Supplement for Invariant Risk Minimization Games

Kartik Ahuja [1] **Karthikeyan Shanmugam** [1] **Kush R. Varshney** [1] **Amit Dhurandhar** [1]

1. Examples of hypothesis classes that satisfy affine closure

- **Linear classifiers**: The sum of linear functions (polynomial) leads to a linear function (polynomial), and so does scalar multiplication. Therefore, linear classifiers satisfy affine closure.
- **Reproducing Kernel Hilbert Space** (RKHS): RKHS is a Hilbert space, which is a vector spaces of functions. Therefore, **kernel based classifiers** (Hofmann et al., 2008) satisfy affine closure.
- **Ensemble models:** Consider binary classification and boosting models (Freund et al., 1999). Let $\mathcal{H}_{\mathsf{weak}}$ be the set of weak learners $\omega : \mathcal{X} \to \mathbb{R}$. The final function that is input to a sigmoid is $w = \sum_{m=1}^{k} \theta_m \omega_m$, where each $\theta_m \in \mathbb{R}$. The set of functions spanned by the weak learners is defined as $\mathsf{Span}(\mathcal{H}_{\mathsf{weak}}) = \{\sum_{m=1}^{k} \theta_m \omega_m | \forall m \in \{1,..,k\}, \theta_m \in \mathbb{R}, k \in \mathbb{N}\}$. $\mathsf{Span}(\mathcal{H}_{\mathsf{weak}})$ forms a vector space. Therefore, ensemble models that may use arbitrary number of weak learners satisfy affine closure.
- L^p **spaces**. The set of functions $f : \mathcal{X} \to \mathbb{R}$ for which $\|f\|_p = [\int_{\mathcal{X}} |f(x)|^p dx]^{\frac{1}{p}} < \infty$ is defined as $L^p(\mathcal{X})$. $L^p(\mathcal{X})$ is a vector space (Ash & Doléans-Dade, 2000).

ReLU networks with arbitrary depth: Neural networks are known to be universal function approximators. Let us assume $\mathcal{X}$ to be a compact subset of $\mathbb{R}^n$. The output of a ReLU network is a continuous function on $\mathcal{X}$, which implies it is bounded and thus the function described by a ReLU network is in $L^1(\mathcal{X})$ space. It is clear that the set of functions parametrized by ReLU networks are a subset of functions in $L^1(\mathcal{X})$ space. In the other direction, from Lu et al. (2017), we know that ReLU networks can come arbitrarily close to any function in L^1 sense. Since ReLU networks come arbitrarily close to the function and are not exactly equal we cannot argue that affine closure is satisfied. However, we argue later that since the networks can arbitrarily approximate any function in $L^1(\mathcal{X})$ it is sufficient to prove our results (our main result Theorem 1 and Corollary 1).

[1]IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY. Correspondence to: Kartik Ahuja <kartik.ahuja@ibm.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

2. Theorems and Proofs

In this section, we discuss the proofs to the lemmas, theorems, and corollaries in the main manuscript. We refer to the equations in the main manuscript as M followed by the index of the equation in the main manuscript, for e.g., M2 is equation 2 in the main manuscript.

Theorem 1. *If Assumption 1 holds, then* $\tilde{\mathcal{S}}^{\mathsf{IV}} = \tilde{\mathcal{S}}^{\mathsf{EIRM}}$

Proof. In the first part, we want to show that $\tilde{\mathcal{S}}^{\mathsf{IV}} \subseteq \tilde{\mathcal{S}}^{\mathsf{EIRM}}$. We will use proof by contradiction.

Let us assume that there exists an element $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \in \tilde{\mathcal{S}}^{\mathsf{IV}}$, which does not belong to $\tilde{\mathcal{S}}^{\mathsf{EIRM}}$. This implies that there exists at least one $e \in \mathcal{E}_{tr}$ in the ensemble game, which strictly prefers the action $\bar{w}^e \in \mathcal{H}_w$ to following its current action w^e. In other words, at least one of the inequalities in (M3) is not satisfied, which can be written as

$$R^e\left(\left[\frac{\bar{w}^e + \sum_{q \neq e} w^q}{|\mathcal{E}_{tr}|}\right] \circ \Phi\right) < R^e(w \circ \Phi) \qquad (1)$$

The function $w^{'} = \frac{\bar{w}^e + \sum_{q \neq e} w^q}{|\mathcal{E}_{tr}|} \in \mathcal{H}_w$ (From Assumption 1). Therefore, $w^{'}$ is a strictly better classifier than w with a fixed representation Φ for environment e, which contradicts the condition that $w \in \arg\min_{\bar{w} \in \mathcal{H}_w} R^e(\bar{w} \circ \Phi)$ (which follows from $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \in \tilde{\mathcal{S}}^{\mathsf{IV}}$).

This proves the first part.

In the second part, we want to show that $\tilde{\mathcal{S}}^{\mathsf{EIRM}} \subseteq \tilde{\mathcal{S}}^{\mathsf{IV}}$. Let us assume that there exists an element $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \in \tilde{\mathcal{S}}^{\mathsf{EIRM}}$, which does not belong to $\tilde{\mathcal{S}}^{\mathsf{IV}}$. Following Assumption 1, w lies in $\mathcal{H}_w$. Since $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \notin \tilde{\mathcal{S}}^{\mathsf{IV}}$ there exists at least one $e \in \mathcal{E}_{tr}$ and a classifier $w^{'} \in \mathcal{H}_w$ strictly better than w for a fixed representation Φ. If this were not the case, w will be an invariant predictor w.r.t. Φ across $\mathcal{E}_{tr}$, which would contradict $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \notin \tilde{\mathcal{S}}^{\mathsf{IV}}$. Therefore

$$R^e(w^{'} \circ \Phi) < R^e(w \circ \Phi) \qquad (2)$$

Let us construct a new auxiliary classifier $\tilde{w}^e$ as follows. $\tilde{w}^e = w^{'}|\mathcal{E}_{tr}| - \sum_{q \neq e} w^q$. It follows from Assumption 1 that $\tilde{w}^e \in \mathcal{H}_w$. Observe that the ensemble defined as $\frac{\tilde{w}^e + \sum_{q \neq e} w^q}{|\mathcal{E}_{tr}|}$ simplifies to $w^{'}$. This means that environment e can deviate from w^e to $\tilde{w}^e \in \mathcal{H}_w$ and strictly gain from this deviation. This contradicts the fact that $\{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}$ is a Nash equilibrium ($\{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}$ is a Nash equilibrium because $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \in \tilde{\mathcal{S}}^{\mathsf{EIRM}}$).

□

Corollary 1. *If Assumption 1 holds, then* $\hat{\mathcal{S}}^{\mathsf{IV}} = \hat{\mathcal{S}}^{\mathsf{EIRM}}$

Proof. The proof follows straightaway from Theorem 1. For each $w \circ \Phi \in \hat{\mathcal{S}}^{\mathsf{IV}}$ we look at the corresponding tuple $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \in \tilde{\mathcal{S}}^{\mathsf{IV}}$. From Theorem 1, $(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w) \in \tilde{\mathcal{S}}^{\mathsf{EIRM}}$. Therefore, $w \circ \Phi \in \hat{\mathcal{S}}^{\mathsf{EIRM}}$. The other side follows the same way. □

2.1. Extending Theorem 1 and Corollary 1 to ReLU networks

In the proof of Theorem 1, we used the affine closure property in (1) and (2). However, in (1) and (2), we only need to construct models that can achieve risk arbitrarily close to the models in the LHS of equations (1) and (2). Let $\mathcal{H}_w$ the set of functions of ReLU networks with arbitrary depth defined on compact sets $\mathcal{X}$. These functions are in L^1 class as explained earlier. From (Lu et al., 2017), we can choose ReLU networks from $\mathcal{H}_w$ that approximate the classifiers in the LHS of (1) and (2) arbitrarily. We elaborate on this. Suppose the function to be approximated in the LHS is f. From (Lu et al., 2017), for each $\epsilon > 0$, there exists a ReLU network $\hat{f}$ such that $\mathbb{E}_X[|f - \hat{f}|] \leq \epsilon$. The question is does $\mathbb{E}_X[|f - \hat{f}|] \leq \epsilon$ also ensure that the difference in risks is mitigated $|R^e(f, Y) - R^e(\hat{f}, Y)| \leq \tilde{\epsilon}$. If the loss function ℓ is Lipschitz in the scores (e.g., cross-entropy loss, hinge loss), then if the functions are arbitrarily close the risks will also be arbitrarily close. We show this below.

$$\begin{aligned} &|R^e(f, Y) - R^e(\hat{f}, Y)| \\ &= |\mathbb{E}^e[\ell(f(X), Y) - \ell(\hat{f}(X), Y)]| \\ &\leq \mathbb{E}^e[|\ell(f(X), Y) - \ell(\hat{f}(X), Y)|] \\ &\leq \mathbb{E}^e[L|f(X) - \hat{f}(X)|] \end{aligned} \tag{3}$$

where L is the Lipschitz constant for ℓ.

