Clustered Invariant Risk Minimization

Tomoya Murata[*] **Atsushi Nitanda** [†‡] **Taiji Suzuki**[§¶]
murata@msi.co.jp atsushi_nitanda@cfar.a-star.edu.sg taiji@mist.i.u-tokyo.ac.jp

[*]NTTDATA Mathematical Systems Inc., Japan
[†]CFAR and IHPC, Agency for Science, Technology and Research (A*STAR), Singapore
[‡]College of Computing and Data Science, Nanyang Technological University (NTU), Singapore
[§]The University of Tokyo, Japan
[¶]Center for Advanced Intelligence Project, RIKEN, Japan

Abstract

This study extends the problem settings of Invariant Risk Minimization(IRM) for Out-of-Distribution generalization problems to unknown clustered environments settings. In this scenario, where a given set of environments exhibits an unknown clustered structure, our objective is to identify a single invariant feature extractor and per-cluster regressors (or classifiers) built on top of the feature extractor. To achieve this, we propose a new framework called Clustered IRM for simultaneously identifying the cluster structure and the invariant features. Our theoretical analysis demonstrates that the required number of training environments for such identification is only $O(d_{\rm sp}+K^2)$, where $d_{\rm sp}$ represents the dimensionality of the spurious features, and K is the number of clusters. Numerical experiments validate the effectiveness of our proposed framework.

1 Introduction

In machine learning, achieving robust performance beyond training data distribution is a fundamental problem. In particular, Out-of-Distribution (OoD) generalization problems focus on the model's ability to make accurate predictions when confronted with a data distribution different from the training distribution (Blanchard et al., 2011; Liu et al., 2021). OoD generalization holds paramount significance in real-world applications where the deployment environment may differ significantly from the training conditions, thereby necessitating models to exhibit adaptability and resilience. However, the traditional Empirical Risk Minimization (ERM) approach often struggles with OoD samples, as it tends to overfit to the training distribution, particularly to so-called *spurious features* of the data samples, such as image backgrounds, resulting in poor generalization on unseen data. This highlights the urgent need to develop robust methodologies and algorithms that can help models generalize effectively in diverse and dynamic environments.

There are many approaches to tackle OoD problems. Confidence-based methods leverage measures of uncertainty, such as predictive entropy or confidence scores, to identify OoD samples (Hendrycks and Gimpel, 2016; Lee et al., 2017; DeVries and Taylor, 2018; Liang et al., 2018; Liu et al., 2021; Huang et al., 2021; Li et al., 2022). By detecting instances where the model's confidence is low, these methods can flag potential OoD samples for further scrutiny or employ specialized handling mechanisms. Data augmentation techniques manipulate training data (or features) to encompass a broader range of variations and scenarios by using random transformations, synthetic data generation, or domain-specific perturbations, thereby enhancing the model's ability to generalize to unseen instances (Volpi et al., 2018; Li et al., 2021; Yao et al., 2022). Distribution matching approaches aim to align the distributions of the training and testing data, either explicitly through domain adaptation techniques or implicitly by learning domain-invariant representations (Long et al., 2013, 2015; Ganin et al., 2016; Motiian et al., 2017; Li et al., 2018a,b; Niu et al., 2023). By reducing distributional disparities between domains, these methods mitigate the impact of domain shift on model performance.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

Recently, Invariant Risk Minimization (IRM), which has its origins in the invariant causal prediction theory of causal inference with structural equation modeling (SEM) (Peters et al., 2016), has been proposed for OoD generalization problems, which offers a principled framework for dealing with distribution shift by explicitly modeling and minimizing the effects of distributional differences between domains or environments (Arjovsky et al., 2019). At its core, IRM aims to capture *invariant features* on which the optimal regressors or classifiers remain consistent across environments. Prioritizing the discovery of features that remain consistent across diverse environments helps create models that exhibit robustness to domain shift, thereby enhancing their utility and reliability in real-world deployment scenarios. IRM has gained significant attention due to its fundamental importance, and many of its variants have been developed (Ahuja et al., 2020; Krueger et al., 2021; Ahuja et al., 2021a; Zhou et al., 2022; Chen et al., 2022; Lin et al., 2022a). Additionally, several studies have extended the problem settings of IRM to broaden the applicability of the IRM principle (Creager et al., 2021; Lin et al., 2022b; Yong et al., 2024).

While traditional IRM assumes the existence of a single true regressor (or classifier), β, and invariant feature extractor, Φ, in the data generation processes among environments, we can consider cases where the true regressor depends on the environment, i.e., $\beta_e \neq \beta_{e'}$ for $e \neq e'$. In such scenarios, without additional assumptions, capturing invariant features using IRM becomes impossible because a single regressor cannot remain consistent across multiple environments. Our focus lies in scenarios where the given set of environments exhibits an unknown clustered structure, with each cluster sharing a single true feature extractor common to all clusters and having a single regressor within each cluster (Figure 2.1). Then, we aim to identify a single invariant feature extractor Φ and per-cluster regressors $\{\beta_k\}_{k\in[K]}$ on top of the feature extractor, where K is the number of clusters. For instance, in federated learning (Konecnỳ et al., 2016; McMahan et al., 2017), where multiple clients (representing different environments) collaborate to learn machine learning models without directly sharing their local datasets, various types of heterogeneity among the local datasets are quite common (Hsieh et al., 2020). This means that the data distributions across clients are not identical, resulting in varying true regressors or classifiers among them. To illustrate these situations, let's consider the scenario of learning recommendation systems from distributed user preference data, where each user corresponds to an environment. Since individual users have distinct preferences even when the input data is identical, true regressors can vary across users. To address these challenges, assuming a clustered structure among environments proves beneficial in realistic recommendation tasks (Sarwar et al., 2002; Li and Kim, 2003). The clustered assumption has been frequently employed in the literature on clustered multi-task learning (Jacob et al., 2008) clustered federated learning (Sattler et al., 2020; Ghosh et al., 2020; Zeng et al., 2023).

To the best of our knowledge, there are currently no frameworks designed to accommodate a clustered structure among environments in IRM settings. Simply applying IRM in such scenarios would prove inadequate because the consistency of the regressor across environments does not hold in general. Furthermore, since the cluster structure is typically unknown, independently applying IRM to each cluster is not feasible. Even if the cluster structure were known, employing independent IRM would necessitate $O(Kd_{\text{sp}})$ environments to capture the invariant features, which is K times greater than that of standard IRM's environments $O(d_{\text{sp}})$, where d_{sp} represents the dimensionality of the spurious features.

Main Contributions

This work proposes an invariant risk minimization framework for environments with an unknown clustered structure, that identifies both the clustering structure and the invariant features *simultaneously*. Theoretically, we demonstrate that our framework can capture clustered linear invariant predictors, where the shared feature extractor does not rely on any spurious features, and the obtained predictors actually match the true ones, within $O(d_{\text{sp}} + K^2)$ environments. Here, d_{sp} represents the dimensionality of the spurious features, and K is the number of clusters. When $d_{\text{sp}} \gg K$[1], this complexity can be significantly smaller than $O(Kd_{\text{sp}})$ of the naive application of IRM per cluster with the knowledge of the true cluster structure. Furthermore, when faced with new environments, the true predictors can be recovered by selecting the best-fit predictor from the trained per-cluster predictors based on additional samples from these new environments. Although the original proposed problem includes a combinatorial clustering constraint, we provide a practical implementation based on soft clustering with Gumbel-softmax to efficiently solve the proposed optimization problem.

Related Work

Several studies have explored the necessary number of training environments to capture invariant features. Arjovsky et al. (2019) have obtained an upper bound of $O(d_{\text{sp}})$ for IRM in regression problems under their linear general position assumption. Rosenfeld et al.

[1]For example, Zhou et al. (2022) have discussed the possibility of extremely large d_{sp}.

(2020) have shown that the complexity of $O(d_{\mathrm{sp}})$ for IRM cannot be improved for binary classification tasks with Gaussian hidden variables. Wang et al. (2022) have proposed a new approach called ISR-Cov based on Principal Component Analysis (PCA). Notably, they demonstrated that ISR-Cov requires only $O(1)$ training environments to capture invariant features. However, this approach relies on the assumption that the distribution of the invariant features is completely consistent across all environments, which is not assumed in Arjovsky et al. (2019) or this study. Although many theoretical studies on IRM, including this one, mainly focus on infinite sample settings, Ahuja et al. (2021b) have discussed the sample complexity of IRM for several distribution shift settings. Rosenfeld et al. (2020) have also discussed the limitations of IRM. Specifically, they theoretically demonstrated that IRM might fail to capture invariant features when the true feature extractor is non-linear. From a different perspective, Zhou et al. (2022) have pointed out an overfitting problem in IRM for finite sample settings and proposed using a sparse constraint on the feature space to avoid overfitting. Lu et al. (2021) have considered extracting non-linear invariant features based on a causal approach called iCaRL, assuming that the conditional distribution of the invariant features given the response variable belongs to a general exponential family of distributions. Applications of IRM to federated learning (Gupta et al., 2022) and continual learning (Alesiani et al., 2023) have also been proposed.

2 Notation and Problem Settings

Notation. For vector $x \in \mathbb{R}^d$, x_i denotes the i-th element of x. $\|\cdot\|$ denotes the Euclidean L_2 norm $\|\cdot\|_2$: $\|x\| = \sqrt{\sum_i x_i^2}$ for vector x. For a natural number m, $[m]$ denotes the set $\{1, 2, \ldots, m\}$. For a set A, $|A|$ means the number of elements. For any random vector X, $\mathbb{E}[X]$ and $\mathbb{V}[X]$ denote the mean and variance of X respectively. I_d denotes the identity matrix of size $d \times d$. If the size can be inferred from the context, we simply write I instead of I_d. $\mathrm{Unif}(a, b)$ means the uniform distribution with support (a, b).

2.1 Problem Settings and Motivations

Here, we present the details of our clustered IRM problem settings and their motivations.

Model. Let $\mathcal{E}$ be a set of environments. It is assumed that there is an unknown partition $\{\mathcal{E}_k\}_{k=1}^K$ of $\mathcal{E}$, i.e., $\mathcal{E} = \cup_{k=1}^K \mathcal{E}_k$ and $\mathcal{E}_k \cap \mathcal{E}_l = \emptyset$ for $k \neq l$. We consider the problem where each environment shares some invariant feature extractor Φ^* and cluster-dependent regressors $\{\beta_k^*\}_{k=1}^K$ on top of Φ^*. Specifically, the data generation processes for environment $e \in \mathcal{E}_k$ are defined as follows

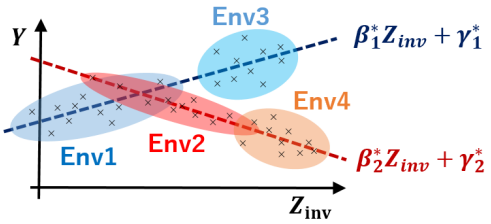


Figure 1: Illustration of our clustered settings: Environments 1 and 3 share the same true regressor, while Environments 2 and 4 share a distinct true regressor.

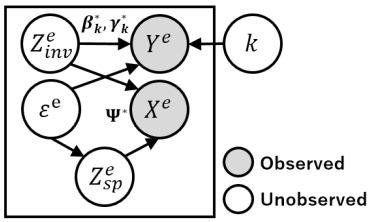


Figure 2: Graphical model of our data generation.

(Figure 2.1): for $Z_{\mathrm{inv}}^e \in \mathbb{R}^{d_{\mathrm{inv}}}$, $Z_{\mathrm{sp}}^e \in \mathbb{R}^{d_{\mathrm{sp}}}$, $Z^e := [(Z_{\mathrm{inv}}^e)^\top, (Z_{\mathrm{sp}}^e)^\top] \in \mathbb{R}^d$, $X^e \in \mathbb{R}^d$, where $d := d_{\mathrm{inv}} + d_{\mathrm{sp}}$, and $\beta_k^* \in \mathbb{R}^{d_{\mathrm{inv}}}$ and $Y^e, \gamma_k^*, \varepsilon^e \in \mathbb{R}$,

- $\mathbb{E}[Z_{\mathrm{inv}}^e] = 0$ and $\mathbb{V}[Z_{\mathrm{inv}}^e] = (\sigma_{\mathrm{inv}}^e)^2 I$,
- $\mathbb{E}[\varepsilon^e] = 0$, $\mathbb{V}[\varepsilon^e] = (\sigma_{\mathrm{tar}}^e)^2$ and $Z_{\mathrm{inv}}^e \perp \varepsilon^e$,
- $Y^e = (\beta_k^*)^\top Z_{\mathrm{inv}}^e + \gamma_k^* + \varepsilon^e$,
- $\mathbb{E}[Z_{\mathrm{sp}}^e] = 0$, $\mathbb{V}[Z_{\mathrm{sp}}^e] = (\sigma_{\mathrm{sp}}^e)^2 I$ and $Z_{\mathrm{sp}}^e \perp Z_{\mathrm{inv}}^e$,
- $X^e = \Psi^*(Z^e)$ for a bijective Ψ^*.

