{L}_{\text{tr}}(f(w))$. Lines: Invariance constraints $\vec{c}(f(w)) = \vec{0}$. Stars: Local constrained optima. (B) Soft-constraint case. Background: Train loss $\mathcal{L}_{\text{tr}}(f(w)) + \mu \left\|\vec{c}(f(w))\right\|^2$. Convergence trajectories w^t from multiple initialization are shown. (C) Loss profiles of the above convergence trajectories. Color-matched with panel B. (Top) Average train domain risk $\mathcal{L}_{\text{tr}}(f(w^t))$. (Bottom) Test domain risk $\mathcal{L}_{\text{test}}(f(w^t))$. Dashed line: True optimal invariant solution (yellow star in panel A/B).

6 Nonlinear Image-based Problems

Here, we explore how MRI-v1 performs in nonlinear problems. Note that unlike the linear problem, there is no guarantee for either IRM-v1 or MRI-v1 for these nonlinear problems.

6.1 Datasets

Shape-Texture Dataset We introduce a new versatile dataset designed for domain generalization on linear and nonlinear image-based regression and classification problems. The generative process of the dataset involves an invariant feature $z_i = e^{i\theta_i}$, a spurious feature $z_s = e^{i\theta_s}$, and a label feature $e^{i\theta_y}$, each of which represents an orientation on a complex unit circle: $e^{i\theta} \in \mathcal{S}^1$. The angles of orientations are generated as $\theta_y \sim \mathcal{U}_{S^1}$, where $\mathcal{U}_{S^1}$ is the circular uniform distribution, and for $* \in \{i, s\}$, $\theta_* = \theta_y$ with probability p_* or $\theta_* \sim \mathcal{U}_{S^1}$ with probability $1 - p_*$.

The parameter $p_i = 0.75$ is fixed across environments, whereas p_s varies from one environment to another. We consider two training environments $\mathcal{E}_{\text{tr}} = \{e_1, e_2\}$ with $p_{s_{e_1}} = 1,\ p_{s_{e_1}} = 0.8$ and one testing environment $\mathcal{E}_{\text{test}} = \{e_0\}$ with $p_{s_{e_0}} = 0$.

In the linear regression task (section 5.2), the observed input is simply the concatenated latent features $x = [e^{i\theta_i}, e^{i\theta_s}]$ and the label is $y = e^{i\theta_y}$. In the linear classification task, the input is

	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 $\mathcal{L}_{\text{test}}$ achieved by different methods in image-based tasks. Oracle uses environments in which the spurious features are uncorrelated with the label. Mean and standard deviation shown up to 2 decimal places.

$x = [\sin(\theta_i), \sin(\theta_s)]$ and the label is $y = H(\sin(\theta_y))$, where H is the sign function. In the nonlinear regression/classification tasks, the input observation is an image composed of two planar waves, in which θ_i is the orientation of the low frequency wave (*i.e.* shape) and θ_s is that of the high frequency wave (*i.e.* texture), as shown in Fig 5.

Colored MNIST (CMNIST) CMNIST (Arjovsky et al., 2019) is a synthetic dataset derived from MNIST for binary classification. In this dataset, the label y assigned to an image is based on the digit bit z_i (1 for digits $0 \sim 4$ and -1 for $5 \sim 9$) such that $y = z_i$ with probability p_i or $-z_i$ with probability $1 - p_i$. The color bit z_s (1 for red -1 for green) is chosen based on the label such that $z_s = y$ with probability p_s or $-y$ with probability $1 - p_s$. We consider two versions, CMNISTa and CMNISTb, with two sets of environmental parameters: CMNISTa is the version from Arjovsky et al. (2019) with $p_i = 0.75$, $p_{s_{e_1}} = 0.9,\ p_{s_{e_1}} = 0.8$, and $p_{s_{e_0}} = 0.1$ for the training $\mathcal{E}_{\text{tr}} = \{e_1, e_2\}$ and the testing $\mathcal{E}_{\text{test}} = \{e_0\}$ environments; CMNISTb uses $p_i = 0.9$, $p_{s_{e_1}} = 1,\ p_{s_{e_1}} = 0.8$, $p_{s_{e_0}} = 0.1$. In the nonlinear tasks, the input observation x is the colored MNIST image. In the abstracted versions, called toy-CMNISTa/b, the input observation is the two-bit data $x = [z_i, z_s]$ (Kamath et al., 2021).