Below we illustrate an example of Lipschitz continuous loss ℓ. Consider cross entropy for binary classification (labels $Y = 0$ and $Y = 1$). Suppose $f(x) = s$ is the score assigned to class 1, it is converted into probability as $e^s/(1 + e^s)$. The cross-entropy loss is simplified as

$$\ell(s, Y) = Ys - \log(1 + e^s) \tag{4}$$

Observe $\frac{\partial \ell(s,Y)}{\partial s} = Y - \frac{1}{1+e^s}$ and $|\frac{\partial \ell(s,Y)}{\partial s}| \leq 1$. Therefore, $\ell(s, Y)$ is Lispchitz continuous in s.

Theorem 2. *We assume each environment* $e \in \mathcal{E}_{all}$

$$\begin{aligned} &Y^e \leftarrow Z_1^{e\mathsf{T}}\gamma + \epsilon^e, \quad Z_1^e \perp \epsilon^e, \quad \mathbb{E}[\epsilon^e] = 0 \\ &X^e \leftarrow S(Z_1^e, Z_2^e) \end{aligned} \tag{5}$$

Here $\gamma \in \mathbb{R}^c$, $Z_1^e \in \mathbb{R}^c$, $Z_2^e \in \mathbb{R}^q$, $S \in \mathbb{R}^{n \times (c+q)}$. *Assume that* Z_1 *is invertible component of* S, *i.e.,* $\exists\, \tilde{S} \in \mathbb{R}^{c \times n}$ *such that* $\tilde{S}(S(z_1, z_2)) = z_1$ *for all* $z_1 \in \mathbb{R}^c$ *and* $z_2 \in \mathbb{R}^q$. *Let* $\Phi \in \mathbb{R}^{n \times n}$ *have rank* r. *If at least* $n - r + \frac{n}{r}$ *training environments* $\mathcal{E}_{tr} \subseteq \mathcal{E}_{all}$ *lie in linear general position of degree* r, *then any predictor obtained from* EIRM *game over the training environments in* $\hat{\mathcal{S}}^{\mathsf{EIRM}}$ *is invariant across all the testing environments* $\mathcal{E}_{all}$.

Proof. We restate the Theorem 9 from (Arjovsky et al., 2019). In Theorem 2, we just need to replace the last sentence with the following to obtain Theorem 9 from (Arjovsky et al., 2019). If at least $n - r + \frac{n}{r}$ training environments $\mathcal{E}_{tr} \subseteq \mathcal{E}_{all}$ lie in linear general position of degree r, then any predictor in $\hat{\mathcal{S}}^{\mathsf{IV}} = \{w \circ \Phi \mid \Phi\mathbb{E}[X^e X^{e\mathsf{T}}]\Phi^{\mathsf{T}} w = \Phi\mathbb{E}[X^e Y^e], \forall e \in \mathcal{E}_{tr}\}$ is invariant across all the testing environments $\mathcal{E}_{all}$. Since $\mathcal{H}_w = \mathbb{R}^{n \times 1}$ is affine closed, then from Corollary 1 it follows that $\hat{\mathcal{S}}^{\mathsf{EIRM}} = \hat{\mathcal{S}}^{\mathsf{IV}}$. This completes the proof. □

Lemma 1. *If Assumptions 2 and 3 are satisfied, then for any* $w^{'} \in \mathcal{H}_w$ *and* $\Phi \in \mathcal{H}_\Phi$, $w^{'} \circ \Phi^{-1} \in L^p(\mathcal{Z})$.

Proof. To show $w^{'} \circ \Phi^{-1} \in L^p(\mathcal{Z})$ let us first express the integral $\int_{\mathcal{Z}} |w^{'}(\Phi^{-1}(z))|^p dz$ by using substitution rules (Rudin, 1987). We can use the substitution rule because both $\mathcal{X}$ and $\mathcal{Z}$ are n dimensional, the function Φ is bijective, differentiable and Lipshitz continuous (From Asumption 2 and 3). Substitute $z = \Phi(x)$. Then, $\int_{\mathcal{Z}} |w^{'}(\Phi^{-1}(z))|^p dz = \int_{\Phi^{-1}(\mathcal{Z})} |w^{'}(x)|^p \mathsf{det}(J(\Phi(x)))dx$. Here $J(\Phi(x))$ is the Jacobian of the transformation Φ. Since Φ is a Lipschitz continuous map, its determinant is also bounded. We show this as follows.

Lipschitz continuity implies that for any $x, x' \in \mathcal{X}$, $\|\Phi(x) - \Phi(x')\| \leq \gamma\|x - x'\|$ where γ is the Lipschitz constant. In particular, since $\Phi(\cdot)$ is differentiable (Assumption 2), this means that the length of any partial derivative vector $\|\frac{\delta\Phi(x)}{\delta x_i}\| \leq \gamma$ for any coordinate $i \in [n]$. Now, we apply the Hadamard inequality (Garling, 2007) for the determinant of the square matrix $J(\Phi(x))$:

$\mathsf{det}(J(\Phi(x))) \leq \prod_{i\in[n]} \|\frac{\delta\Phi(x)}{\delta x_i}\| \leq \gamma^n$. Therefore,

$$\begin{aligned}\int_{\mathcal{Z}} |w^{'}(\Phi^{-1}(z))|^p dz &= \int_{\Phi^{-1}(\mathcal{Z})} |w^{'}(x)|^p \mathsf{det}(J(\Phi(x)))dx \\ &\leq \gamma^n \int_{\Phi^{-1}(\mathcal{Z})} |w^{'}(x)|^p dx \\ &\leq \gamma^n \int_{\mathcal{X}} |w^{'}(x)|^p dx \end{aligned} \tag{6}$$

Since, $w \in L^p(\mathcal{X})$ (Assumption 3) we have that $w^{'} \circ \Phi^{-1} \in L^p(\mathcal{Z})$ from the above inequality. □

Theorem 3. *If Assumptions 2 and 3 are satisfied and $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ is not empty, then $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}} = \hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I}) = \hat{\mathcal{S}}^{\mathsf{EIRM}}_{\mathcal{X}}(\mathsf{I})$*

Proof. In the first part, we want to show that $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}} \subseteq \hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I})$. We will use proof by contradiction.

Suppose $(w \circ \Phi) \in \bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ but not in $\hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I})$. First note that $w \circ \Phi \in L^p(\mathcal{X})$ (From definition of the set $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$). This implies that there must exist an environment e and a classifier $w^{'} : \mathcal{X} \to \mathcal{Y}$ which is better than $(w \circ \Phi)$. Therefore, we can state that

$$R^e(w^{'}) < R^e(w \circ \Phi) \tag{7}$$

Define a classifier $\tilde{w} = w^{'} \circ \Phi^{-1}$. From Lemma 1 it follows $\tilde{w} \in L^p(\mathcal{Z})$. Define the risk achieved by this classifier as $R^e(\tilde{w} \circ \Phi)$. We simplify this as follows.

$$\begin{aligned} &R^e(\tilde{w} \circ \Phi) = R^e((w^{'} \circ \Phi^{-1}) \circ \Phi) = \\ &R^e(w^{'} \circ (\Phi^{-1} \circ \Phi)) = R^e(w^{'} \circ \mathsf{I}) = R^e(w') \end{aligned} \tag{8}$$

Therefore, the risk of $\tilde{w} \circ \Phi$ is better than the risk achieved by $w \circ \Phi$. This contradicts that $w \circ \Phi$ is an invariant predictor. We show this as follows. Since $w \circ \Phi$ is an invariant predictor with Φ as the representation it implies $w \in \arg\min_{\bar{w}} R^e(\bar{w} \circ \Phi)$. However, $\tilde{w}$ is clearly better than w with Φ as the representation (8) , which leads to a contradiction. This proves the first part.