Remark (Generation process of Z_{sp}^e). Our model assumes that Z_{sp}^e depends on ε^e but not on Z_{inv}^e, and this may appear somewhat limited compared to the previous studies. However, under a mild condition, we can show that a generation process for Z_{sp}^e that does depend on Z_{inv}^e can effectively be addressed within the framework of the original problem. To illustrate, consider a new generation process where a random vector $\widetilde{Z}_{\mathrm{sp}}^e$ in $\mathbb{R}^{d_{\mathrm{sp}}}$ depends on both ε^e and Z_{inv}^e. Assume there exists a bijective function $\widetilde{\Psi} : \mathbb{R}^d \to \mathbb{R}^d$ such that $X^e = \widetilde{\Psi}(\widetilde{Z}^e)$, where $\widetilde{Z}^e := (Z_{\mathrm{inv}}^e, \widetilde{Z}_{\mathrm{sp}}^e)^\top$. Further, if we assume the existence of a bijective $H : \mathbb{R}^d \to \mathbb{R}^d$ such that $\widetilde{Z}^e = H(Z^e)$, we find that $\Psi^* := \widetilde{\Psi}^* \circ H$ is also bijective and $\Psi^*(Z^e) = X^e$. Therefore, the new scenario with $\widetilde{Z}_{\mathrm{sp}}^e \not\perp \widetilde{Z}_{\mathrm{inv}}^e$ essentially reduces to the original problem scenario with $Z_{\mathrm{sp}}^e \perp Z_{\mathrm{inv}}^e$.

We define *optimal feature extractor* $\Phi^* : \mathbb{R}^d \to \mathbb{R}^{d_{\mathrm{inv}}}$ by $\Phi^*(x) := [I_{d_{\mathrm{inv}}}, O]\,(\Psi^*)^{-1}(x)$ for $x \in \mathbb{R}^d$. Let $\mathcal{E}_{\mathrm{tr}}$ be the set of training environments. Similarly, we define $\{\mathcal{E}_{\mathrm{tr},k}\}_{k=1}^K$.

Objectives. This paper aims to achieve several objectives. The first objective is to identify an invariant predictor $\{\hat{\beta}_k \circ \hat{\Phi}, \hat{\gamma}_k\}_{k=1}^K$ and a partition $\{\hat{\mathcal{E}}_{\mathrm{tr},k}\}_{k=1}^K$ that satisfies $\hat{\beta}_k \circ \hat{\Phi}(X^e) + \hat{\gamma}_k = \beta_k^* \circ \Phi^*(X^e) + \gamma_k^*$ for every $e \in \mathcal{E}_{\mathrm{tr}}$, based on observations $(X^e, Y^e)_{e\in\mathcal{E}_{\mathrm{tr}}}$. The second objective is to demonstrate that the learned clustering structure accurately matches the true clustering structure. The final objective is to determine $\hat{k}$ such that $\hat{\beta}_{\hat{k}} \circ \hat{\Phi}(X^e) + \hat{\gamma}_{\hat{k}} = \beta_k^* \circ \Phi^*(X^e) + \gamma_k^*$ for a new environment $e \notin \mathcal{E}_{\mathrm{tr}}$, using additional observations from this new environment.

Applications. An important application involves robust prediction at test time in the trained environments, particularly when a type of concept drift arises from changes in spurious features. Suppose the generation process of the spurious features changes over time. In such cases, the standard ERM approach struggles to make accurate predictions after the change because it heavily relies on these spurious features during training. In contrast, IRM approaches focuses solely on capturing invariant features during training, enabling it to maintain effective prediction even after the change.

Another application involves few-shot learning on a new environment. After training regressors or classifiers and feature extractors by solving (1) on the training environments, we aim to transfer the trained predictors to a new environment. Directly transferring the trained predictors is generally challenging because it is unknown which cluster the new environment belongs to. However, it is possible to predict the cluster to which the new environment belongs by comparing the loss or accuracy on a few samples from the new environment. Once the cluster is determined, we can easily transfer the predictor to the new environment.

3 Clustered IRM and Its Optimization Problem Formulation

The following is our proposed optimization problem.

$$\begin{aligned}
&\text{For } \{\hat{\beta}_k\}_{k=1}^K \subset \mathbb{R}^{d_{\mathrm{inv}}},\ \{\hat{\gamma}_k\}_{k=1}^K \subset \mathbb{R},\ \hat{\Phi} \in \mathbb{R}^{d_{\mathrm{inv}}\times d},\ \{\hat{\mathcal{E}}_{\mathrm{tr},k}\}_{k=1}^K,\\
&\min_{\{\hat{\gamma}_k\},\hat{\Phi},\{\hat{\mathcal{E}}_{\mathrm{tr},k}\}} \frac{1}{|\mathcal{E}_{\mathrm{tr}}|}\sum_{e\in\mathcal{E}_{\mathrm{tr}}}\sum_{k=1}^K \mathbf{1}_{e\in\hat{\mathcal{E}}_{\mathrm{tr},k}} \mathcal{L}^e(\hat{\beta}_k,\hat{\gamma}_k,\hat{\Phi}) \qquad (1)\\
&\text{s.t. } \forall k\in[K], \forall e\in\hat{\mathcal{E}}_{\mathrm{tr},k}: \nabla_\beta \mathcal{L}^e(\beta,\hat{\gamma}_k,\hat{\Phi})|_{\beta=\hat{\beta}_k} = 0,\\
&\mathcal{E}_{\mathrm{tr}} = \cup_{k=1}^K \hat{\mathcal{E}}_{\mathrm{tr},k}, \forall k\neq l: \hat{\mathcal{E}}_{\mathrm{tr},k}\cap\hat{\mathcal{E}}_{\mathrm{tr},l} = \emptyset.
\end{aligned}$$

Here, $\mathcal{L}^e(\beta,\gamma,\Phi) := \mathbb{E}_{X^e,Y^e}[\ell(\beta^\top\Phi(X^e)+\gamma, Y^e)]$. For example, the squared loss $\ell(\hat{y},y) := (1/2)(\hat{y}-y)^2$ is used for regression tasks. Our true model $(\{\beta_k^*\}_{k=1}^K, \{\gamma_k^*\}_{k=1}^K, \Phi^*, \{\mathcal{E}_{\mathrm{tr},k}\}_{k=1}^K)$ satisfies the constraints and thus the optimization problem is at least feasible.

Problem (2) is a natural extension of the standard IRM formulation. The difference from the standard IRM is that the invariance constraint is imposed on each predicted cluster rather than on all the training environments. Additionally, the clustering constraint enforces that the predicted clusters must form a partition of the training environments.

By introducing new binary variables $\{\hat{p}^e\}_{e\in\mathcal{E}_{\mathrm{tr}}} \subset \{p\in\{0,1\}^K : \sum_{k=1}^K p_k = 1\}$, which are related to $\{\hat{\mathcal{E}}_{\mathrm{tr},k}\}_{k=1}^K$ by the equality $\{e\in\mathcal{E}_{\mathrm{tr}} : \hat{p}_k^e = 1\} = \hat{\mathcal{E}}_{\mathrm{tr},k}$, problem (1) can be equivalently transformed into the following simplified problem:

$$\begin{aligned}
&\min_{\{\hat{\gamma}_k\},\hat{\Phi},\{\hat{p}^e\}} \frac{1}{|\mathcal{E}_{\mathrm{tr}}|}\sum_{e\in\mathcal{E}_{\mathrm{tr}}}\sum_{k=1}^K \hat{p}_k L^e(\hat{\beta}_k,\hat{\gamma}_k,\hat{\Phi}) \qquad (2)\\
&\text{s.t. } \forall k\in[K], \forall e\in\mathcal{E}_{\mathrm{tr}}: \hat{p}_k^e \nabla_\beta \mathcal{L}^e(\beta,\hat{\gamma}_k,\hat{\Phi})|_{\beta=\hat{\beta}_k} = 0.
\end{aligned}$$

Practical Implementations The problem (1) or (2) is a bi-level optimization problem that involves combinatorial optimization due to its clustering nature, making it difficult to solve directly. To circumvent these challenges, we naturally relax the hard constraints to soft ones; binary vector $\hat{p}^e$ is replaced with the softmax output of real valued vector $\hat{s}^e$, and the IRM constraint is expressed as a regularization term. Specifically, we formulate a practical variant of (2) as follows.

$$\begin{aligned}
&\text{For } \{\hat{\beta}_k\}_{k=1}^K \subset \mathbb{R}^{d_{\mathrm{inv}}},\ \{\hat{\gamma}_k\}_{k=1}^K \subset \mathbb{R},\ \hat{\Phi} \in \mathbb{R}^{d_{\mathrm{inv}}\times d},\ \{\hat{s}^e\}_{e\in\mathcal{E}_{\mathrm{tr}}} \subset \mathbb{R}^K,\\
&\min_{\{\hat{\beta}_k\},\{\hat{\gamma}_k\},\hat{\Phi},\{\hat{s}^e\}} \frac{1}{|\mathcal{E}_{\mathrm{tr}}|}\sum_{e\in\mathcal{E}_{\mathrm{tr}}}\sum_{k=1}^K \sigma(\hat{s}^e)_k \hat{\mathcal{L}}^e(\hat{\beta}_k,\hat{\gamma}_k,\hat{\Phi})\\
&\qquad + \frac{\lambda}{2}\max_{e\in\mathcal{E}_{\mathrm{tr}}}\sum_{k=1}^K \left\|\sigma(\hat{s}^e)_k \nabla\hat{\mathcal{L}}^e(\beta,\hat{\gamma}_k,\hat{\Phi})|_{\beta=\hat{\beta}_k}\right\|^2. \qquad (3)
\end{aligned}$$

Here, $\hat{\mathcal{L}}^e$ denotes the finite sample approximation of $\mathcal{L}^e$, and $\sigma : \mathbb{R}^K \to [0,1]^K$ is the standard softmax function, which is a natural relaxation of the one-hot encoding. Consistent with prior research on invariant risk minimization, the invariance constraint is replaced by a penalty term on the gradient norm, with parameter λ. One difference from the standard penalty term in (Arjovsky et al., 2019) is the replacement of the mean operator with the max operator. This modification proved crucial for effectively capturing invariance in our numerical experiments. Empirical comparisons between the mean and max operators are detailed in Section 5.3. To optimize (3), any first-order methods can be applied in principle. In practice, we construct an unbiased estimator of the full gradient of the cost function. The specifics of creating an unbiased estimator for the penalty term using minibatches can be found in (Arjovsky et al., 2019).

Remark (Use of Gumbel-softmax). In our implementation, we adopted the Gumbel-softmax function σ_{GS}

instead of the standard softmax one. Here, $\sigma_{\mathrm{GS}} : \mathbb{R}^K \to [0,1]^K$ is defined as $\sigma_{\mathrm{GS}}(s)_k = \exp(s_k + g_k)/\sum_{k=1}^K \exp(s_k + g_k)$, where $g_k \overset{\text{i.i.d.}}{\sim} \mathrm{Gumbel}(0,1)$[2]. The introduction of Gumbel noise not only enhances the sparsity of the output probability distribution but also aids in $\{\hat{s}^e\}_{e\in\mathcal{E}_{\mathrm{tr}}}$ escape from local optimum. At inference time, we simply use argmax operator to determine the cluster index. Empirical comparisons between the naive softmax and the Gumbel-softmax are detailed in Section 5.3.

4 Theoretical Analysis

In this section, we present a rigorous theoretical analysis of the properties of the optimal solutions to problem (1). Our analysis primarily concentrates on the squared loss and is conducted under infinite samples settings All the proofs can be found in Section A.

Let $k(e)$ be the predicted cluster index of train environment $e \in \mathcal{E}$, i.e., the unique $k \in [K]$ such that $e \in \hat{\mathcal{E}}_{\mathrm{tr},k}$. Also, let $k^*(e)$ be the true cluster index of any environment $e \in \mathcal{E}$, i.e., the unique $k \in [K]$ such that $e \in \mathcal{E}_k$.

We constrain both the true and trained feature extractors to be linear, consistent with the theoretical analysis in the previous IRM literature.

Assumption 1. $\Psi^* : \mathbb{R}^d \to \mathbb{R}^d$ is linear. Also, we limit the functions class of latent variable extractor $\Phi : \mathbb{R}^d \to \mathbb{R}^{d_{\mathrm{inv}}}$ to be learned to the all linear functions.

From here, we identify liner functions Ψ^*, Φ^* and $\hat{\Phi}$ as matrix $\Psi^* \in \mathbb{R}^{d\times d}$ and $\Phi^*, \hat{\Phi} \in \mathbb{R}^{d_{\mathrm{inv}}\times d}$ respectively.