Note that both Shape-Texture and CMNIST datasets can be equally understood as being generated from Fig 1 with causal directions $Y \to Z_i$ or $Z_i \to Y$ (Fig 9).

6.2 Result

In addition to IRM-v1 and MRI-v1, we also report the performance of the *ERM* algorithm, which minimizes the average empirical risk across environments without any invariance constraints (Table 2-3). For synthetic datasets, we also report the ERM's performance on the dataset in which the spurious features are made uncorrelated with the label (*Oracle* performance, Table 2-3). We used a small convolutional network (single convolution layer) for the image-based shape-texture datasets, and the LeNet architecture network (LeCun et al., 1998) for the other datasets.

We tested the algorithms under a wide range of hyperparameters for both the Shape-Texture (Fig 3) and the CMNIST-b (Fig 6) datasets. In both PM and ALM setting, we found that MRI-v1 showed significantly better test-domain performance than IRM-v1 overall in terms of loss (Table 2). This is also largely true for accuracy as well (Table 3), except that for the ALM setting of CMNIST-b, IRM-v1 shows slightly higher accuracy than MRI-v1.

In the appendix, we also report the results for other recent domain generalization methods like MMD (Li et al., 2018b), GroupDRO (Sagawa et al., 2019), and IB-IRM (Ahuja et al., 2021) on the shape-texture classification task, which showed performances very similar to ERM for a wide range range of hyperparameters (Fig 7) without showing much invariance.

7 Discussion

In this work, we introduce MRI, a novel method based on conserving the label-conditioned feature expectation $\mathbb{E}_e[f(x)|y]$ across environments. This method substantially outperforms IRM on certain

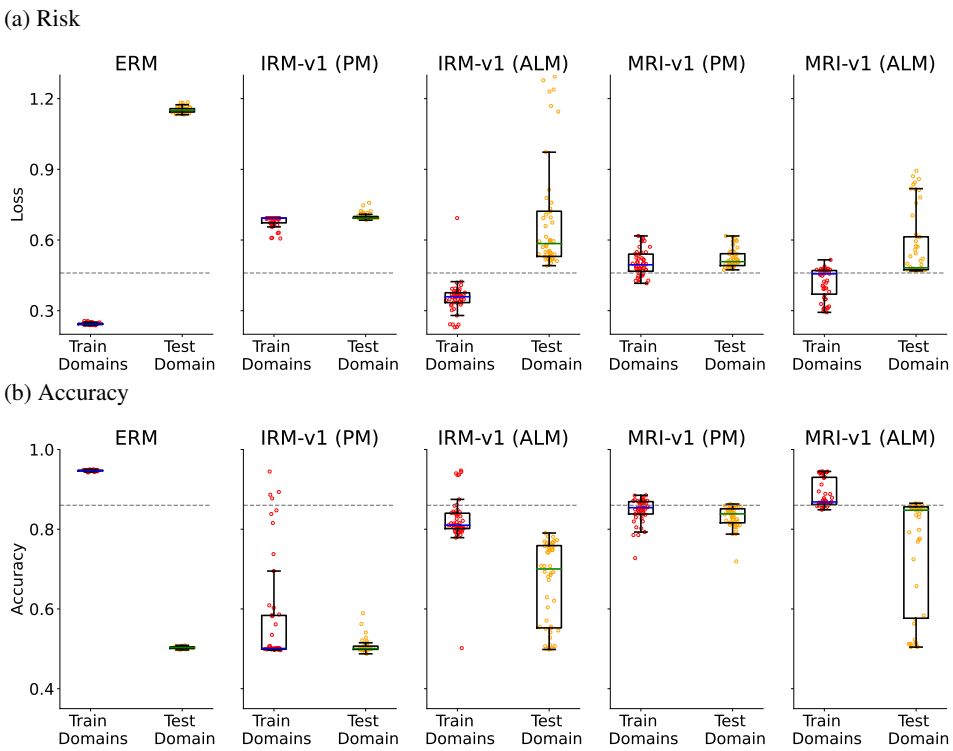


Figure 3: Performance of different algorithm under a range of hyperparameters on shape-texture classification dataset (a) Averaged train and test domain risk and (b) accuracy. Black dotted line denotes the Oracle performance.