The second side $\hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I}) \subseteq \bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$. Suppose $w \in \hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I})$ but not in $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$. Select any $\Phi : \mathcal{X} \to \mathcal{Z}$ from the set of representations for which invariant predictors exist in the set $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ (recall that we assumed $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ is not empty). Define a predictor $\tilde{w} = w \circ \Phi^{-1}$. Since $w \in L^p(\mathcal{X})$, from Lemma 1 we know that $\tilde{w}$ is in $L^p(\mathcal{Z})$. There should exist an environment e for which $\tilde{w}$ is not the optimal classifier given Φ otherwise w will be in the set $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$, which would be a contradiction. Φ is a representation for which an invariant predictor exists, let $w^{'}$ be the classifier and $w' \circ \Phi$ be the invariant predictor in $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$. $\exists$ an environment e for which $w^{'}$ is strictly better than $\tilde{w}$ given Φ. We write this condition as

$$R^e(w^{'} \circ \Phi) < R^e(\tilde{w} \circ \Phi) = R^e(w) \tag{9}$$

$w^{'} \circ \Phi \in \bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ and from the definition of the set it follows that $w^{'} \circ \Phi \in L^p(\mathcal{X})$. Also, $w^{'} \circ \Phi$ is better than w from (9). However, w is an invariant predictor with $\Phi = \mathsf{I}$, which leads to contradiction.

From Theorem 1 it follows that $\hat{\mathcal{S}}^{\mathsf{EIRM}}_{\mathcal{X}}(\mathsf{I}) = \hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I})$. This completes the proof. □

Theorem 4. *If Assumption 4 is satisfied, then a pure strategy Nash equilibrium of the game Γ^{EIRM} exists, i.e., $\mathcal{S}^{\mathsf{EIRM}}$ is not empty. Suppose there exists a $\left(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}\right) \in \mathcal{S}^{\mathsf{EIRM}}$ such that $\forall q \in \mathcal{E}_{tr}$ w^q is in the interior of $\mathcal{H}_w$, then the corresponding ensemble predictor $\frac{1}{|\mathcal{E}^{tr}|}\sum_{q=1}^{|\mathcal{E}_{tr}|} w^q \circ \Phi$ is invariant across all the training environments $\mathcal{E}_{tr}$.*

Proof. We will use the classic result from (Debreu, 1952), which shows the sufficient conditions for the existence of pure Nash equilibrium in continuous action games. We provide this result in the next section Theorem 7, where we continue the discussion on concepts in game theory. Informally speaking, the result states that if the game is concave with compact and convex action sets, then the pure Nash equilibrium exists.

The set of actions of each environment $\mathcal{H}_w$ is a closed bounded and convex subset (following the Assumption 4). Recall the definition of the utility of a player e in the EIRM game is given as

$$\begin{aligned} &u_e[w^e, w^{-e}, \Phi] = -R^e(w^{av} \circ \Phi) = \\ &= -\mathbb{E}^e[\ell((w^{av} \circ \Phi)(x), Y)] \end{aligned} \tag{10}$$

Following Assumption 4, we simplify the inner term in the expectation as follows.

$$\ell((w^{av} \circ \Phi)(x), Y) = \ell(\Phi(x)^{\mathsf{T}}[\frac{1}{|\mathcal{E}_{tr}|}\sum_{q=1}^{|\mathcal{E}_{tr}|} w^q], Y) \tag{11}$$

$\ell(\Phi(x)^t w, Y) = h_Y(w)$. $h_Y(w)$ is a convex function of w (From Assumption 4). Define $g : \mathbb{R}^d \times \mathbb{R}^d ... \times \mathbb{R}^d \to \mathbb{R}^d$ as $g(\mathbf{w}^1, ..., \mathbf{w}^{|\mathcal{E}_{tr}|}) = \frac{1}{|\mathcal{E}_{tr}|}\sum_k w^k$. Note that g is an affine mapping. The function in (11) can be expressed as $h_Y(g(w^1, ...w^{|\mathcal{E}_{tr}|}))$. The composition of a convex function with an affine function is also convex (Boyd & Vandenberghe, 2004). We use this to conclude that the composition $h_Y(g(\mathbf{w}^1, ...\mathbf{w}^{|\mathcal{E}_{tr}|}))$ is a convex function in $\mathbf{w}^1, ...\mathbf{w}^{|\mathcal{E}_{tr}|}$. We express (10) in terms of h and g as

$$u_e[w^e, w^{-e}, \Phi] = -\mathbb{E}^e[h_Y(g(w^1, ..., w^{|\mathcal{E}_{tr}|}))] \tag{12}$$

Each term inside the expectation above is concave. Therefore, u_e is concave in w^e (follows directly from Jensen's

inequality applied to u_e). h_Y is a continuous function in $\mathbf{w}$ (from Assumption 4) and g is a continuous function as well, the composition of the two continuous functions is also continuous. As a result, u_e is continuous. Therefore, the EIRM game above satisfies the assumptions in Theorem 5 ((Debreu, 1952), which implies that a pure NE exists. This proves the first part of the theorem. We now discuss the second part of the which provides a simple condition for the existence of invariant predictor.

Say the weights that comprise one of the NE are given as $\{w_*^q\}_{q=1}^{|\mathcal{E}_{tr}|}$. This set of weights satisfy

$$w_*^e = \arg\min_{w^e \in \mathcal{H}_w} -u_e(w^e, w_*^{-e}, \Phi) \tag{13}$$

From the Assumption 4, w_*^e is in the interior of $\mathcal{H}_w$. Therefore, we can construct a ball around it in which it is the smallest point, which implies it is a local minima of $-u_e(w^e, w_*^{-e}, \Phi)$. Since local minima is also the global minima for convex functions; it follows that the solution would be equivalent to searching over the space of all the linear functions, i.e.

$$w_*^e = \arg\min_{w^e \in \mathbb{R}^d} -u_e(w^e, w_*^{-e}, \Phi) \tag{14}$$

The above argument holds for all the environments because each solution w_*^e is in the interior. Therefore, we can transform the EIRM game from the current restricted space $\mathcal{H}_w$ to the space of all the linear functions. The space of the linear functions satisfy affine closure property unlike the space of bounded linear functions $\mathcal{H}_w$. From Theorem 1 it follows that the ensemble classifier $\frac{1}{|\mathcal{E}_{tr}|}\sum_{q=1}^{|\mathcal{E}_{tr}|} w_*^q$ composed with Φ will be an invariant predictor. □

In Theorem 4 we assumed that the model and the representation are both linear functions. We now discuss the existence under a more general class of models.

Assumption 5 $\mathcal{H}_w$ is a family of functions parametrized by $\theta \in \Theta$. We assume that Θ is compact. We assume $w_\theta \in \mathcal{H}_w$, where $w_\theta : \mathbb{R}^d \to \mathbb{R}$ is continuous in its inputs.

Consider a multilayer perceptron (MLP) with say ReLU activation. Each weight in the network belongs $[w_{min}, w_{max}]$. This family of neural networks satisfies the Assumption 5 above.

Suppose that each environment is looking to solve for a probability distribution over the parameters of the neural network written as vector w^e given as p_{w^e}. We rewrite the expected loss of the environments as follows.

$$\bar{u}_e[p_{w^e}, p_{w^{-e}}, p_\Phi] = \mathbb{E}_{\Pi_e p_{w^e} \times p_\Phi}\Big[u_e[w^e, w^{-e}, \Phi]\Big]$$

.