The essence of the standard IRM theory lies in assuming diversity among the environments. In the following assumption, we consider absolute continuity of the environment-specific parameters for simple discussion.

Assumption 2 (Diversity of environments)**.** $\{\mu_e, (\sigma^e_{\mathrm{inv}})^2, (\sigma^e_{\mathrm{sp}})^2\}_{e\in\mathcal{E}_{\mathrm{tr}}}$ obeys some absolute continuous probability distribution on $\Pi_{e\in\mathcal{E}_{\mathrm{tr}}}(\mathbb{R}^{d_{\mathrm{sp}}} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0})$, where $\mu_e := \mathbb{E}_{Z^e_{\mathrm{sp}},\varepsilon^e}[\varepsilon^e Z^e_{\mathrm{sp}}]$.

The following natural assumption helps identify the true cluster structure.

Assumption 3 (Well-conditionedness of the clustering problem)**.** $\mathcal{E}_{\mathrm{tr},k} \neq \emptyset$ for every $k \in [K]$, and $(\beta^*_k)^\top\Phi^* \neq (\beta^*_l)^\top\Phi^*$ or $\gamma^*_k \neq \gamma^*_l$ holds for every $k \neq l$

The following easy lemma given in Rosenfeld et al. (2020) demonstrates that the output of any linear feature extractor can be expressed as the sum of a linearly transformed invariant variable z_{inv} and a linearly transformed spurious variable z_{sp} under Assumption 1. This clarification is valuable as it highlights the concept of invariance within our problem settings: when $B_\Phi = 0$, the feature extractor Φ becomes invariant.

Lemma 4.1. *Suppose that Assumption 1 holds. For any $\Phi \in \mathbb{R}^{d_{\mathrm{inv}}\times d}$, there exist $A_\Phi \in \mathbb{R}^{d_{\mathrm{inv}}\times d_{\mathrm{inv}}}$ and $B_\Phi \in \mathbb{R}^{d_{\mathrm{inv}}\times d_{\mathrm{sp}}}$ such that $\Phi x = A_\Phi z_{\mathrm{inv}} + B_\Phi z_{\mathrm{sp}}$ for every $x \in \mathbb{R}^d$, where $z := (\Phi^*)^{-1}(x)$ and $z = [z_{\mathrm{inv}}^\top, z_{\mathrm{sp}}^\top]$.*

The following theorem demonstrates that the optimal solutions to our proposed problem (1) effectively capture the desired invariance under a training environments complexity of $O(d_{\mathrm{sp}} + K^2)$ without the knowledge of the true cluster structure. It's worth noting that standard IRM typically requires an environments complexity of $O(d_{\mathrm{sp}})$. This indicates that a naive application of IRM to each cluster would demand $O(Kd_{\mathrm{sp}})$ environments to capture invariance, which is much larger than ours, even if the cluster structure is known.

Theorem 4.2. *Let ℓ be the squared loss. Let $(\{\hat{\beta}_k\}_{k=1}^K, \{\hat{\gamma}_k\}_{k=1}^K, \hat{\Phi}, \{\hat{\mathcal{E}}_{\mathrm{tr},k}\}_{k=1}^K)$ be an optimal solution of (1). Suppose that Assumptions 1 and 2 hold. Then, if the number of training environments satisfies $|\mathcal{E}_{\mathrm{tr}}| \geq d_{\mathrm{sp}} + 2K^2$, then $\hat{\Phi}$ satisfies $B_{\hat{\Phi}} = 0$, i.e., the feature extractor does not depend on environmental feature z_{sp}, where $B_{\hat{\Phi}}$ is defined in Lemma 4.1.*

From Theorem 4.2, we can derive the following.

Corollary 4.3. *Suppose that the same conditions as Theorem 4.2 hold. Then, it is satisfied that*

1. *$\hat{\beta}^\top_{k(e)}\hat{\Phi} = (\beta^*_{k^*(e)})^\top\Phi^*$, $\gamma_{k(e)} = \gamma^*_{k^*(e)}$ for $\forall e \in \mathcal{E}_{\mathrm{tr}}$,*
2. *There exists a permutation $\pi : [K] \to [K]$ such that $\hat{\mathcal{E}}_{\mathrm{tr},\pi(k)} = \mathcal{E}_{\mathrm{tr},k}$ for $\forall k \in [K]$.*
3. *For $\forall e^{\mathrm{new}} \in \mathcal{E} \setminus \mathcal{E}_{\mathrm{tr}}$, $\hat{k}(e^{\mathrm{new}}) \in \mathrm{argmin}_{k\in[K]}\mathcal{L}^{e^{\mathrm{new}}}(\hat{\beta}_k, \hat{\gamma}_k, \hat{\Phi})$ satisfies $\hat{\beta}^\top_{\hat{k}(e^{\mathrm{new}})}\hat{\Phi} = (\beta^*_{k^*(e^{\mathrm{new}})})^\top\Phi^*$ and $\hat{\gamma}_{\hat{k}(e^{\mathrm{new}})} = \gamma^*_{k^*(e^{\mathrm{new}})}$.*

The first statement indicates that the predictor trained from problem (1) matches the true predictor for every training environment $e \in \mathcal{E}_{\mathrm{tr}}$. The second statement demonstrates that the trained cluster structure on $\mathcal{E}_{\mathrm{tr}}$ accurately captures the true one. The final statement implies that for any new environment $e^{\mathrm{new}} \in \mathcal{E} \setminus \mathcal{E}_{\mathrm{tr}}$, we can correctly predict the cluster index $k^*(e^{\mathrm{new}})$ by selecting the trained predictor that minimizes the risk $\mathcal{L}^{e^{\mathrm{new}}}$. While the last claim also assumes infinite samples, we can expect that identifying $k^*(e^{\mathrm{new}})$ requires only a few samples due to the finiteness of the hypothesis class. This point is validated in the numerical results given in Section 5.

[2]The p.d.f. of the standard Gumbel distribution $\mathrm{Gumbel}(0,1)$ is defined as $\exp(-(x + \exp(-x)))$.

Algorithm	Synthetic↓	Pathological↑	Rotated↑	Shifted↑	Hybrid↑
Global ERM	6.39 ± 0.71	0.45 ± 0.01	0.26 ± 0.0	0.18 ± 0.01	0.41 ± 0.01
Global IRM	6.13 ± 0.55	0.64 ± 0.03	0.36 ± 0.01	0.18 ± 0.01	0.6 ± 0.01
ClfPerEnv ERM	2.14 ± 0.48	0.63 ± 0.01	0.42 ± 0.01	0.45 ± 0.0	0.43 ± 0.01
EnvInd ERM	2.16 ± 0.51	0.63 ± 0.01	0.42 ± 0.01	0.44 ± 0.01	0.43 ± 0.0
ClustInd ERM	1.61 ± 0.41	0.62 ± 0.02	0.44 ± 0.01	0.45 ± 0.0	0.44 ± 0.01
ClustInd IRM	0.81 ± 0.35	0.84 ± 0.02	$\mathbf{0.53 \pm 0.02}$	0.57 ± 0.04	0.63 ± 0.01
ClustJoint ERM	1.63 ± 0.41	0.45 ± 0.0	0.33 ± 0.03	0.35 ± 0.02	0.41 ± 0.0
ClustJoint IRM	$\mathbf{0.72 \pm 0.28}$	$\mathbf{0.86 \pm 0.02}$	$\mathbf{0.53 \pm 0.03}$	$\mathbf{0.59 \pm 0.13}$	$\mathbf{0.65 \pm 0.02}$

Table 1: Comparison of the mean and standard deviation of the best test metric on the five datasets for **zero-shot learning on training environments**. The learning curves are found in Section B.3.

Algorithm	Synthetic ↓	Pathological↑	Rotated↑	Shifted↑	Hybrid↑
Global ERM	5.36 ± 0.42	0.83 ± 0.02	0.4 ± 0.02	0.37 ± 0.02	0.51 ± 0.01
Global IRM	5.04 ± 0.62	0.84 ± 0.02	0.43 ± 0.03	0.37 ± 0.02	0.61 ± 0.0
ClfPerEnv ERM	2.03 ± 1.19	0.86 ± 0.01	0.56 ± 0.01	0.59 ± 0.03	0.55 ± 0.02
EnvInd ERM	1.37 ± 1.07	0.84 ± 0.01	0.57 ± 0.01	0.57 ± 0.01	0.53 ± 0.02
ClustInd ERM	1.33 ± 0.86	$\mathbf{0.87 \pm 0.01}$	0.6 ± 0.03	0.58 ± 0.01	0.56 ± 0.01
ClustInd IRM	0.88 ± 0.7	$\mathbf{0.87 \pm 0.0}$	0.65 ± 0.01	$\mathbf{0.68 \pm 0.01}$	0.66 ± 0.02
ClustJoint ERM	1.9 ± 0.67	0.83 ± 0.03	0.51 ± 0.03	0.53 ± 0.06	0.52 ± 0.02
ClustJoint IRM	$\mathbf{0.85 \pm 0.55}$	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.69 \pm 0.02}$	0.65 ± 0.12	$\mathbf{0.68 \pm 0.0}$

Table 2: Comparison of the mean and standard deviation of the best test metric on the five datasets for **few-shot learning on test environments**. The learning curves are found in Section B.3.

5 Numerical Experiments

In this section, we provide numerical results to validate our theoretical results.

Table 5: The number of total samples per environment for each cluster in our numerical experiments.

Dataset	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Synthetic	$2,000$	$2,000$	$2,000$	$2,000$
Pathological	$3,110$	$2,793$	$2,638$	$3,011$
Rotated	$2,917$	$2,917$	$2,917$	$2,917$
Shifted	$2,917$	$2,917$	$2,917$	$2,917$
Hybrid	$3,182$	$3,182$	-	-

5.1 Setup

Here, we provide the experimental settings in detail.

Datasets and Tasks. We conducted experiments on both regression and classification tasks.

For the regression task, we created a synthetic dataset (referred to as "Synthetic") comprising four clusters that satisfy our theoretical settings. Each cluster consisted of five training environments and one test environment. The concrete generation procedures are found in Section B.1 of the supplementary material.

For the classification task, we utilized four types of clustered MNIST datasets as described in Zeng et al. (2023) using MNIST dataset(LeCun et al., 1998): Pathological MNIST, Rotated MNIST, Shifted MNIST, and Hybrid MNIST (we sometimes omit "MNIST"). Pathological MNIST is divided into four clusters: Cluster 1 contains only samples with labels $\{0, 1, 2\}$; Cluster 2 contains labels $\{3, 4\}$; Cluster 3 contains labels $\{5, 6\}$; and Cluster 4 contains labels $\{7, 8, 9\}$. In Rotated MNIST, images are rotated by 0, 90, 180, and 270 degrees, forming four distinct clusters. In Shifted MNIST, the label y is shifted to $(y + s)\%10$ where $s = 0, 3, 6, 9$, also resulting in four clusters. In Hybrid MNIST, there are two clusters: one containing standard MNIST samples and the other containing Fashion-MNIST(Xiao et al., 2017) samples. For every datasets, to increase the complexity of the problem, we randomly flipped the true label y^* to any value between 0 and 9 with a probability of 0.1 for each sample. We utilized this noisy label y during both training and testing phases. Additionally, to introduce spurious features, we modified the pixels in the training datasets for the training environments: the top, bottom, left, and right 2 pixels were replaced with $(z(y)/10) * 255$. Here, $z(y)$ was generated by flipping the noisy label y to any value between 0 and 9 with a probability $p^e(y)$, where $p^e(y) \sim \text{Unif}(0, 0.2)$ was independently sampled for each environment. This ensured that $z(e)$ was correlated with the true label y. For the test datasets of the training environments and the training/test datasets of the test environments, we

Algorithm	Synthetic ↓	Pathological↑	Rotated↑	Shifted↑	Hybrid↑
ClustJoint IRM (mean)	1.36 ± 0.29	0.63 ± 0.1	0.42 ± 0.04	0.44 ± 0.05	0.46 ± 0.02
ClustJoint IRM (max) w/o Gumbel	0.88 ± 0.38	$\mathbf{0.86 \pm 0.01}$	0.52 ± 0.03	$\mathbf{0.59 \pm 0.14}$	$\mathbf{0.66 \pm 0.02}$
ClustJoint IRM (max)	$\mathbf{0.72 \pm 0.28}$	$\mathbf{0.86 \pm 0.02}$	$\mathbf{0.53 \pm 0.03}$	$\mathbf{0.59 \pm 0.13}$	0.65 ± 0.02

Table 3: Comparison of the mean and standard deviation of the best test metric on the five datasets for **few-shot learning on test environments**.