	S-T Cls.		CMNISTa		CMNISTb	
	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$
Oracle	0.86 ± 0.00	0.86 ± 0.00	0.74 ± 0.00	0.74 ± 0.00	0.94 ± 0.00	0.94 ± 0.00
ERM	0.95 ± 0.00	0.50 ± 0.00	0.85 ± 0.00	0.10 ± 0.00	0.94 ± 0.00	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	$\mathbf{0.66} \pm 0.03$	0.93 ± 0.00	$\mathbf{0.91} \pm 0.00$
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	$\mathbf{0.86} \pm 0.01$	0.66 ± 0.02	0.65 ± 0.02	0.93 ± 0.00	0.9 ± 0.00

Table 3: Average train domain and test domain accuracy comparison achieved by different methods in image-based tasks. Oracle uses environments in which the spurious features are uncorrelated with the label. Mean and standard deviation shown up to 2 decimal places.

classes of OOD generalization problems, in both linear and non-linear settings. Here, we conclude with a discussion on the limitations of our method and the limitations of accuracy as a metric to compare the degree of invariance achieved by various algorithms.

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. However, accuracy is not sensitive enough to reliably measure the degree of invariance of a model. Consider the linear classification problem of toy-CMINSTb shown in Fig 8. For this problem, MRI-v1 has a unique optimum, which is the optimal invariant solution, while IRM-v1 exhibits multiple local optima, including non-invariant solutions. Yet, all these solutions exhibit identical accuracy (except for the zero-predictor solution of IRM-v1), since the accuracy landscape (Fig 8f) is flat over a wide range of solutions: invariant accuracy is achieved as long as $w_i > |w_s|$. As a result, accuracy cannot be used as

a measure of invariance in this problem. Therefore, the fact that MRI-v1 and IRM-v1 exhibit similar test-domain accuracy does not necessarily indicate that they achieve similar degree of invariance.

Limitations A common limitation of both MRI and IRM (Rosenfeld et al., 2020; Ahuja et al., 2021) is that they require significant support overlap across domains. This may limit the applicability of these methods on certain domain generalization benchmarks (Gulrajani and Lopez-Paz, 2020) that consists of domains with different styles or natural backgrounds, like VLCS and Terra-Incognita. Indeed, we find that both methods fail to improve over ERM (in terms of test accuracy) on both VLCS and Terra-Incognita datasets (Table 4).

Another limitation of MRI is that it requires the label distribution ($P_e(y)$) and the invariant feature distribution ($P_e(z_i)$) to remain invariant across domains. In contrast, IRM does not require such assumptions. However, the practical implementation of IRM, IRM-v1, which can behave very differently from IRM (Kamath et al., 2021), has been shown to fail for such problems (Ahuja et al., 2020, 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*.

Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473.

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.

Fang, C., Xu, Y., and Rockmore, D. N. (2013). Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664.

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*.

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

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://github.com/IBM/MRI`
   (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://github.com/IBM/MRI`
   (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

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

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).

B.1 Linear Datasets

For Fig 2, 4, 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 Fig 2, we used the SGD optimizer with no momentum. For Fig 4 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 nonlinear datasets are presented in Table 2-4 and Fig 3, 7, and, 6. The results in Table 2-4 were produced using 10 random seeds.

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 used $\mu = 10$ and $\lambda = 10$ and for optimization using PM we used $\mu = 10000$. 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 Fig 3 and Fig 7, we used 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 the DomainBed (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 $10^{Uniform[-1,1)}$.