$\bar{u}_e$ is the utility of each environment in the EIRM game when the environment selects p_{w^e}.

Theorem 5. *If Assumption 5 is satisfied, then a mixed strategy Nash equilibrium of* Γ^{EIRM} *is guaranteed to exist.*

Proof. The proof is a direct consequence of the existence result (Glicksberg, 1952), which we restate in Theorem 7. □

The main message of the above theorem is that we relax the requirement of having a deterministic classifier, then we are guaranteed to have a solution for general models as well.

Assumption 6 $\mathcal{X}$ and $\mathcal{Z}$ are finite sets. Let $\mathcal{H}_w$ be the family of all the maps such that $w : \mathcal{Z} \to [-u, u]$, where $0 < u < \infty$.

Under Assumption 6, each classifier w can be understood as a parametrized function with $|\mathcal{X}|$ parameters. Suppose $\mathcal{X} = \{x_1, ..., x_m\}$. In this case, we can think of the function's value at each x_k, $(w \circ \Phi)(x_k) = w_k$, as a parameter w_k chosen from $[-u, u]$. If each of the function values w_k, where $k \in \{1, ..., m\}$, are in $(-u, u)$, then we say w is in the interior of $\mathcal{H}_w$.

Theorem 6. *If Assumption 6 is satisfied, then a pure strategy Nash equilibrium of the game* Γ^{EIRM} *exists, i.e.,* $\mathcal{S}^{\mathsf{EIRM}}$ *is not empty. Suppose there exists a* $\left(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}\right) \in \mathcal{S}^{\mathsf{EIRM}}$ *such that each* $w^q \; \forall q \in \mathcal{E}_{tr}$ *is in the interior of* $\mathcal{H}_w$*, then the corresponding ensemble predictor* $\frac{1}{|\mathcal{E}^{tr}|}\sum_{q=1}^{|\mathcal{E}_{tr}|} w^q \circ \Phi$ *is invariant across all the training environments* $\mathcal{E}_{tr}$*.*

Proof. We follow the same style of proof as Theorem 4.

The set of actions of each environment $\mathcal{H}_w$ is a closed bounded and convex subset (following the Assumption 6). Recall the definition of the utility of a player e in the EIRM game. Let ℓ be square loss function

$$\begin{aligned}
&u_e[w^e, w^{-e}, \Phi] = -R^e(w^{av} \circ \Phi) = \\
&= -\mathbb{E}_{X^e, Y^e}[(Y^e - \frac{1}{|\mathcal{E}_{tr}|}\sum_q w^q(\Phi(x_i)))^2] \\
&= -\mathbb{E}_{X^e, Y^e}[\big(Y^e - \frac{1}{|\mathcal{E}_{tr}|}\sum_q (w_i^q)\big)^2] \\
&= -\mathbb{E}_{Y^e}[\sum_{i=1}^m P(X^e = x_i|Y^e)\big(Y^e - \frac{1}{|\mathcal{E}_{tr}|}\sum_q (w_i^q)\big)^2]
\end{aligned} \tag{15}$$

In the above expression, $w^q(\Phi(x_i)) = w_i^q$. Observe that $\sum_{i=1}^m P(X^e = x_i|Y^e)\big(Y^e - \frac{1}{|\mathcal{E}_{tr}|}\sum_q (w_i^q)\big)^2$ is convex in $\{w_i^q\}_{i \in \{1,..,m\}, q \in \{1,..,|\mathcal{E}_{tr}|\}}$. Therefore, $u_e[w^e, w^{-e}, \Phi]$ is convex in $\{w_i^q\}_{i,q}$. We can repeat the same argument when ℓ is cross-entropy. The existence of NE follows directly from Debreu's result (See Theorem 7). The remaining proof follows steps identical to that in the proof of Theorem 4. □

3. Game Theory Concepts Continued

This section is a continuation to the Section 3.1 on Game Theory Concepts. We discuss some classic results on the existence of NE. Let us now consider continuous action games. We make the following assumption.

Assumption NE 1 For each i:

- S_i is a compact, convex subset of $\mathbb{R}^{n_i}$
- $u_i(s_i, s_{-i})$ is continuous in s_{-i}
- $u_i(s_i, s_{-i})$ is continuous and concave in s_i .

Theorem 7. *(Debreu, 1952) If Assumption NE 1 is satisfied for game Γ, then a pure strategy Nash equilibrium exists.*

We extend the definition of pure strategy NE to mixed strategies (discussion on mixed strategies given in the next section, where we continue the discussion on concepts in game theory), where instead of choosing an action deterministically, each player chooses a probability distribution over the set of actions. We assume that each set S_i is a compact subset of $\mathbb{R}^{n_i}$. Define the set of Lesbegue measures over S_i as $\Delta(S_i)$. Each player i, draws a probability distribution θ_i from $\Delta(S_i)$. The joint strategy played by all the players is the product of their individual distributions written as $\Pi_{k \in N} \theta_k$

Nash equilibrium in mixed strategies. A strategy $\theta^* = \Pi_{k \in N} \theta_k^*$ is said to be a mixed strategy Nash Equilibrium (NE) if it satisfies

$$\mathbb{E}_{\theta^*}\Big[u_i(S_i, S_{-i}^*)\Big] \geq \mathbb{E}_{\theta_{-i}^*}\Big[u_i(k, S_{-i})\Big], \forall k \in S_i, \forall i$$

where $\theta_{-i}^* = \Pi_{k \neq i} \theta_k^*$.

Theorem 8. *(Nash, 1950) Every finite game has a mixed strategy Nash equilibrium.*

Next, we relax some of the above assumptions.

Assumption NE 2 For each i

- S_i is a non empty, compact subset of $\mathbb{R}^{n_i}$
- $u_i(s_i, s_{-i})$ is continuous in s_i and s_{-i}

Theorem 9. *(Glicksberg, 1952) If Assumption NE 2 is satisfied, then the game has a mixed strategy Nash equilibrium.*

4. Deriving the expression for backpropagation

For instance x, the predicted score from Environment 1,2 (Model 1,2) for class k is given as $w_1^k \circ x$, $w_2^k \circ x$ respectively, where w_j^k is the score output by neural network j for class k. The overall score is given as $w_1^k \circ x + w_2^k \circ x$. We take the softmax to get the overall probability for class k as

$$p_k = \frac{\exp\Big[w_1^k \circ x + w_2^k \circ x\Big]}{\sum_j \exp\Big[w_1^j \circ x + w_2^j \circ x\Big]} \tag{16}$$

The softmax vector is $p = [p_0, p_1]$. Denote $w_j^k \circ x = s_j^k$. The log-likelihood for instance x with label y is given as

$$\begin{aligned}
&\log[p_y] \\
&= w_1^y \circ x + w_2^y \circ x - \log\Big(\sum_j \exp\Big[w_1^j \circ x + w_2^j \circ x\Big]\Big) \\
&= s_1^y + s_2^y - \log\Big(\sum_j \exp\Big[s_1^j + s_2^j\Big]\Big)
\end{aligned} \tag{17}$$

The gradient of log-likelihood w.r.t score of each model is given as

$$\begin{aligned}
\frac{\partial \log[p_y]}{\partial s_j^k} &= I(k = y) - \frac{\exp\Big[s_1^k + s_2^k\Big]}{\sum_j \exp\Big[s_1^j + s_2^j\Big]} \\
&= I(k = y) - p_k
\end{aligned} \tag{18}$$

We convert y into a one hot encoded vector $\bar{y}$ and simplify the above expression as

$$\frac{\partial \log[p_u]}{\partial s_j} = \bar{y} - p = \tilde{e} \tag{19}$$

5. Computing Environment

The experiments were done on 2.3 GHZ Intel Core i9 processor with 32 GB memory (2400 MHz DDR4).