Algorithm	Synthetic ↓	Pathological↑	Rotated↑	Shifted↑	Hybrid↑
ClustJoint IRM (mean)	1.39 ± 0.51	0.85 ± 0.01	0.6 ± 0.01	0.57 ± 0.09	0.57 ± 0.04
ClustJoint IRM (max) w/o Gumbel	1.08 ± 0.7	$\mathbf{0.87 \pm 0.01}$	0.62 ± 0.05	$\mathbf{0.65 \pm 0.12}$	0.67 ± 0.02
ClustJoint IRM (max)	$\mathbf{0.85 \pm 0.55}$	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.69 \pm 0.02}$	$\mathbf{0.65 \pm 0.12}$	$\mathbf{0.68 \pm 0.0}$

Table 4: Comparison of the mean and standard deviation of the best test metric on the five datasets for **few-shot learning on test environments**.

simply assigned 255 to the frame part of each image.

The information about the number of data samples per environment are provided in Table 5. The number of training environments was twenty, and the number of test environments was set to the number of clusters (two or four). For training environments, we randomly divided each dataset as 80% training dataset and 20% test dataset (before introducing the spurious features). The latter dataset was used to evaluate the performance of the prediction when the distribution of spurious features changes (i.e., zero-shot learning). For test environments, we randomly divided 0.5% training dataset for few-shot learning and 99.5% test dataset for evaluation of the few-shot learning.

Models. We constrained both regressors (or classifiers) and feature extractors to be linear in our experiments. For the regression task, we set the feature dimension to $d_{\mathrm{inv}} = 10$. For the classification tasks, we utilized a feature dimension of $d_{\mathrm{inv}} = 100$.

Evaluation. We compared the expected risk for the regression task and classification accuracy for the classification tasks, averaged over environments. Four independent random trials[3]were conducted, and we report the average and standard deviation of these metrics. We assessed the performance of the implemented algorithms in two scenarios. In the first scenario, we evaluated the algorithms on the test dataset of the training environments. It's important to note that the test datasets of the training environments had different distributions from the training datasets due to changes in the distributions of the spurious features. Hence, this scenario is considered as zero-shot learning. In the second scenario, we evaluated the algorithms on the test dataset of the test environments after fine-tuning them on the extremely small datasets of the test environments (roughly 10 to 15 samples per environment). This scenario is considered as few-shot learning.

Implemented Algorithms. Seven algorithms as baselines and the proposed method were implemented to demonstrate the effectiveness of our framework. *Global ERM* and *Global IRM* utilize a single classifier (or regressor) and feature extractor across all training environments. Global ERM optimizes ERM loss only, while Global IRM additionally optimizes the IRM penalty as in Arjovsky et al. (2019). *ClfPerEnv ERM* shares a single feature extractor, but each environment has its own classifier. In *EnvInd ERM*, each environment independently learns its own classifier and feature extractor. Since these algorithms learn per environment parameters, only ERM was applied. *ClustInd ERM* and *ClustInd IRM* assume knowledge of the true cluster structure and execute ERM and IRM on each cluster independently, respectively. Note that these two algorithms should be regarded as references because they rely on the true cluster structure. *ClustJoint ERM* corresponds to our proposed framework with $\lambda = 0$. *ClustJoint IRM* represents the proposed algorithm.

For IRM-type algorithms, we initially set $\lambda = 10^{-6}$ and after $3,000$ epochs, we changed it to $\lambda = 10^{4.4}$, and totally ran $30,000$ epochs. As noted in Section 3, we adopted the max operator rather than mean one for the IRM penalty term. For few-shot learning, we first choose the pair of the best classifier and feature extractor by evaluating the training loss or the training accuracy on the training datasets of the test environments (For ClustInd type algorithms, this process was omitted, and only the predictor corresponding to the true cluster was selected). Then, from the chosen classifier and feature extractor, we fully fine-tuned these

[3]The randomness included not only optimization process, but also the data generation process.

[4]Using small λ in initial learning phase in IRM leaning is theoretically verified in Chen et al. (2024).

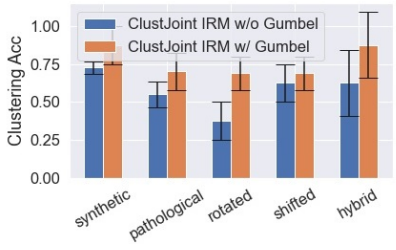


Figure 3: Clustering accuracy comparison between the proposed method without and with the Gumbel-softmax function.

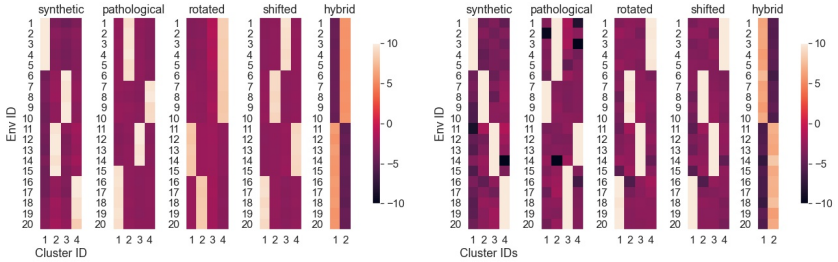


Figure 4: The clustering scores $\{\hat{s}_e\}_{e\in\mathcal{E}_{\text{tr}}}$ (each $\hat{s}_e \in \mathbb{R}^4$) at the final epoch of the proposed method on random seed 0. The environments IDs are sorted by the true cluster IDs. **Left**: *without* the Gumbel-softmax function. **Right**: *with* the Gumbel-softmax function.

parameters using the training datasets of the test environments. The details of the optimization algorithms and their hyper-parameter settings are provided in Section B.2.

5.2 Quantitative performance comparison

Tables 1 and 2 summarized the comparisons of the best test Mean Squared Error (MSE) for Synthetic dataset and the best test accuracy for four Clustered MNIST datasets. From these results, we can see that our proposed ClustJoint IRM performed significantly and consistently better than the other methods without using the true clustered structure information.[5] Furthermore, our method even outperformed ClustInd IRM for several cases both in zero-shot settings and few-shot ones. This is likely because ClustInd IRM does not share the feature extractor across environments, preventing it from utilizing information from the other environments. This is reflected in the theoretical superiority of ClustJoint IRM over ClustInd IRM in terms of environment complexity.

5.3 Ablation study

Here, we validate the effectiveness of adopting the max operator in the IRM penalty term and the use of Gumbel-softmax to transform real-valued scores $\{\hat{s}_e\}_{e\in\mathcal{E}_{\text{tr}}}$ into probability distributions.

Tables 3 and 4 compare the performances between the two baselines and the proposed method: (i) using the mean operator in the IRM penalty; (ii) using the max operator in the IRM penalty with the naive softmax function; and (iii) the proposed method ClustJoint IRM. From these results, we conclude that the techniques were effective.

To further analyze the impact of Gumbel-softmax in more detail, we measured the clustering accuracy for methods (ii) and (iii) based on their final epoch clustering scores. As shown in Figure 4, Gumbel-softmax consistently improved clustering performance over the naive softmax. Figure 4 visualizes the clustering scores for methods (ii) and (iii) on a single random seed. On the Rotated MNIST dataset, method (ii) failed to capture the true cluster structure, whereas method (iii) successfully identified the clusters. Generally, method (iii) achieved good clustering accuracy and this empirically validates our relaxed formulation for the hard clustering problem in problem (2).

6 Limitations and Future Work

There are several limitations to this work. The tightness of the theoretical complexity of training environments has not been investigated in this study. Additionally, our analysis focuses solely on the infinite-sample setting. In particular, the sample complexity of determining the true cluster indices of test environments in few-shot learning scenarios remains unexplored, making it an important direction for future research. Moreover, the data generation process is restricted to the case of linear feature extractors and linear regressors. Also, while our clustering assumption is limited to the simplest case, it can be extended to more general settings, such as fuzzy clustering, which presents an interesting avenue for future research. Since one of the applications of clustered IRM is in federated learning, developing distributed algorithms is also a crucial area for future work. In addition to these challenges, other challenges arise from the standard IRM framework itself. For example, Kamath et al. (2021) pointed out that IRMv1, which constrains the regressor to be linear and incorporates gradient norm regularization, which is

[5] In the few-shot learning settings on Pathological MNIST, the best test accuracy was competitive. This is probably because the features required to learn classifiers do not differ significantly across environments, making fine-tuning on Pathological MNIST relatively easy. In this case, a few samples may be sufficient to learn for all methods.

also adopted in our algorithm, may fail to capture the expected invariance in certain simple problem settings. Consequently, its generalization performance in new environments may degrade. Additionally, Gulrajani and Lopez-Paz (2020) introduced the benchmark DomainBed and, through careful experiments, suggested that IRM does not necessarily outperform ERM in domain generalization. Addressing these inherent limitations of IRM itself is also an important direction for future work.

7 Conclusion

We extended the standard IRM problems to clustered environments settings. We developed a new framework for simultaneously identifying the cluster structure on the environments and the invariant features. We showed that the required number of training environments for such identification is only $O(d_{\mathrm{sp}} + K^2)$, which can be much smaller than a naive bound $O(Kd_{\mathrm{sp}})$. In our numerical experiments, we compared our proposed method with several baselines in both zero-shot and few-shot learning settings. The results consistently demonstrated the superiority of our method over the baselines, including the per-cluster independent IRM, which utilized the true clustered structures.

Acknowledgments

TS was partially supported by JSPS KAKENHI (24K02905) and JST CREST (JPMJCR2015). This research is supported by the National Research Foundation, Singapore, Infocomm Media Development Authority under its Trust Tech Funding Initiative, and the Ministry of Digital Development and Information under the AI Visiting Professorship Programme (award number AIVP-2024-004). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, Infocomm Media Development Authority, and the Ministry of Digital Development and Information.

References

Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.-C., Bengio, Y., Mitliagkas, I., and Rish, I. (2021a). Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:3438–3450.

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

Ahuja, K., Wang, J., Dhurandhar, A., Shanmugam, K., and Varshney, K. R. (2021b). Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*.

Alesiani, F., Yu, S., and Niepert, M. (2023). Continual invariant risk minimization. *arXiv preprint arXiv:2310.13977*.

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

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

Chen, Y., Huang, W., Zhou, K., Bian, Y., Han, B., and Cheng, J. (2024). Understanding and improving feature learning for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 36.

Chen, Y., Zhou, K., Bian, Y., Xie, B., Wu, B., Zhang, Y., Ma, K., Yang, H., Zhao, P., Han, B., et al. (2022). Pareto invariant risk minimization: Towards mitigating the optimization dilemma in out-of-distribution generalization. *arXiv preprint arXiv:2206.07766*.

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

DeVries, T. and Taylor, G. W. (2018). Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. (2020). An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597.

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

Gupta, S., Ahuja, K., Havaei, M., Chatterjee, N., and Bengio, Y. (2022). Fl games: A federated learning framework for distribution shifts. *arXiv preprint arXiv:2205.11101*.

Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. (2020). The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR.

Huang, R., Geng, A., and Li, Y. (2021). On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34:677–689.

Jacob, L., Vert, J.-p., and Bach, F. (2008). Clustered multi-task learning: A convex formulation. *Advances in neural information processing systems*, 21.

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.

Konecnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 8.

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.

Lee, K., Lee, H., Lee, K., and Shin, J. (2017). Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

Li, H., Pan, S. J., Wang, S., and Kot, A. C. (2018a). Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409.

Li, P., Li, D., Li, W., Gong, S., Fu, Y., and Hospedales, T. M. (2021). A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895.

Li, Q. and Kim, B. M. (2003). Clustering approach for hybrid recommender system. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 33–38. IEEE.

Li, X., Dai, Y., Ge, Y., Liu, J., Shan, Y., and Duan, L.-Y. (2022). Uncertainty modeling for out-of-distribution generalization. *arXiv preprint arXiv:2202.03958*.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. (2018b). Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 624–639.

Liang, S., Li, Y., and Srikant, R. (2018). Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*.

Lin, Y., Dong, H., Wang, H., and Zhang, T. (2022a). Bayesian invariant risk minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16021–16030.

Lin, Y., Zhu, S., Tan, L., and Cui, P. (2022b). Zin: When and how to learn invariance without environment partition? *Advances in Neural Information Processing Systems*, 35:24529–24542.

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

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

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. (2013). Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207.

Lu, C., Wu, Y., Hernández-Lobato, J. M., and Schölkopf, B. (2021). Nonlinear invariant risk minimization: A causal approach. *arXiv preprint arXiv:2102.12353*.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. (2017). Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725.

Niu, Z., Yuan, J., Ma, X., Xu, Y., Liu, J., Chen, Y.-W., Tong, R., and Lin, L. (2023). Knowledge distillation-based domain-invariant representation learning for domain generalization. *IEEE Transactions on Multimedia*.

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*, 78(5):947–1012.

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

Sarwar, B. M., Karypis, G., Konstan, J., and Riedl, J. (2002). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324.