	VLCS		Terra-Incognita	
	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$	$\mathcal{L}_{\text{tr}}$	$\mathcal{L}_{\text{test}}$
ERM	0.12 ± 0.0	0.92 ± 0.02	0.41 ± 0.01	1.52 ± 0.12
IRM-v1 (ALM)	0.14 ± 0.01	1.03 ± 0.04	0.45 ± 0.01	$\mathbf{1.30} \pm 0.21$
MRI-v1 (ALM)	0.24 ± 0.01	$\mathbf{0.64} \pm 0.01$	0.46 ± 0.01	$\mathbf{1.30} \pm 0.07$

(a) Risk

	VLCS		Terra-Incognita	
	Train domains	Test domain	Train domains	Test domain
ERM	0.96 ± 0.0	$\mathbf{0.75} \pm 0.01$	0.81 ± 0.01	$\mathbf{0.39} \pm 0.01$
IRM-v1 (ALM)	0.95 ± 0.0	0.74 ± 0.01	0.78 ± 0.01	0.38 ± 0.03
MRI-v1 (ALM)	0.90 ± 0.01	0.71 ± 0.01	0.78 ± 0.01	0.36 ± 0.01

(b) Accuracy

Table 4: Performance of different algorithms on VLCS and Terra-Incognita datasets. (a) Averaged train and test domain risk and (b) accuracy. Mean and standard deviation shown up to 2 decimal places.

CMNIST For the CMNIST datasets, we used 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 Fig 6, we used the same training procedure as above for training except that the seeds and algorithm-specific parameters were chosen at random to generate 50 random hyperparameter combinations for each algorithm. For training with PM, μ was chosen independently and randomly from the distribution $10^{Uniform[2,5)}$. For training with ALM, λ and μ were chosen randomly from the distribution $Uniform[1, 10)$. The total number of gradient steps were set to 5000.

VLCS/Terra-Incognita VLCS (Fang et al., 2013) and Terra-Incognita (Beery et al., 2018) are popular domain generalization datasets that contain realistic images. For the binary classification tasks, we only used two classes from each dataset (VLCS: bird and person, Terra-Incognita: opossum and raccoon). We used the LeNet architecture (LeCun et al., 1998) for both the datasets. We used mini-batch training with a mini-batch size of 256. We trained the models 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. The total number of gradient steps were set to 2000. The weight decay was set to 0.001. No gradient clipping was used. The images were resized to the shape (64,64,3) and normalized with the mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225). For data augmentation, we used Torchvision's RandomHorizontalFlip transform, RandomGrayscale transform, RandomResizedCrop transform with scale = (0.7, 1.0), and ColorJitter transform with brightness, contrast, saturation, and hue set to 0.3.