6. Description of the Datasets

6.1. Colored MNIST Digits

We use the exact same environment as in Arjovsky et al. (2019). Arjovsky et al. (2019) propose to create an environment for training to classify digits in MNIST digits data [1], where the images in MNIST are now colored in such a way that the colors spuriously correlate with the labels. The task is to classify whether the digit is less than 5 (not including 5) or more than 5. There are three environments (two training containing 30,000 points each, one test containing 10,000 points) We add noise to the preliminary label ($\tilde{y} = 0$ if digit is between 0-4 and $\tilde{y} = 1$ if the digit is between 5-9) by flipping it with 25 percent probability to construct the final

[1] https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist/load_data

labels. We sample the color id z by flipping the final labels with probability p_e, where p_e is 0.2 in the first environment, 0.1 in the second environment, and 0.9 in the third environment. The third environment is the testing environment. We color the digit red if $z = 1$ or green if $z = 0$.

6.2. Colored Fashion MNIST

We modify the fashion MNIST dataset [2] in a manner similar to the MNIST digits dataset. Fashion MNIST data has images from different categories: "t-shirt", "trouser", "pullover", "dress", "coat", "sandal", "shirt", "sneaker", "bag", "ankle boots". We add colors to the images in such a way that the colors correlate with the labels. The task is to classify whether the image is that of foot wear or a clothing item. There are three environments (two training, one test) We add noise to the preliminary label ($\tilde{y} = 0$: "t-shirt", "trouser", "pullover", "dress", "coat", "shirt" and $\tilde{y} = 1$: "sandle", "sneaker", "ankle boots") by flipping it with 25 percent probability to construct the final label. We sample the color id z by flipping the noisy label with probability p_e, where p_e is 0.2 in the first environment, 0.1 in the second environment, and 0.9 in the third environment, which is the test environment. We color the object red if $z = 1$ or green if $z = 0$.

6.3. Colored Desprites Dataset

We modify the Desprites dataset [3] in a manner similar to the MNIST digits dataset. The task is to classify if the image is a circle or a square. We take the preliminary binary labels $\tilde{y} = 0$ for a circle and $\tilde{y} = 1$ for a square. We add noise to the preliminary label by flipping it with 25 percent probability to construct the final label. We sample the color id z by flipping the noisy label with probability p_e, where p_e is 0.2 in the first environment, 0.1 in the second environment, and 0.9 in the third environment, which is the test environment. We color the object red if $z = 1$ or green if $z = 0$.

6.4. Structured Noise in Fashion MNIST

In the previous three experiments, we used color in the images to create correlations. In this experiment, we use a different mechanism to create correlations in Fashion MNIST dataset. We add a small square ($3\times$ 3), in the top left corner of some images and an even smalller square (2×2) in the bottom right corner of other images. The location of the box is correlated with labels. The preliminary labels are the same as in the other experiment with Fashion MNIST. There are three environments (two training, one test). We add noise to the preliminary label by flipping it with 25 percent probability to construct the final label. We sample the location id z by flipping the noisy label with probability p_e, where p_e is 0.2 in the first environment, 0.1 in the second environment, and 0.9 in the third environment, which is the test environment. We place the square in the top left if $z = 1$ or bottom right if $z = 0$.

[2] https://www.tensorflow.org/api_docs/python/tf/keras/datasets/fashion_mnist/load_data

[3] https://github.com/deepmind/dsprites-dataset

7. Architecture, Hyperparameter and Training Details

Architecture for 2 player EIRM game with fixed Φ

In the game with fixed Φ, we used the following architecture for the two models. The model used is a simple multilayer perceptron with following parameters.

- Input layer: Input batch $(\mathsf{batch}, \mathsf{len}, \mathsf{wid}, \mathsf{depth}) \rightarrow$ Flatten
- Layer 1: Fully connected layer, output size = 390, activation = ELU, L2-regularizer = 1.25e-3, Dropout = 0.75
- Layer 2: Fully connected layer, output size = 390, activation = ELU, L2-regularizer = 1.25e-3, Dropout = 0.75
- Output layer: Fully connected layer, output size = 2

We use the above architecture across all the experiments. The shape of the input in the above architecture depends on the dimensions of the data that are input.

Architecture for 2 player EIRM game with variable Φ

In the game with variable Φ, we used the following architecture.

The architecture for the representation learner is

- Input layer: Input batch $(\mathsf{batch}, \mathsf{len}, \mathsf{wid}, \mathsf{depth}) \rightarrow$ Flatten
- Layer 1: Fully connected layer, output size = 390, activation = ELU, L2-regularizer = 1.25e-3, Dropout = 0.75
- Output layer: Fully connected layer, output size = 390, activation = ELU, L2-regularizer = 1.25e-3, Dropout = 0.75

The output from the representation learner above is fed into two MLPs one for each environment (we use the same architecture for both environments).

- Layer 1: Fully connected layer, output size = 390, activation = ELU, L2-regularizer = 1.25e-3, Dropout = 0.75
- Layer 2: Fully connected layer, output size = 390, activation = ELU, L2-regularizer = 1.25e-3, Dropout = 0.75
- Output layer: Fully connected layer, output size = 2

We use the above architecture across all the experiments. The shape of the input in the above architecture depends on the dimensions of the data that are input.

Optimizer and other hyperparameters We used Adam optimizer for training with learning rate set to 2.5e-4. We optimize the cross-entropy loss function. We set the batch size to 256. We terminate the algorithm according to the rules we explained in the Experiments Section in the main manuscript. Thus the number of training steps can vary across different trials. There is a warm start phase for all the methods; we set the warm start phase to be equal to the number of steps in one epoch, where one epoch is the (training data size/ batch size). For the setup with fixed Φ, we set the period to be 2, i.e. in one step first model trains and in the other step the second model trains and this cycle repeats throughout the training. For the setup with variable Φ, we let the two environments and representation learner take turns to update their respective models, environment 1 trains in one step, environment 2 trains in the next step, representation learner trains, and this cycle continues.

Architecture for IRM (Arjovsky et al., 2019)

We used the same architecture that they described in the github repository. [4]. We describe their architecture below.

- Input layer: Input batch $(\text{batch}, \text{len}, \text{wid}, \text{depth}) \rightarrow$ Flatten
- Fully connected layer, output size = 390, activation = ReLU, L2-regularizer = 1.1e-3
- Fully connected layer, output size = 390, activation = ReLU, L2-regularizer = 1.1e-3
- Output layer: Fully connected layer, output size= 2

Optimizer, hyperparameters and some remarks We used Adam optimizer for training with learning rate set to 4.89e-4. We optimize the cross-entropy loss function. We set the batch size to 256. The total number of steps is set to 500. The penalty weight is set to 91257. The penalty term is only used after 190 steps. The code in (Arjovsky et al., 2019) uses a normalization trick to the loss to avoid gradient explosion. We did not find this strategy to be always helpful. Therefore, we carried out experiments for both the cases (with and without normalization of loss) and report the case for which the accuracy is higher.

8. Figures Continued

In this section, we provide the figures for all the datasets and for both V-IRM and F-IRM game since the figures in the manuscript were only provided for one dataset and type of game. The plots in Figure 1,3,4, are the same as in the main manuscript. In Figure 1,3,4, we let each model in its turn use ltr (ltr=5) SGD step updates before the turn of the next model. In all our experiments we set ltr=1, we show the figure with ltr=5 to visually illustrate the oscillations better. In Figure 5-36, we show the plots for the setting with ltr=1 (corresponding to the experiments in Tables 1-4 in the manuscript). The captions under the plot describe the dataset and the game (F-IRM/V-IRM). All the plots in Figure 1-36 use the termination criteria we described. Across all the figures we observe the same trend that we observed and explained in the Experiments Section in the main manuscript.

To illustrate what happens if we let the training go on, we in Figure 36-40 let the training for V-IRM on Desprites dataset continue for many more training steps. Figures 36-40 illustrate that the oscillations are stable and persist. As a result, we continue to encounter the state in which the ensemble model does not exploit spurious correlations.