Sattler, F., Müller, K.-R., and Samek, W. (2020). Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. (2018). Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31.

Wang, H., Si, H., Li, B., and Zhao, H. (2022). Provable domain generalization via invariant-feature subspace recovery. In *International Conference on Machine Learning*, pages 23018–23033. PMLR.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yao, H., Wang, Y., Li, S., Zhang, L., Liang, W., Zou, J., and Finn, C. (2022). Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR.

Yong, L., Zhou, F., Tan, L., Ma, L., Liu, J., HE, Y., Yuan, Y., Liu, Y., Zhang, J. Y., Yang, Y., and Wang, H. (2024). Continuous invariance learning. In *The Twelfth International Conference on Learning Representations*.

Zeng, D., Hu, X., Liu, S., Yu, Y., Wang, Q., and Xu, Z. (2023). Stochastic clustered federated learning. *arXiv preprint arXiv:2303.00897*.

Zhang, Y., Sharma, P., Ram, P., Hong, M., Varshney, K., and Liu, S. (2023). What is missing in irm training and evaluation? challenges and solutions. *arXiv preprint arXiv:2303.02343*.

Zhou, X., Lin, Y., Zhang, W., and Zhang, T. (2022). Sparse invariant risk minimization. In *International Conference on Machine Learning*, pages 27222–27244. PMLR.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model.
   Yes: Section 2 gives our problem settings. Section 3 presents the optimization problems to be solved. Section 4 provides theoretical assumptions.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm.
   Yes: In Section 4, we derive theoretical environments complexity of the proposed method.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   Yes: We will submit anonymized source code as supplementary material to reproduce our numerical results.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   Yes: See Assumptions 1, 2, and 3 in Section 4.
   (b) Complete proofs of all theoretical results.
   Yes: The complete proofs of the statements in Section 4 are given in Section A.
   (c) Clear explanations of any assumptions.
   Yes: See Assumptions 1, 2, and 3 in Section 4.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
   Yes: Our source code will be published.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
   Yes: We provide the details of the experimental settings in Section 5.1, Section B.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
   Yes: See “Evaluation" part in Section 5.1.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
   Yes: It is given in Section B.6.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets.
   Yes: We cite MNIST and Fashion MNIST datasets.
   (b) The license information of the assets, if applicable.
   Not Applicable.
   (c) New assets either in the supplemental material or as a URL, if applicable.
   Not Applicable: We only used public datasets.
   (d) Information about consent from data providers/curators.
   Not Applicable: We only used public datasets.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
   Not Applicable: We only used public datasets.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots.
   Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
   Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
   Not Applicable.

A Theoretical Analysis

A.1 Proof of Lemma 4.1

From Assumption 1, $\Psi^* = [\Psi^*_{\mathrm{inv}}, \Psi^*_{\mathrm{sp}}]$ satisfies $\Psi^*(z) = \Psi^*_{\mathrm{inv}} z_{\mathrm{inv}} + \Psi^*_{\mathrm{sp}} z_{\mathrm{sp}}$ for any $z = [z_{\mathrm{inv}}^\top, z_{\mathrm{sp}}^\top]^\top \in \mathbb{R}^d$. Also, since Φ is linear, we can observe that $\Phi(x) = \Phi\Psi^* z = \Phi\Psi^*_{\mathrm{inv}} z_{\mathrm{inv}} + \Phi\Psi^*_{\mathrm{sp}} z_{\mathrm{sp}}$ for any $x \in \mathbb{R}^d$, where $z = (\Psi^*)^{-1}(x)$. Thus, we can see that there exist A_Φ and B_Φ such that $\Phi(x) = A_\Phi z_{\mathrm{inv}} + B_\Phi z_{\mathrm{sp}}$ for any $x \in \mathbb{R}^d$. This finishes the proof.

A.2 Proof of Theorem 4.2

Strategy The proof can be viewed as an extension of that in Rosenfeld et al. (2020) to our clustered IRM setting.[6] Our goal is to show that $B_{\hat{\Phi}} = 0$, which implies that the extracted features do not contain spurious information, where $B_{\hat{\Phi}}$ appears in Lemma 4.1. The core idea in Rosenfeld et al. (2020) is to construct $\Theta(1)$ linear equations involving $B_{\hat{\Phi}}$ from the IRM constraint. In our case, we carefully derive homogeneous linear equations involving $B_{\hat{\Phi}}$ with a full-rank coefficient matrix, where the full-rank property is ensured by Assumption 2, which imposes the diversity of the environments. Since we must account for all possible pairs of K predicted regressors $\{\hat{\beta}_k\}_{k\in[K]}$ and K true regressors $\{\beta^*_k\}_{k\in[K]}$ in our cluster settings, it follows that a total of $\Theta(K^2)$ equations are required, which is reflected in the resulting environment complexity of $O(d_{\mathrm{sp}} + K^2)$.

We now proceed to the concrete proof.

Let $(\{\hat{\beta}_k\}_{k=1}^K, \{\hat{\gamma}_k\}_{k=1}^K, \hat{\Phi}, \{\hat{\mathcal{E}}_{\mathrm{tr},k}\}_{k=1}^K)$ be any optimal solution of optimization problem (1). From Lemma 4.1, we have $\Phi(x) = A_{\hat{\Phi}} z_{\mathrm{inv}} + B_{\hat{\Phi}} z_{\mathrm{sp}}$ for any $x \in \mathbb{R}^d$, where $z := (\Phi^*)^{-1}(x)$. For simple notation, A and B denote $A_{\hat{\Phi}}$ and $B_{\hat{\Phi}}$ respectively in the following.

First, we define $C_{k,l} := \mathcal{E}_{\mathrm{tr},k} \cap \hat{\mathcal{E}}_{\mathrm{tr},l}$ for $k, l \in [K]$.

Observe that $\mathbb{E}_{X^e}[\hat{\Phi}(X^e)] = A\mathbb{E}_{Z^e_{\mathrm{inv}}}[Z^e_{\mathrm{inv}}] + B\mathbb{E}_{Z^e_{\mathrm{sp}}}[Z^e_{\mathrm{sp}}] = 0$.

Let $k(e)$ and $k^*(e)$ be the predicted and true cluster index of e respectively.

From the invariance constraint, for $e \in \mathcal{E}_{\mathrm{tr}}$, we have

$$\mathbb{E}_{X^e}[\hat{\Phi}(X^e)\hat{\Phi}(X^e)^\top]\hat{\beta}_{k(e)} = \mathbb{E}_{X^e,Y^e}[Y^e\hat{\Phi}(X^e)].$$

Since $\hat{\Phi}(X^e) = AZ_{\mathrm{inv}} + BZ_{\mathrm{sp}}$ and $Z_{\mathrm{inv}} \perp Z_{\mathrm{sp}}$, we have

$$\begin{aligned}\mathbb{E}[\hat{\Phi}(X^e)\hat{\Phi}(X^e)^\top] &= \mathbb{E}_{X^e}[AZ_{\mathrm{inv}}(AZ_{\mathrm{inv}})^\top + (AZ_{\mathrm{inv}})(BZ_{\mathrm{sp}})^\top + BZ_{\mathrm{sp}}(BZ_{\mathrm{sp}})^\top] \\ &= (\sigma^e_{\mathrm{inv}})^2 AA^\top + (\sigma^e_{\mathrm{sp}})^2 BB^\top.\end{aligned}$$

Also, since $Y^e = (Z^e_{\mathrm{inv}})^\top \beta^*_{k^*(e)} + \gamma_{k^*(e)} + \varepsilon^e$ and $Z^e_{\mathrm{inv}} \perp \varepsilon^e$, we have

$$\begin{aligned}\mathbb{E}_{X^e,Y^e}[Y^e\hat{\Phi}(X^e)] &= \mathbb{E}_{X^e,Y^e}[(Z^e_{\mathrm{inv}})^\top \beta^*_{k^*(e)} + \gamma_{k^*(e)} + \varepsilon^e)(AZ^e_{\mathrm{inv}} + BZ^e_{\mathrm{sp}})] \\ &= (\sigma^e_{\mathrm{inv}})^2 A\beta^*_{k^*(e)} + B\mu_e,\end{aligned}$$

where μ^e is defined as $\mathbb{E}_{Z^e_{\mathrm{sp}}}[\varepsilon^e Z^e_{\mathrm{sp}}]$.

Then, we get for every $e \in \hat{\mathcal{E}}_{\mathrm{tr},k}$

$$((\sigma^e_{\mathrm{inv}})^2 AA^\top + (\sigma^e_{\mathrm{sp}})^2 BB^\top)\hat{\beta}_{k(e)} = (\sigma^e_{\mathrm{inv}})^2 A\beta^*_{k^*(e)} + B\mu^e.$$

Note that we can write

$$\hat{\beta}_{k(e)} = \sum_{k=1}^K \mathbf{1}_{e\in\hat{\mathcal{E}}_{\mathrm{tr},k}}\hat{\beta}_k = \sum_{k=1}^K\sum_{l=1}^K \mathbf{1}_{e\in C_{k,l}}\hat{\beta}_l$$

[6] While Rosenfeld et al. (2020) focus on a special case of binary classification, we focus on regression with squared loss. However, this distinction is not essential to the core argument.

and

$$\beta^*_{k^*(e)} = \sum_{k=1}^{K} \mathbf{1}_{e\in\mathcal{E}_{\mathrm{tr},k}} \beta^*_k = \sum_{k=1}^{K}\sum_{l=1}^{K} \mathbf{1}_{e\in C_{k,l}} \beta^*_k.$$

Using these equalities, we have

$$\sum_{k=1}^{K}\sum_{l=1}^{K} (\sigma^e_{\mathrm{inv}})^2 \mathbf{1}_{e\in C_{k,l}} (AA^\top \hat{\beta}_l - A\beta^*_k) + \sum_{k=1}^{K} (\sigma^e_{\mathrm{sp}})^2 \mathbf{1}_{e\in\hat{\mathcal{E}}_{\mathrm{tr},k}} BB^\top \hat{\beta}_k = B\mu^e.$$

Without loss of generality, we can assume $|\hat{\mathcal{E}}_{\mathrm{tr},1}| \geq \cdots \geq |\hat{\mathcal{E}}_{\mathrm{tr},K}|$. Then, we define K_0 as the largest index $k \in [K]$ that satisfies $|\hat{\mathcal{E}}_{\mathrm{tr},k}| \geq K+1$. Note that this definition is well-defined because $|\mathcal{E}_{\mathrm{tr}}| \geq K^2+1$ is assumed.

We pick up $\{e'_i\}_{i=1}^{K_0} \subset \mathcal{E}_{\mathrm{tr}}$ that satisfies $e'_i \in C_{k,i}$ for some $k \in [K]$ with $|C_{k,i}| \geq 2$ for each $i \in [K_0]$. This is always possible because for each $i \in [K_0]$, there exists $k \in [K]$ such that $|C_{k,i}| \geq 2$ from $|\hat{\mathcal{E}}_{\mathrm{tr},i}| \geq K+1$.

Let G be the set $\{(k,l) \in [K] \times [K_0] : C_{k,l} \neq \emptyset\}$. Additionally, let $o : G \to [|G|]$ denote any bijective function that orders the elements of G.

Then, we choose $\{e_i\}_{i=1}^{d_{\mathrm{sp}}} \subset \mathcal{E}_{\mathrm{tr}} \setminus \{e'_i\}_{i=1}^{K_0}$ that satisfies

- $e_i \in \cup_{(k,l)\in G} C_{k,l}$,
- $\forall (k,l) \in G, \exists i \in [d_{\mathrm{sp}}] : e_i \in C_{k,l} \setminus \{e'_l\}$.

This is always possible because $|\cup_{(k,l)\in G} C_{k,l} \setminus \{e'_i\}_{i=1}^{K_0}| = |\mathcal{E}_{\mathrm{tr}}| - (K-K_0)K - K_0 \geq |\mathcal{E}_{\mathrm{tr}}| - K^2 \geq d_{\mathrm{sp}} + K^2$ from the assumption $|\mathcal{E}_{\mathrm{tr}}| \geq d_{\mathrm{sp}} + 2K^2$, and $|C_{k,i} \setminus \{e'_i\}| \geq 1$ for k satisfying $e'_i \in C_{k,i}$ for every $i \in [K_0]$.

From Assumption 2, it holds that $\{\mu_{e_i}\}_{i=1}^{d_{\mathrm{sp}}}$ is linearly independent with probability one. Thus, for every $e \in \mathcal{E}_{\mathrm{tr}}$, it holds that $\mu_e = \sum_{i=1}^{d_{\mathrm{sp}}} \alpha^e_i \mu_{e_i}$ for some $\{\alpha^e_i\}_{i=1}^{d_{\mathrm{sp}}}$.