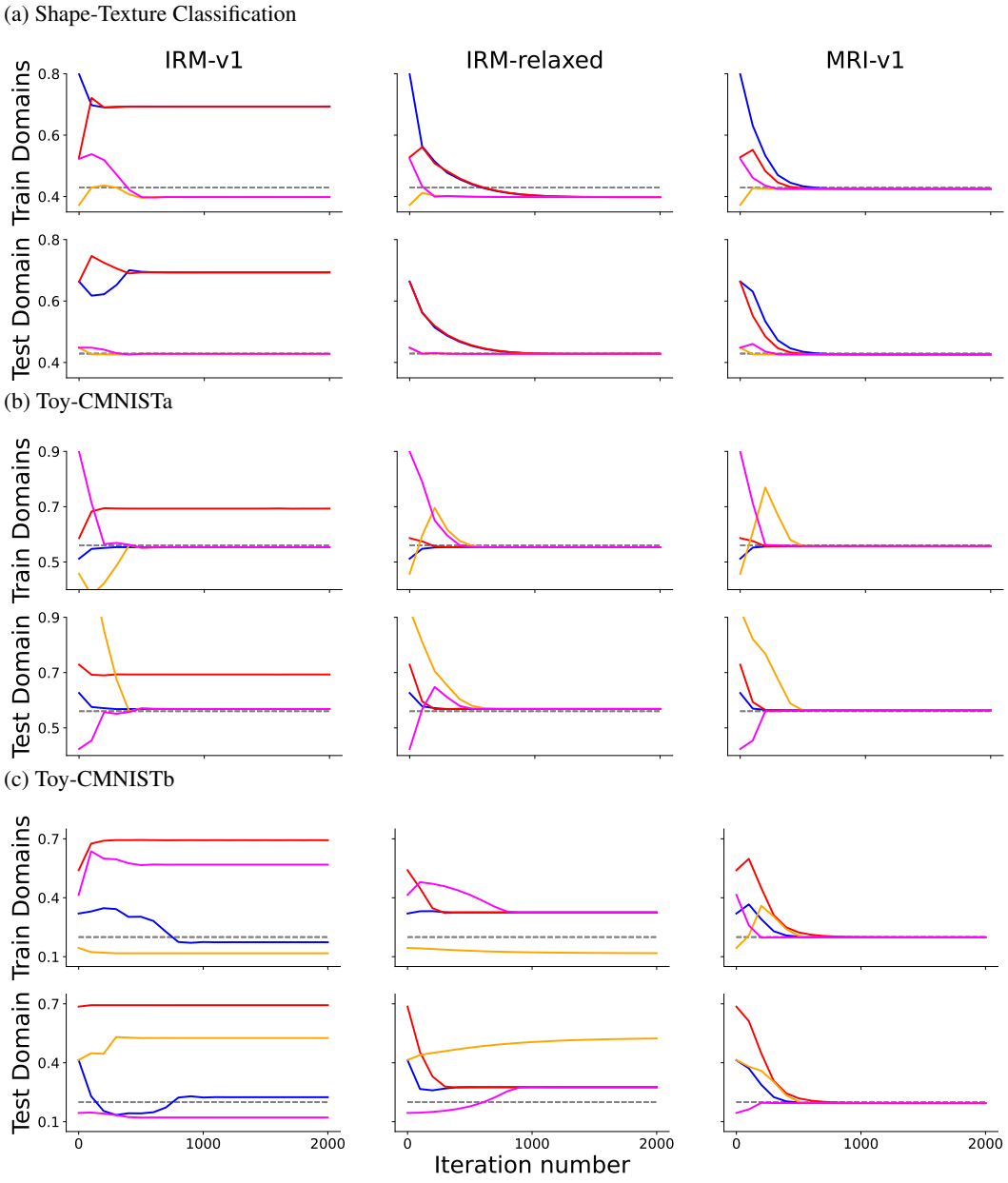
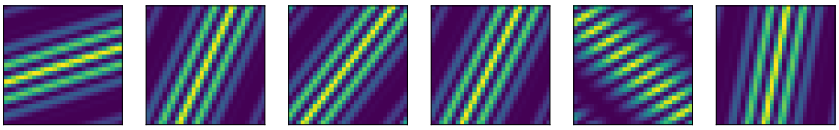
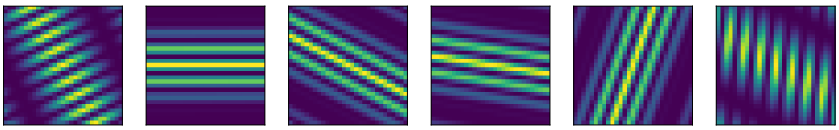


Figure 4: Linear classification tasks. Average train domain risk $\mathcal{L}_{\text{tr}}$ (top) and test domain risk $\mathcal{L}_{\text{test}}$ (bottom) for four different weight initializations on (a) toy Shape-Texture classification dataset, (b) toy-CMNISTa dataset, and (c) toy-CMNISTb dataset (c). Each color corresponds to a different weight initialization. Black dotted line shows the optimal invariant predictor performance.