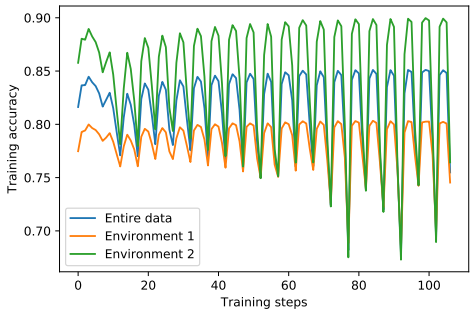

Figure 1. F-IRM, Colored Fashion MNIST: Comparing accuracy of ensemble (ltr=5)

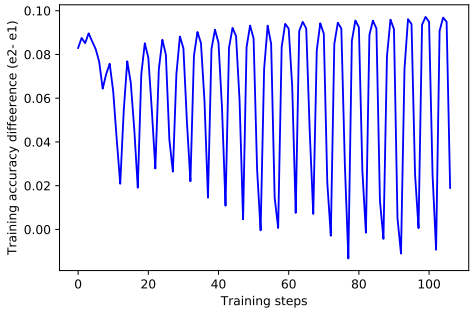

Figure 2. F-IRM, Colored Fashion MNIST: Difference in accuracy of the ensemble model between the two environments (ltr=5)

[4] https://github.com/facebookresearch/InvariantRiskMinimization

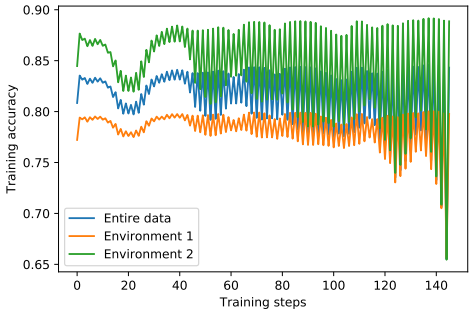

Figure 5. F-IRM, Colored Fashion MNIST: Comparing accuracy of ensemble

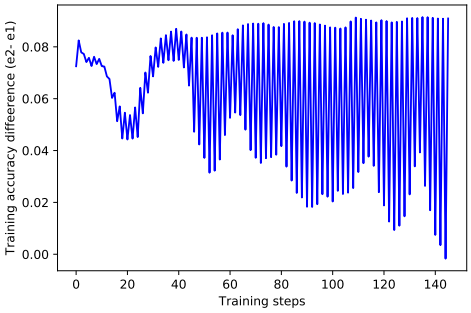

Figure 6. F-IRM, Colored Fashion MNIST: Difference in accuracy of the ensemble model between the two environments

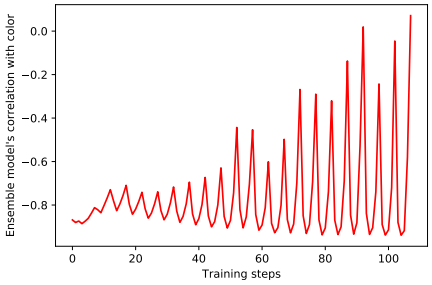

Figure 3. F-IRM, Colored Fashion MNIST: Ensemble's correlation with color (ltr = 5)

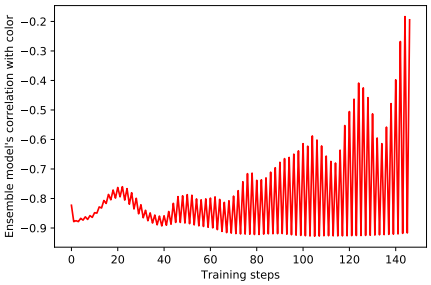

Figure 7. F-IRM, Colored Fashion MNIST: Ensemble's correlation with color

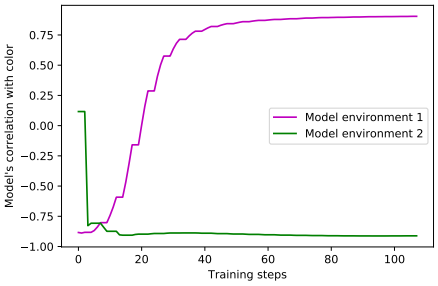

Figure 4. F-IRM, Colored Fashion MNIST: Compare individual model correlations (ltr=5)

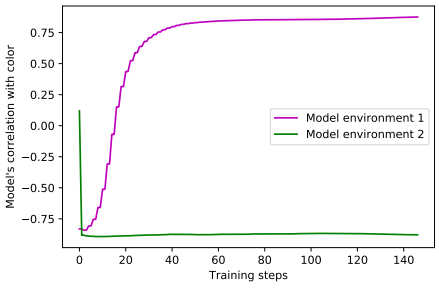

Figure 8. F-IRM, Colored Fashion MNIST: Compare individual model correlations

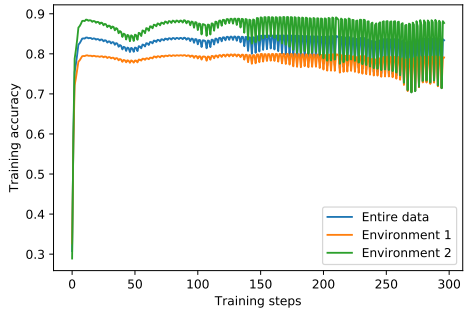

Figure 9. V-IRM Colored Fashion MNIST: Comparing accuracy of ensemble

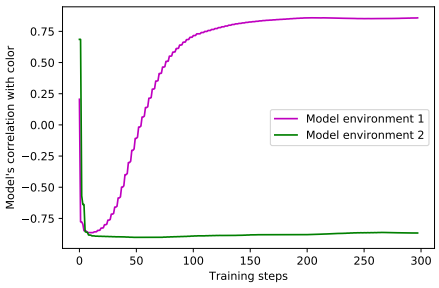

Figure 12. V-IRM Colored Fashion MNIST: Compare individual model correlations.

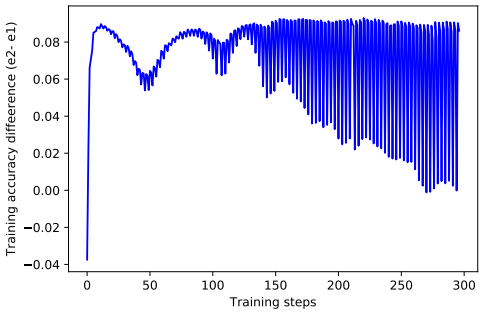

Figure 10. V-IRM Colored Fashion MNIST: Difference in accuracy of the ensemble model between the two environments

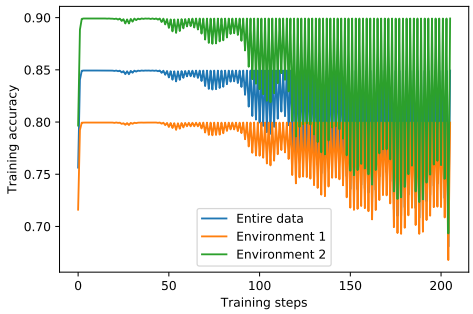

Figure 13. F-IRM Colored Digits MNIST: Comparing accuracy of ensemble

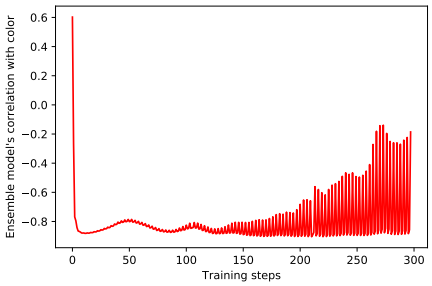

Figure 11. V-IRM Colored Fashion MNIST: Ensemble's correlation with color

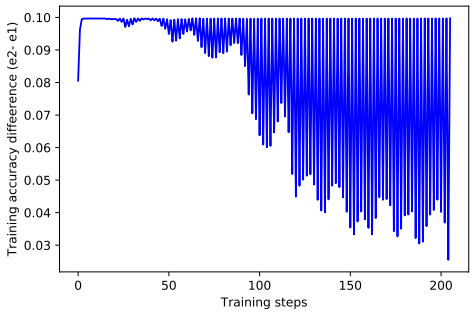

Figure 14. F-IRM Colored Digits MNIST: Difference in accuracy of the ensemble model between the two environments

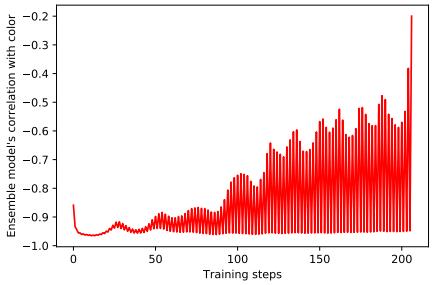

Figure 15. F-IRM Colored Digits MNIST: Ensemble's correlation with color

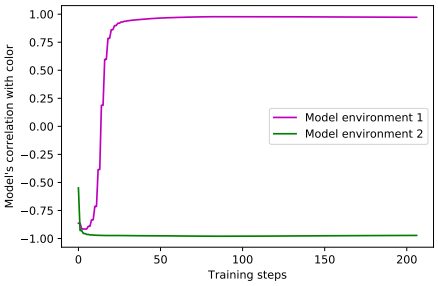

Figure 16. F-IRM Colored Digits MNIST: Compare individual model correlations.