Then, for any $e \in \cup_{(k,l)\in G} C_{k,l}$, since

$$\begin{aligned} B\mu_e &= \sum_{i=1}^{d_{\mathrm{sp}}} \alpha^e_i B\mu_{e_i} \\ &= \sum_{(k,l)\in G}\sum_{i=1}^{d_{\mathrm{sp}}} \alpha^e_i (\sigma^{e_i}_{\mathrm{inv}})^2 \mathbf{1}_{e_i\in C_{k,l}} (AA^\top \hat{\beta}_l - A\beta^*_k) + \sum_{k\in[K_0]}\sum_{i=1}^{d_{\mathrm{sp}}} \alpha^e_i (\sigma^{e_i}_{\mathrm{sp}})^2 \mathbf{1}_{e_i\in\hat{\mathcal{E}}_{\mathrm{tr},k}} BB^\top \hat{\beta}_k, \end{aligned} \tag{4}$$

we get

$$\sum_{(k,l)\in G} u(e, o^{-1}(k,l))(AA^\top \hat{\beta}_l - A\beta^*_k) + \sum_{k\in[K_0]} v(e,k) BB^\top \hat{\beta}_k = 0.$$

where

$$u(e, o^{-1}(k,l)) := (\sigma^e_{\mathrm{inv}})^2 \mathbf{1}_{e\in C_{k,l}} - \sum_{i=1}^{d_{\mathrm{sp}}} \alpha^e_i (\sigma^{e_i}_{\mathrm{inv}})^2 \mathbf{1}_{e_i\in C_{k,l}}$$

for $(k,l) \in G$, and

$$v(e,k) := (\sigma^e_{\mathrm{sp}})^2 \mathbf{1}_{e\in\hat{\mathcal{E}}_{\mathrm{tr},k}} - \sum_{i=1}^{d_{\mathrm{sp}}} \alpha^e_i (\sigma^{e_i}_{\mathrm{sp}})^2 \mathbf{1}_{e_i\in\hat{\mathcal{E}}_{\mathrm{tr},k}}$$

for $k \in [K_0]$.

Now, we arbitrarily pick up $\{e'_i\}_{i=K_0+1}^{K_0+|G|} \subset \mathcal{E}_{\mathrm{tr}} \setminus (\{e_i\}_{i=1}^{d_{\mathrm{sp}}} \cup \{e'_i\}_{i=1}^{K_0})$. This is possible because $|G| \leq K^2$.

Then, we will show that matrix $T := \begin{bmatrix} U_1 & V_1 \\ U_2 & V_2 \end{bmatrix} \in \mathbb{R}^{(K_0+|G|)\times(K_0+|G|)}$ is full-rank, where

$$U_1 := \begin{bmatrix} u(e'_1,1) & \cdots & u(e'_1,|G|) \\ \vdots & \dots & \vdots \\ u(e'_{K_0},1) & \cdots & ,u(e'_{K_0},|G|) \end{bmatrix},$$

$$U_2 := \begin{bmatrix} u(e'_{K_0+1},1) & \cdots & u(e'_{K_0+1},|G|) \\ \vdots & \dots & \vdots \\ u(e'_{K_0+|G|},1) & \cdots & ,u(e'_{K_0+|G|},|G|) \end{bmatrix},$$

$$V_1 := \begin{bmatrix} v(e'_1,1) & \cdots & v(e'_1,K_0) \\ \vdots & \dots & \vdots \\ v(e'_{K_0},1) & \cdots & v(e'_{K_0},K_0) \end{bmatrix}$$

and

$$V_2 := \begin{bmatrix} v(e'_{K_0+1},1) & \cdots & v(e'_{K_0+1},K_0) \\ \vdots & \dots & \vdots \\ v(e'_{K_0+|G|},1) & \cdots & v(e'_{K_0+|G|},K_0) \end{bmatrix}.$$

If this is proven, we immediately obtain $AA^\top \hat{\beta}_l - A\beta^*_k = 0$ for every $(k,l) \in G$ and $BB^\top \hat{\beta}_k = 0$ for every $k \in [K_0]$. This implies $B\mu_{e_i} = 0$ for every $i \in [d_{\mathrm{sp}}]$ from (4), and since $\{\mu_{e_i}\}_{i=1}^{d_{\mathrm{sp}}}$ is linearly independent, $B = 0$ must hold.

Full-rankness of T

Now, we show the full-rankness of square matrix T.

Lemma A.1. *Let $\{x_i\}_{i=1}^d$ be a linearly independent sequence on $\mathbb{R}^d$ and $\check{x}_{d+1} \in \mathbb{R}$ be generated from an absolute continuous variable. Given any $\{\check{x}_i\}_{i=1}^d \subset \mathbb{R}$ and $x_{d+1} \in \mathbb{R}^d$, $\{\widetilde{x}_i\}_{i=1}^{d+1} \subset \mathbb{R}^{d+1}$ is linearly independent with probability one, where $\widetilde{x}_i := [x_i^\top, \check{x}_i]^\top$.*

Proof. Since $\{x_i\}_{i=1}^d$ is linearly independent, $\{\widetilde{x}_i\}_{i=1}^d$ is linearly independent. Thus, what we only need to do is to show $\widetilde{x}_{d+1}$ cannot be expressed as a linear combination of $\{\widetilde{x}_i\}_{i=1}^d$ almost surely. Note that x_{d+1} is expressed as an unique linear combination $x_{d+1} = \sum_{i=1}^d c_i x_i$. However, $\check{x}_{d+1} = \sum_{i=1}^d c_i \check{x}_i$ only happens with probability zero since $\{\check{x}_i\}_{i=1}^d$ is fixed and $\check{x}_{d+1}$ is generated from an absolute continuous random variable on $\mathbb{R}$. Thus, $\{\widetilde{x}_i\}_{i=1}^{d+1}$ is linearly independent almost surely. This finishes the proof. □

Let variables $\{(\sigma_{\mathrm{inv}}^{e_i})^2, (\sigma_{\mathrm{sp}}^{e_i})^2\}_{i=1}^{d_{\mathrm{sp}}}$, $\{(\sigma_{\mathrm{inv}}^{e'_i})^2\}_{i=1}^{K_0+|G|}$ and $\{\alpha_i^{e'_j}\}_{j=1}^{K_0}$ be arbitrary fixed. Note that from Assumption 2, $\{\alpha_i^{e'_j}\}_{i=1,j=K_0+1}^{d_{\mathrm{sp}},K_0+|G|}$ obeys some absolute continuous distribution conditioned on the fixed variables. Then, we can see that square matrix U_2 also obeys some absolute continuous distribution and thus it is full-rank.

Next, when we further conditioned on $\{\alpha_i^{e'_j}\}_{i=1,j=K_0+1}^{d_{\mathrm{sp}},K_0+|G|}$, we can see that $\{(\sigma_{\mathrm{sp}}^{e'_i})^2\}_{i=1}^{K_0}$ (and thus the diagonal entries of V_1) still obeys some absolute continuous distribution from Assumption 2. Recursively applying Lemma A.1 results in the almost sure full-rankness of T.

A.3 Proof of Corollary 4.3

As in the proof of Theorem 4.2, A and B denote $A_{\hat{\Phi}}$ and $B_{\hat{\Phi}}$ respectively for simple notation.

$\hat{\beta}_{k(e)}^\top \hat{\Phi} = (\beta^*_{k^*(e)})^\top \Phi^*$ and $\hat{\gamma}_{k(e)} = \gamma^*_{k^*(e)}$ for every $e \in \mathcal{E}_{\mathrm{tr}}$

First, note that $\hat{\Phi}(x) = Az_{\mathrm{inv}} = A\Phi^*(x)$ for any $x \in \mathbb{R}^d$, where $z_{\mathrm{inv}} := \Phi^*(x)$, since $B = 0$. Thus, we have $\hat{\Phi} = A\Phi^*$.

We can see that $\sum_{e\in\mathcal{E}_{\rm tr}}\mathcal{L}^e(\hat{\beta}_{k(e)},\hat{\gamma}_{k(e)},\hat{\Phi})=\sum_{e\in\mathcal{E}_{\rm tr}}\mathcal{L}^e(\beta^*_{k^*(e)},\gamma^*_{k^*(e)},\Phi^*)$. To see this, we first show that $l.h.s.\geq r.h.s.$: for every $e\in\mathcal{E}_{\rm tr}$,

$$\begin{aligned}\mathcal{L}^e(\hat{\beta}_{k(e)},\hat{\gamma}_{k(e)},\hat{\Phi}) &= \mathbb{E}_{X^e,Y^e}[\ell(\hat{\beta}_{k(e)}^\top\hat{\Phi}(X^e)+\hat{\gamma}_{k(e)},Y^e)]\\ &= \mathbb{E}_{Z^e,Y^e}[\ell((A^\top\hat{\beta}_{k(e)})^\top Z^e_{\rm inv}+\hat{\gamma}_{k(e)}),Y^e)]\\ &\geq \mathbb{E}_{Z^e,Y^e}[\ell((\beta^*_{k^*(e)})^\top Z^e_{\rm inv})+\gamma^*_{k^*(e)},Y^e)]\\ &= \mathcal{L}^e(\beta^*_{k^*(e)},\gamma^*_{k^*(e)},\Phi^*)\\ &= (\sigma^e_{\rm tar})^2.\end{aligned}$$

Here, the inequality holds because $\beta^*_{k^*(e)}$, $\gamma^*_{k^*(e)}$ is the minimizer of $\mathcal{L}^e(\beta,\gamma,\Phi^*)$ with minimum $(\sigma^e_{\rm tar})^2$. On the other side, since $(\{\beta^*_k\}_{k=1}^K,\{\gamma^*_k\}_{k=1}^K,\Phi^*,\{\mathcal{E}_{{\rm tr},k}\}_{k=1}^K)$ is a feasible solution of the optimization problem (1), we immediately get $l.h.s.\leq r.h.s.$ since $(\{\hat{\beta}_k\}_{k=1}^K,\{\hat{\gamma}_k\}_{k=1}^K,\hat{\Phi},\{\hat{\mathcal{E}}_{{\rm tr},k}\}_{k=1}^K)$ and Φ is an optimal solution of (1).

Then, observe that

$$\begin{aligned}&\sum_{e\in\mathcal{E}_{\rm tr}}\mathbb{E}_{X^e}\left\{\left((\hat{\beta}_{k(e)}^\top\hat{\Phi}(X^e)+\hat{\gamma}_{k(e)}\right)-\left((\beta^*_{k^*(e)})^\top\Phi^*(X^e)+\gamma^*_{k^*(e)}\right)\right\}^2\\ =&\sum_{e\in\mathcal{E}_{\rm tr}}\mathbb{E}_{X^e,\varepsilon^e}\left\{\left((\hat{\beta}_{k(e)}^\top\hat{\Phi}(X^e)+\hat{\gamma}_{k(e)}-Y^e\right)+\varepsilon^e\right\}^2\\ =&\sum_{e\in\mathcal{E}_{\rm tr}}(\mathcal{L}^e(\hat{\beta}_{k(e)},\hat{\gamma}_{k(e)},\hat{\Phi})-(\sigma^e_{\rm tar})^2)\\ =&\sum_{e\in\mathcal{E}_{\rm tr}}(\mathcal{L}^e(\hat{\beta}_{k(e)},\hat{\gamma}_{k(e)},\hat{\Phi})-\mathcal{L}^e(\beta^*_{k^*(e)},\gamma^*_{k^*(e)},\Phi^*))=0\end{aligned}$$

and thus we obtain

$$\hat{\beta}_{k(e)}^\top\hat{\Phi}(X^e)+\hat{\gamma}_{k(e)}=(\beta^*_{k^*(e)})^\top\Phi^*(X^e)+\gamma^*_{k^*(e)}$$

almost surely for every $e\in\mathcal{E}_{\rm tr}$. Then, it hold that $(A^\top\hat{\beta}_{k(e)})^\top Z^e_{\rm inv}+\hat{\gamma}_{k(e)}=(\beta^*_{k^*(e)})^\top Z^e_{\rm inv}+\gamma^*_{k^*(e)}$. Thus, taking expectations with respect to $Z^e_{\rm inv}$ gives $\hat{\gamma}_{k(e)}=\gamma^*_{k^*(e)}$ because $\mathbb{E}[Z^e]=0$. Also, we can see that $A^\top\hat{\beta}_{k(e)}=\beta^*_{k^*(e)}$ by multiplying $(Z^e_{\rm inv})^\top$ from right and taking expectations with respect to $Z^e_{\rm inv}$. Therefore, we obtain $\hat{\beta}_{k(e)}^\top\hat{\Phi}=(\beta^*_{k^*(e)})^\top\Phi^*$.