(a) Train environment e_1: $p_i = 0.75,\ p_s = 1$



(b) Train environment e_2: $p_i = 0.75,\ p_s = 0.8$



(c) Test environment e_0: $p_i = 0.75,\ p_s = 0$

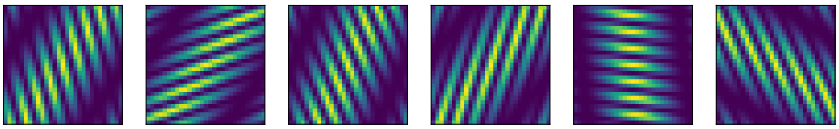
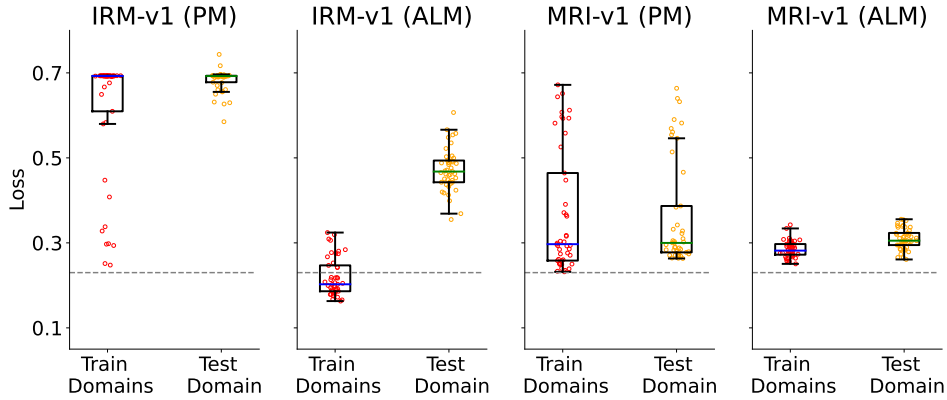


Figure 5: Sample Shape-Texture dataset images.

(a) Risk



(b) Accuracy

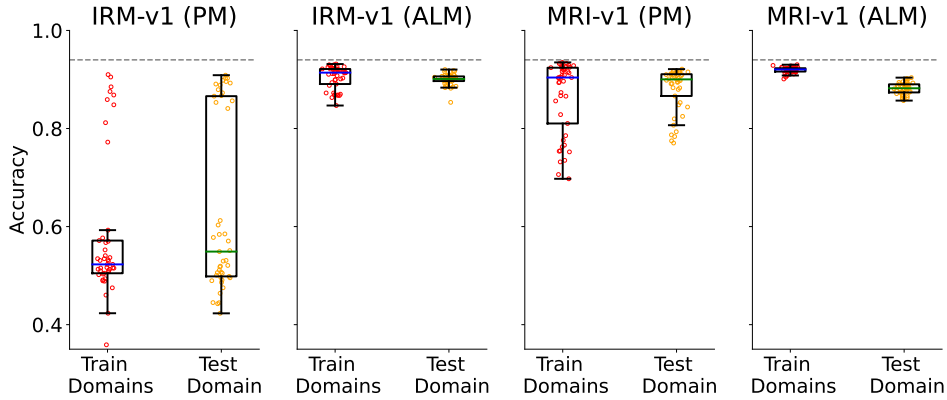


Figure 6: Result on CMNISTb dataset. (a) Averaged train and test domain risk and (b) accuracy using 50 random hyperparameters for IRM-v1 and MRI-v1 trained using both PM and ALM. Black dotted line denotes the Oracle performance.

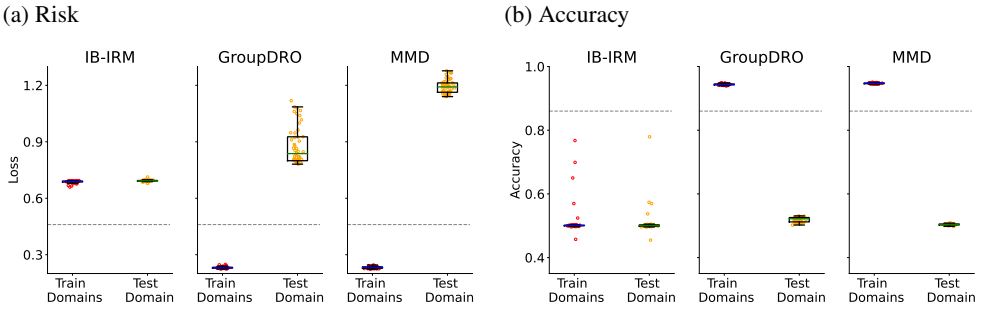


Figure 7: Result on shape-texture classification dataset. (a) Averaged train and test domain risk and (b) accuracy using 50 random hyperparameters for the following algorithms: IB-IRM, GroupDRO and MMD. Black dotted line denotes the Oracle performance.