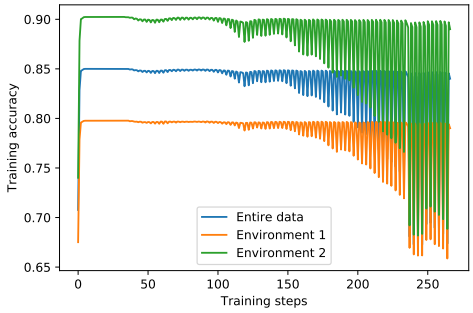

Figure 17. V-IRM Colored Digits MNIST: Comparing accuracy of ensemble

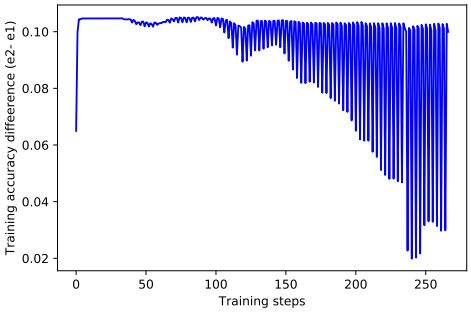

Figure 18. V-IRM Colored Digits MNIST: Difference in accuracy of the ensemble model between the two environments

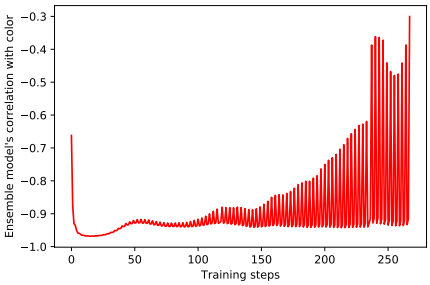

Figure 19. V-IRM Colored Digits MNIST: Ensemble's correlation with color

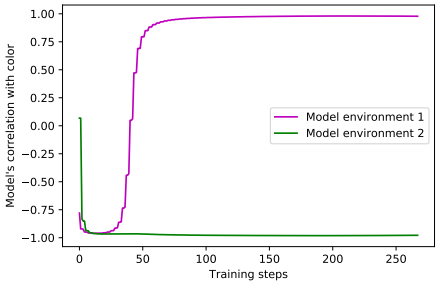

Figure 20. V-IRM Colored Digits MNIST: Compare individual model correlations

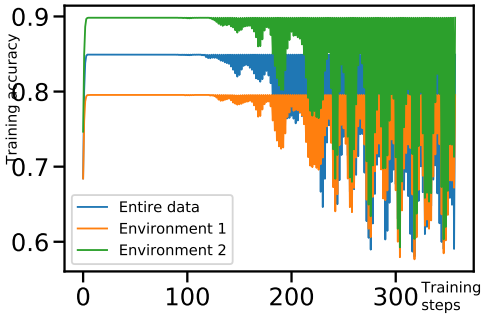

Figure 21. F-IRM Colored Desprites: Comparing accuracy of ensemble

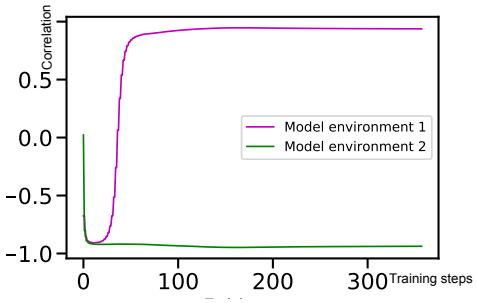

Figure 24. F-IRM Colored Desprites: Compare individual model correlations

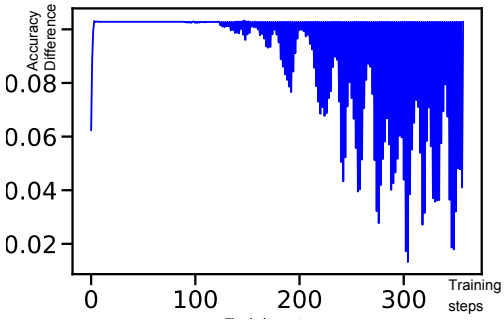

Figure 22. F-IRM Colored Desprites: Difference in accuracy of the ensemble model between the two environments

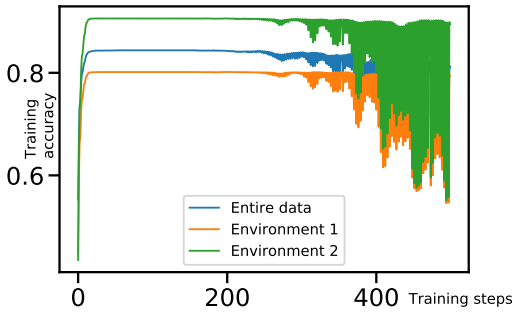

Figure 25. V-IRM Colored Desprites: Comparing accuracy of ensemble

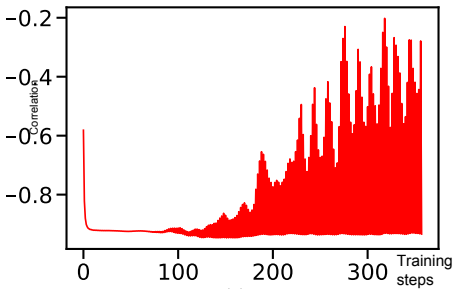

Figure 23. F-IRM Colored Desprites: Ensemble's correlation with color

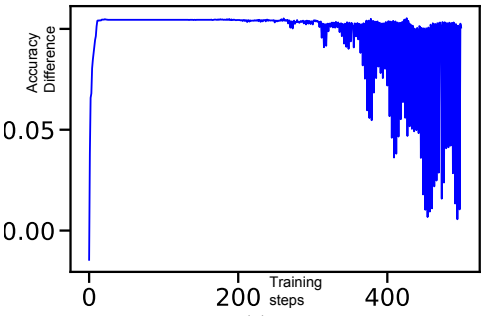

Figure 26. V-IRM Colored Desprites: Difference in accuracy of the ensemble model between the two environments

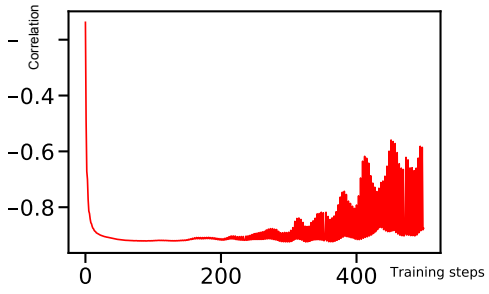

Figure 27. V-IRM Colored Desprites: Correlation of the ensemble model with color

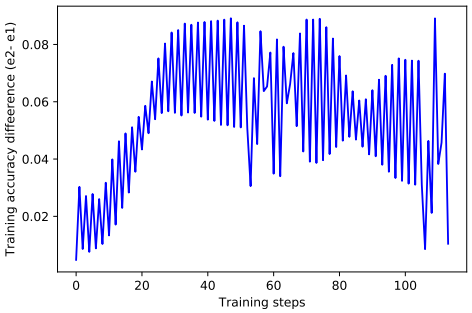

Figure 30. F-IRM Structured Noise Fashion MNIST: Difference in accuracy of the ensemble model between the two environments