$\hat{\mathcal{E}}_{{\rm tr},\pi(k)}=\mathcal{E}_{{\rm tr},k}$ for every $k\in[K]$ for some permutation π

First, observe that for every $e_1,e_2\in\hat{\mathcal{E}}_{{\rm tr},k}$, $k^*(e_1)=k^*(e_2)$. This is because $(\beta^*_{k^*(e_1)})^\top\Phi^*=\hat{\beta}_k^\top\hat{\Phi}=(\beta^*_{k^*(e_2)})^\top\Phi^*$ and $\gamma_{k^*(e_1)}=\hat{\gamma}_k=\gamma_{k^*(e_2)}$, and then from Assumption 3, we have $k^*(e_1)=k^*(e_2)$.

Based on these observations, we define $\pi:\mathcal{K}_0\to[K]$, where $\mathcal{K}_0:=\{k\in[K]:\hat{\mathcal{E}}_{{\rm tr},k}\neq\emptyset\}$, as follows: for each $k\in\mathcal{K}_0$, we pick arbitrary $e\in\hat{\mathcal{E}}_{{\rm tr},k}$ and assign $\pi(k):=k^*(e)$. Note that this definition does not depend on the choice of e from the above observation. Note that $\hat{\mathcal{E}}_{{\rm tr},k}\subset\mathcal{E}_{{\rm tr},\pi(k)}$ for every $k\in\mathcal{K}_0$.

Now, we will show that $\mathcal{K}_0$ is actually $[K]$. To see this, observe that $|\mathcal{E}_{\rm tr}|=\sum_{k\in\mathcal{K}_0}|\hat{\mathcal{E}}_{{\rm tr},k}|=\sum_{k'\in{\rm Im}(\pi)}\sum_{k\in\mathcal{K}_0}\mathbf{1}_{\pi(k)=k'}|\hat{\mathcal{E}}_{{\rm tr},k}|\leq\sum_{k'\in{\rm Im}(\pi)}|\mathcal{E}_{{\rm tr},k'}|$. Here, the last inequality comes from the fact that $\{\hat{\mathcal{E}}_{{\rm tr},k}\}_{k\in[K]}$ is a partition and $\hat{\mathcal{E}}_{{\rm tr},k}\subset\mathcal{E}_{{\rm tr},\pi(k)}$. Thus, if $\mathcal{K}_0\subsetneq[K]$, we would have a contradiction because $|{\rm Im}(\pi)|\leq|\mathcal{K}_0|<K$ and thus the right hand side would be smaller than $|\mathcal{E}_{\rm tr}|$ from the non-emptyness of $\mathcal{E}_{{\rm tr},k}$ for every $k\in[K]$ in Assumption 3.

Also, we can easily prove that π is surjective by using a similar arguments in the proof of $\mathcal{K}_0=[K]$: if ${\rm Im}(\pi)$ was not $[K]$, then we would have a contradiction. This means that π is actually bijective since π is a function from a finite set $[K]$ to $[K]$.

Since π is bijective, we can show that $\hat{\mathcal{E}}_{{\rm tr},k}=\mathcal{E}_{{\rm tr},\pi(k)}$ for every $k\in[K]$ as follows. Suppose that there exist $k_0\in[K]$ and $e\in\mathcal{E}_{\rm tr}$ such that $e\in\mathcal{E}_{{\rm tr},\pi(k_0)}\setminus\hat{\mathcal{E}}_{{\rm tr},k_0}$ holds. Then, it holds that $e\in\mathcal{E}_{{\rm tr},\pi(k(e))}$, and thus $\pi(k(e))=\pi(k_0)$. From the bijectivity of π, we have $k(e)=k_0$. This contradicts $e\notin\hat{\mathcal{E}}_{{\rm tr},k_0}$. Thus, we get $\mathcal{E}_{{\rm tr},\pi(k)}\subset\hat{\mathcal{E}}_{{\rm tr},k}$ and thus $\mathcal{E}_{{\rm tr},\pi(k)}=\hat{\mathcal{E}}_{{\rm tr},k}$ for every $k\in[K]$. Finally, taking the inverse of π gives the desired result.

$\hat{\beta}_{\hat{k}(e^{\mathrm{new}})}^{\top}\hat{\Phi} = (\beta^*_{k^*(e^{\mathrm{new}})})^{\top}\Phi^*$ **and** $\hat{\gamma}_{\hat{k}(e^{\mathrm{new}})} = \gamma^*_{k^*(e^{\mathrm{new}})}$ **for every** $e^{\mathrm{new}} \in \mathcal{E}$

First note that there exists $e^{\mathrm{tr}} \in \mathcal{E}_{\mathrm{tr}}$ such that $k^*(e^{\mathrm{tr}}) = k^*(e^{\mathrm{new}})$, because $\mathcal{E}_{\mathrm{tr},k} \neq \emptyset$ for every $k \in [K]$. Also, note that $\hat{\mathcal{E}}_{\mathrm{tr},\pi^{-1}(k^*(e^{\mathrm{tr}}))} = \mathcal{E}_{\mathrm{tr},k^*(e^{\mathrm{tr}})}$ from the second statement of Corollary 4.3.

Observe that $\pi^{-1}(k^*(e^{\mathrm{tr}})) \in [K]$ satisfies

$$\begin{aligned}
&\mathcal{L}^{e^{\mathrm{new}}}((\hat{\beta}_{\pi^{-1}(k^*(e^{\mathrm{tr}}))}, \hat{\gamma}_{\pi^{-1}(k^*(e^{\mathrm{tr}}))}, \hat{\Phi})) \\
&= \mathbb{E}_{X^{e^{\mathrm{new}}}, Y^{e^{\mathrm{new}}}}[\ell(\hat{\beta}_{\pi^{-1}(k^*(e^{\mathrm{tr}}))}^{\top}\hat{\Phi}(X^{e^{\mathrm{new}}}) + \hat{\gamma}_{\pi^{-1}(k^*(e^{\mathrm{tr}}))}, Y^{e^{\mathrm{new}}})] \\
&= \mathbb{E}_{X^{e^{\mathrm{new}}}, Y^{e^{\mathrm{new}}}}[\ell((\beta^*_{k^*(e^{\mathrm{tr}})})^{\top}\Phi^*(X^{e^{\mathrm{new}}}) + \gamma^*_{k^*(e^{\mathrm{tr}})}, Y^{e^{\mathrm{new}}})] \\
&= \mathbb{E}_{X^{e^{\mathrm{new}}}, Y^{e^{\mathrm{new}}}}[\ell((\beta^*_{k^*(e^{\mathrm{new}})})^{\top}\Phi^*(X^{e^{\mathrm{new}}}) + \gamma^*_{k^*(e^{\mathrm{new}})}, Y^{e^{\mathrm{new}}})] \\
&= (\sigma_{\mathrm{tar}}^{e^{\mathrm{new}}})^2,
\end{aligned}$$

where $(\sigma_{\mathrm{tar}}^{e^{\mathrm{new}}})^2$ is the Bayes optimal error for environment e^{new} given the optimal feature extractor Φ^*.

Here, for the second inequality, we used

$$\hat{\beta}_{\pi^{-1}(k^*(e^{\mathrm{tr}}))}^{\top}\hat{\Phi}(X^{e^{\mathrm{tr}}}) + \hat{\gamma}_{\pi^{-1}(k^*(e^{\mathrm{tr}}))} = (\beta^*_{k^*(e^{\mathrm{tr}}))})^{\top}\Phi^*(X^{e^{\mathrm{tr}}}) + \gamma^*_{k^*(e^{\mathrm{tr}})}$$

almost surely, from the first statement of Corollary 4.3 and the fact that $\pi(k(e)) = k^*(e)$ (and thus $k(e) = \pi^{-1}(k^*(e))$) for every $e \in \mathcal{E}_{\mathrm{tr}}$.

Then, we have

$$\begin{aligned}
&\mathbb{E}_{X^{e^{\mathrm{new}}}}\left\{\left((\hat{\beta}_{\hat{k}(e^{\mathrm{new}})}^{\top}\hat{\Phi}(X^{e^{\mathrm{new}}}) + \hat{\gamma}_{\hat{k}(e^{\mathrm{new}})}\right) - \left((\beta^*_{k^*(e^{\mathrm{new}}))})^{\top}\Phi^*(X^{e^{\mathrm{new}}}) + \gamma^*_{k^*(e^{\mathrm{new}})}\right)\right\}^2 \\
&= \mathbb{E}_{X^{e^{\mathrm{new}}}, \varepsilon^{e^{\mathrm{new}}}}\left\{\left((\hat{\beta}_{\hat{k}(e^{\mathrm{new}})}^{\top}\hat{\Phi}(X^{e^{\mathrm{new}}}) + \hat{\gamma}_{\hat{k}(e^{\mathrm{new}})} - Y^{e^{\mathrm{new}}}\right) + \varepsilon^{e^{\mathrm{new}}}\right\}^2 \\
&= \mathcal{L}^{e^{\mathrm{new}}}(\hat{\beta}_{\hat{k}(e^{\mathrm{new}})}, \hat{\gamma}_{\hat{k}(e^{\mathrm{new}})}, \hat{\Phi}) - (\sigma_{\mathrm{tar}}^{e^{\mathrm{new}}})^2 \\
&= 0.
\end{aligned}$$

Here, we used the fact that

$$\hat{k}(e^{\mathrm{new}}) \in \mathrm{argmin}_{k\in[K]}\mathcal{L}^{e^{\mathrm{new}}}(\hat{\beta}_k, \hat{\gamma}_k, \hat{\Phi})$$

and thus $\mathcal{L}^{e^{\mathrm{new}}}(\hat{\beta}_{\hat{k}(e^{\mathrm{new}})}, \hat{\gamma}_{\hat{k}(e^{\mathrm{new}})}, \hat{\Phi})$ must be $(\sigma_{\mathrm{tar}}^{e^{\mathrm{new}}})^2$.

Therefore, we conclude that $\hat{\beta}_{\hat{k}(e^{\mathrm{new}})}^{\top}\hat{\Phi}(X^{e^{\mathrm{new}}}) + \hat{\gamma}_{\hat{k}(e^{\mathrm{new}})} = (\beta^*_{k^*(e^{\mathrm{new}}))})^{\top}\Phi^*(X^{e^{\mathrm{new}}}) + \gamma^*_{k^*(e^{\mathrm{new}})}$ almost surely.

Then, it hold that $(A^{\top}\hat{\beta}_{\hat{k}(e^{\mathrm{new}})})^{\top}Z_{\mathrm{inv}}^{e^{\mathrm{new}}} + \hat{\gamma}_{\hat{k}(e^{\mathrm{new}})} = (\beta^*_{k^*(e^{\mathrm{new}})})^{\top}Z_{\mathrm{inv}}^{e^{\mathrm{new}}} + \gamma^*_{k^*(e^{\mathrm{new}})}$, because $\hat{\Phi}(X^{e^{\mathrm{new}}}) = AZ_{\mathrm{inv}}^{e^{\mathrm{new}}} = A\Phi^*(X^{e^{\mathrm{new}}})$. Thus, taking expectations with respect to $Z_{\mathrm{inv}}^{e^{\mathrm{new}}}$ gives $\hat{\gamma}_{\hat{k}(e^{\mathrm{new}})} = \gamma^*_{k^*(e^{\mathrm{new}})}$ because $\mathbb{E}[Z_{\mathrm{inv}}^{e}] = 0$. Also, we can see that $A^{\top}\hat{\beta}_{\hat{k}(e^{\mathrm{new}})} = \beta^*_{k^*(e^{\mathrm{new}})}$ by multiplying $(Z_{\mathrm{inv}}^{e^{\mathrm{new}}})^{\top}$ from right and taking expectations with respect to $Z_{\mathrm{inv}}^{e^{\mathrm{new}}}$. Therefore, we obtain $\hat{\beta}_{\hat{k}(e^{\mathrm{new}})}^{\top}\hat{\Phi} = (\beta^*_{k^*(e^{\mathrm{new}})})^{\top}\Phi^*$ and $\hat{\gamma}_{\hat{k}(e^{\mathrm{new}})} = \gamma^*_{k^*(e^{\mathrm{new}})}$.