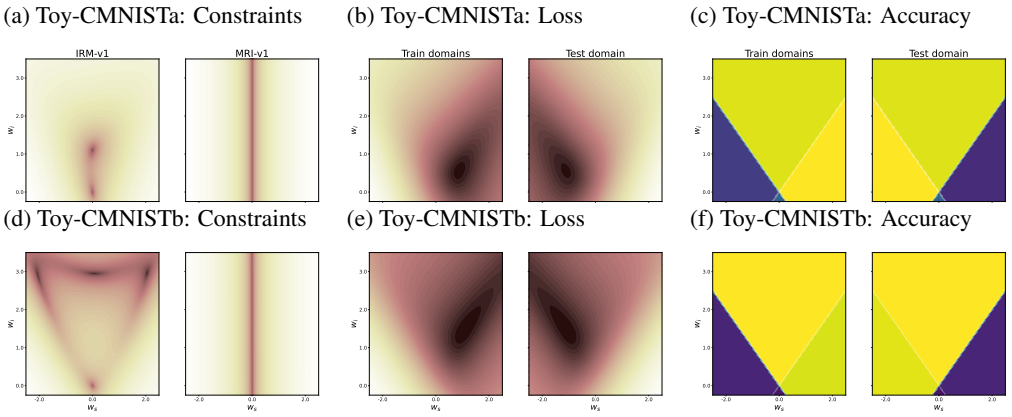


Figure 8: Linear classification of Toy-CMNISTa/b datasets. (a) Invariance constraints for IRM-v1 (left) and MRI-v1(right) on toy-CMNISTa dataset. Y-axis: weight corresponding to the invariant feature w_i. X-axis: weight corresponding to the invariant feature w_s. (b) Averaged risk over train (left) and test (right) domains as a function of the predictor weights on toy-CMNISTa dataset. (c) Averaged accuracy over train (left) and test (right) domains as a function of the predictor weights on toy-CMNISTa dataset. (d)-(f) Same as panels a-c except on toy-CMNISTb dataset.

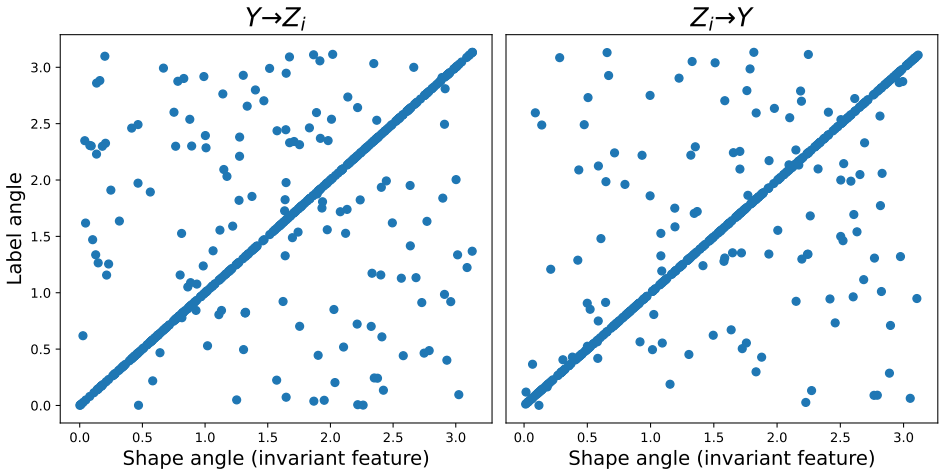


Figure 9: Joint distribution of label angle and shape angle (invariant feature) for the Shape-Texture datasets. This shows that the Shape-Texture dataset generated by either causal directions $Z_i \to Y$ or $Y \to Z_i$ are equivalent.