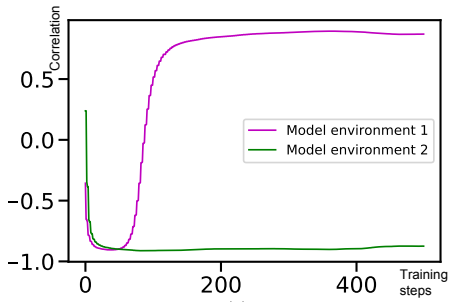

Figure 28. V-IRM Colored Desprites: Compare individual model correlations

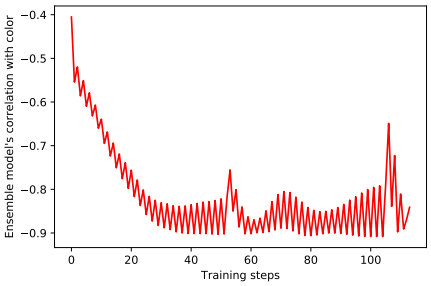

Figure 31. F-IRM Structured Noise Fashion MNIST: Correlation of the ensemble model with color

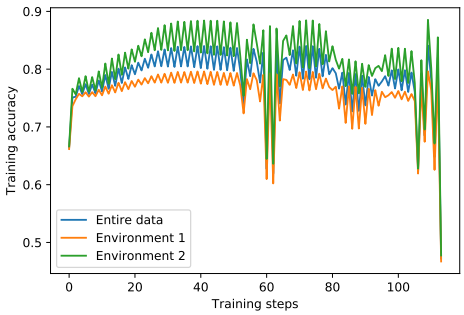

Figure 29. F-IRM Structured Noise Fashion MNIST: Comparing accuracy of ensemble

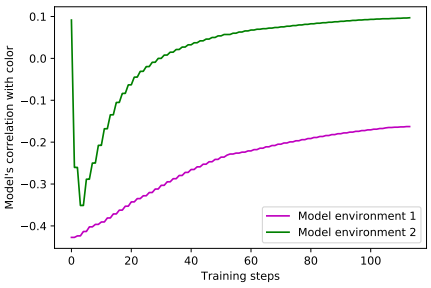

Figure 32. F-IRM Structured Noise Fashion MNIST: Individual model correlation with color

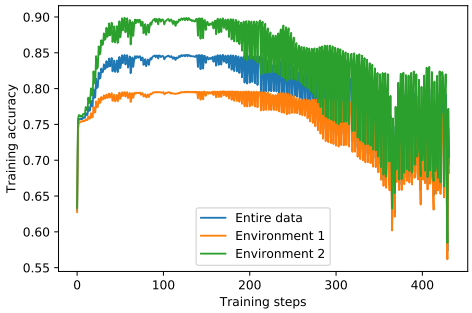

Figure 33. V-IRM Structured Noise Fashion MNIST: Comparing accuracy of ensemble

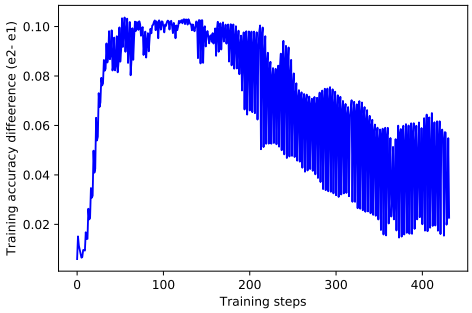

Figure 34. V-IRM Structured Noise Fashion MNIST: Difference in accuracy of the ensemble model between the two environments,

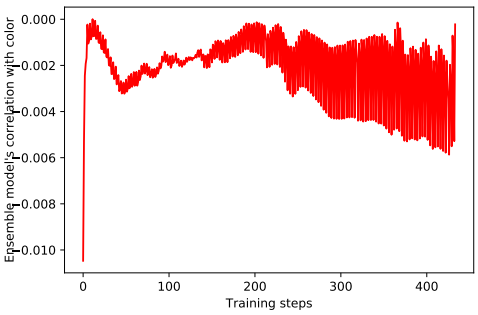

Figure 35. V-IRM Structured Noise Fashion MNIST: Ensemble's correlation with color

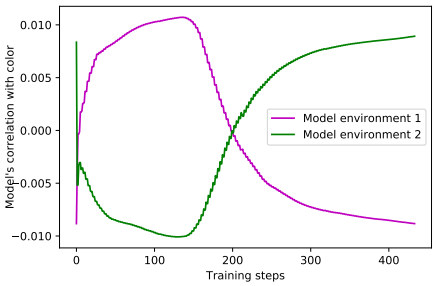

Figure 36. V-IRM Structured Noise Fashion MNIST: Individual model correlation with color

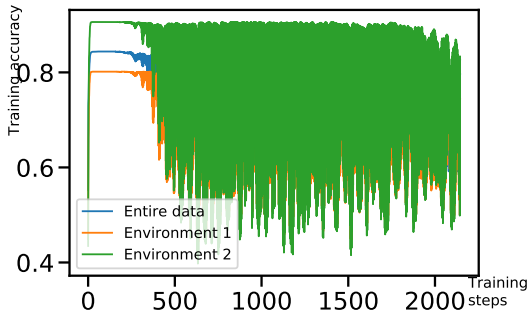

Figure 37. V-IRM Colored Desprites: Comparing accuracy of ensemble (More train steps)

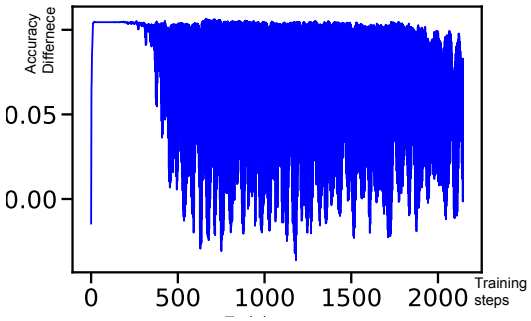

Figure 38. V-IRM Colored Desprites: Difference in accuracy of the ensemble model between the two environments (More train steps)

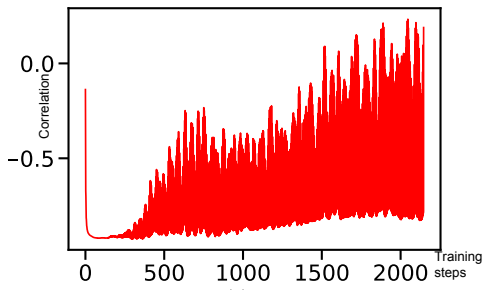

Figure 39. V-IRM Colored Desprites: Ensemble's correlation with color (More train steps)

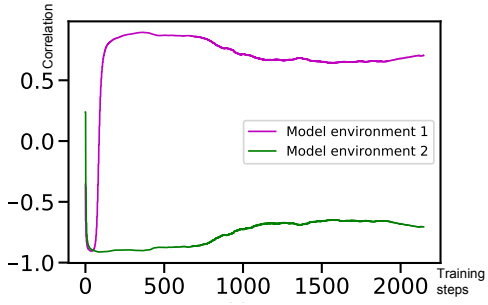

Figure 40. V-IRM Colored Desprites: Individual model correlation with color (More train steps)

References

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

Ash, R. B. and Doléans-Dade, C. A. *Probability and Measure Theory*. Academic Press, San Diego, California, 2000.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Debreu, G. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38(10): 886–893, 1952.

Freund, Y., Schapire, R., and Abe, N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Garling, D. J. *Inequalities: a journey into linear analysis*. Cambridge University Press, 2007.

Glicksberg, I. L. A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.

Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, pp. 6231–6239, 2017.

Nash, J. F. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

Rudin, W. Real and complex analysis (mcgraw-hill international editions: Mathematics series). 1987.