This finishes all the proofs of Corollary 4.3. □

B Supplementary for Numerical Experiments

B.1 The generation procedures of the synthetic dataset

Initially, we generated $\beta_k^* \sim N(0, I_{d_{\mathrm{inv}}})$ for each cluster index $k \in [4]$ and $\Psi^* \sim N(0, I_{\mathrm{d}})$, where d_{inv} and d were set to 10 and 20 respectively. We set $\gamma_k^* = 0$ for every $k \in [4]$. Then, for each environment, we sampled $Z_{\mathrm{inv}}^e \sim N(0, (\sigma_{\mathrm{inv}}^e)^2)$ and $\varepsilon^e := \sum_{i=1}^{d_{\mathrm{sp}}}\varepsilon_i^e$ with $\varepsilon_i^e \sim N(0, (\sigma_{\mathrm{tar}}^e)^2/r)$ in an i.i.d. manner, where $d_{\mathrm{sp}} = 10$. The values σ_{inv}^e and σ_{tar}^e were independently sampled from $\mathrm{Unif}(0.5, 1.0)$. We then defined Y^e as $Y^e = (\beta^*_{k(e)})^{\top}Z_{\mathrm{inv}}^e + \varepsilon^e$. Furthermore, spurious features Z_{sp}^e were generated as $\sigma_{\mathrm{sp}}^e[\varepsilon_1^e, \ldots, \varepsilon_{d_{\mathrm{sp}}}^e]^{\top}$, where σ_{sp}^e was sampled from $\mathrm{Unif}(0.67, 1.5)$ for the training datasets of the training environments and from $\mathrm{Unif}(0.2, 5.0)$ for the test dataset of the training environments and the train/test datasets of the test environments. Finally, X^e was generated as $X^e = \Psi^*[(Z_{\mathrm{inv}}^e)^{\top}, (Z_{\mathrm{sp}}^e)^{\top}]^{\top}$.

B.2 The details of the optimization algorithms

We utilized Adam to learn model parameters, employing a learning rate of 0.001 and weight decay of 0.0001. For the clustering scores $\hat{s}^e$ of ClustJoint, we employed Momentum SGD with a learning rate of 0.1 for the synthetic dataset and 1.0 for the clustered MNIST, with a momentum of 0.9 and weight decay of 0.0. The minibatch size was set to $1,024$ suggested in Zhang et al. (2023). For fine-tuning, we used the Adam optimizer with a learning rate of 0.001 and weight decay of 0.001.

B.3 Learning curves

We provide the learning curves of the clustered MNIST experiments (Figure 5). It can be observed that the learning curves of the proposed method were not so unstable and consistently outperformed the other methods on average.

B.4 Comparisons of the mean and max operators for the IRM penalty

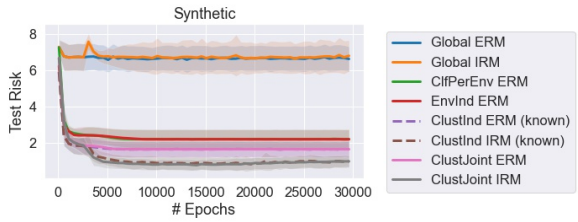
Here, we give the comparisons of the mean and max operator in the IRM penalty term for the implemented IRM algorithms in Figure 6. From these results, we can see that using the max operator led to better overall performances. This supports the decision of choosing the max operator in the IRM penalty term in Sections 3 and 5.

B.5 Clustering scores visualization

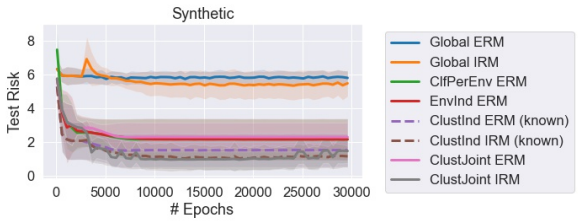
Figure 7 compares the clustering scores at the final epoch of the proposed method with and without Gumbel-softmax. While the clustering remains incomplete even with Gumbel-softmax, the results appear to be better than those with the naive softmax. This improvement is reflected in the higher clustering accuracy shown in Figure 4.

B.6 Computing Infrastructures

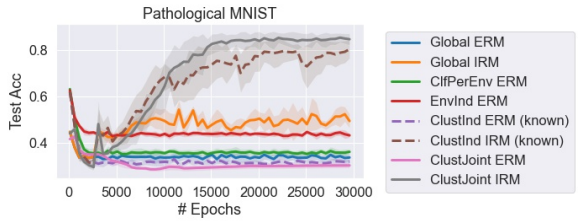
- OS: Ubuntu 16.04.6
- CPU: AMD EPYC 7552 48-Core Processor.
- CPU Memory: 1.0 TB.
- Programming language: Python 3.9.12.
- Deep learning framework: Pytorch 1.12.1.



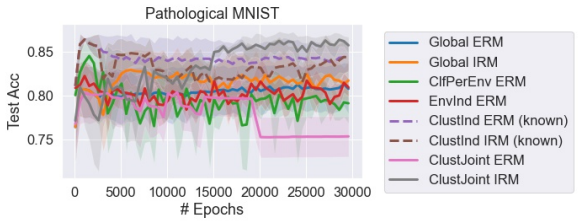
(a) Synthetic (Zero-shot on train envs.)



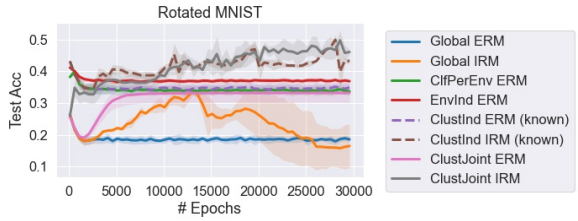
(b) Synthetic (Few-shot on test envs.)



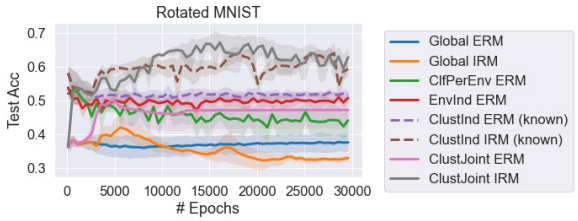
(c) Pathological (Zero-shot on train envs.)



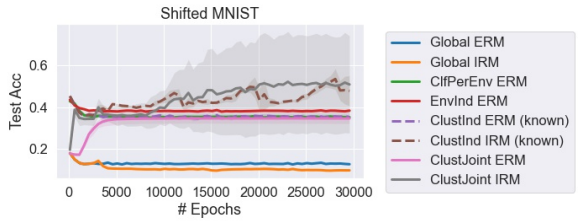
(d) Pathological (Few-shot on test envs.)



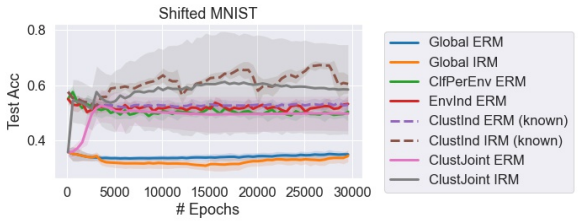
(e) Rotated (Zero-shot on train envs.)



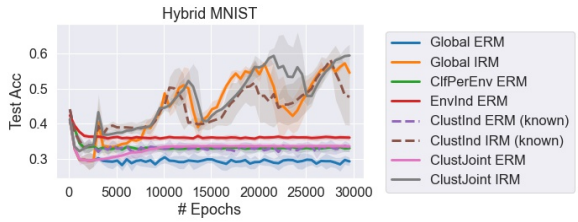
(f) Rotated (Few-shot on test envs.)



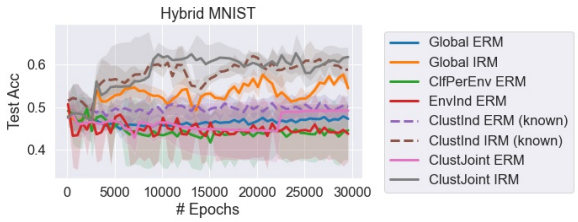
(g) Shifted (Zero-shot on train envs.)



(h) Shifted (Few-shot on test envs.)

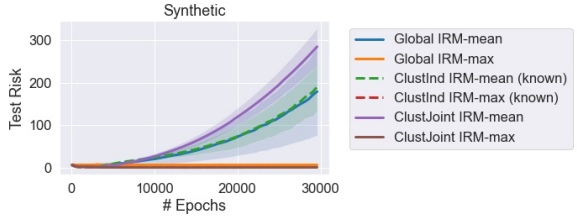


(i) Hybrid (Zero-shot on train envs.)

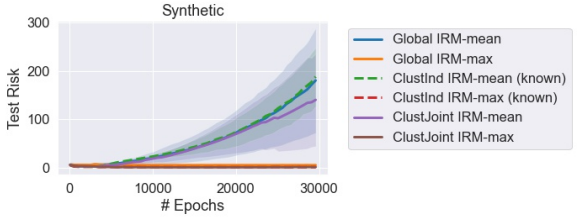


(j) Hybrid (Few-shot on test envs.)

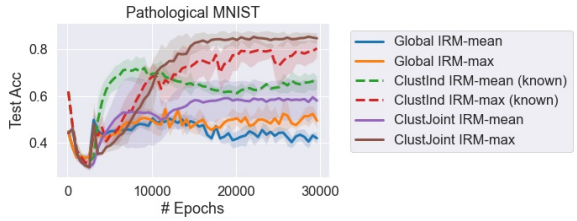
Figure 5: Comparison of test accuracy on the five datasets for zero-shot learning on training environments and few-shot learning on test environments (higher is better). “(known)" indicates that the algorithm used the true clustered structure of the environments.



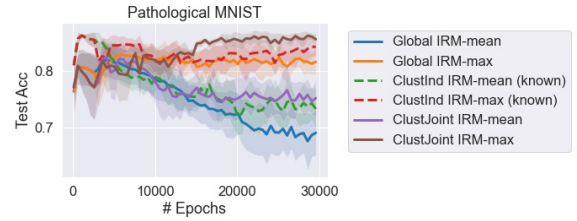
(a) Synthetic (Zero-Shot on train envs.)



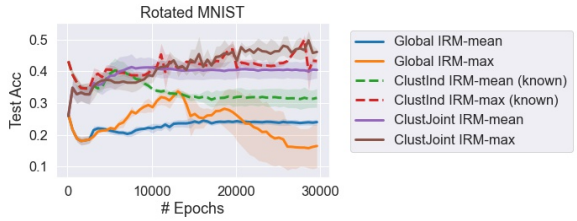
(b) Synthetic (Few-shot on test envs.)



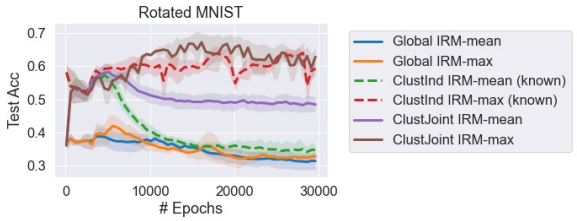
(c) Pathological (Zero-Shot on train envs.)



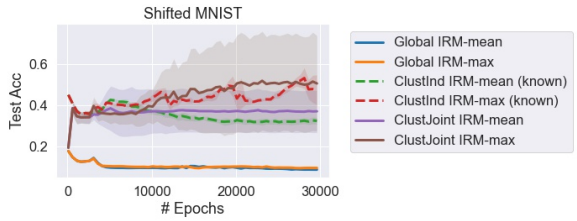
(d) Pathological (Few-shot on test envs.)



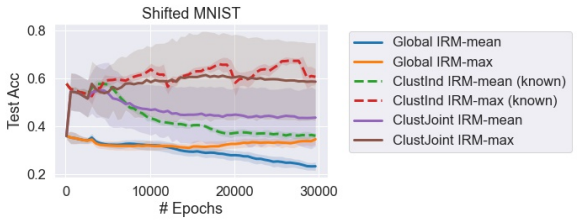
(e) Rotated (Zero-Shot on train envs.)



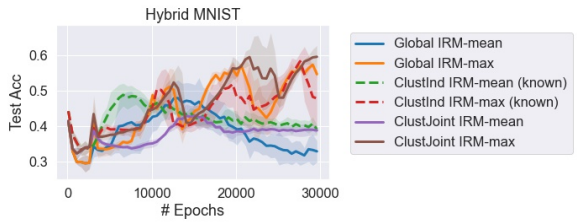
(f) Rotated (Few-shot on test envs.)



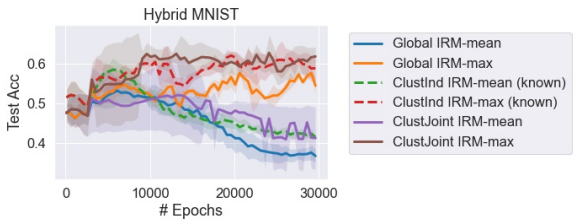
(g) Shifted (Zero-Shot on train envs.)



(h) Shifted (Few-shot on test envs.)

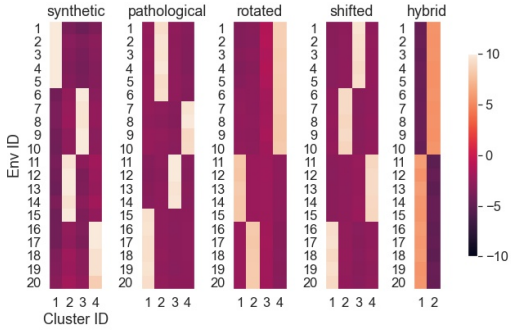


(i) Hybrid (Zero-Shot on train envs.)

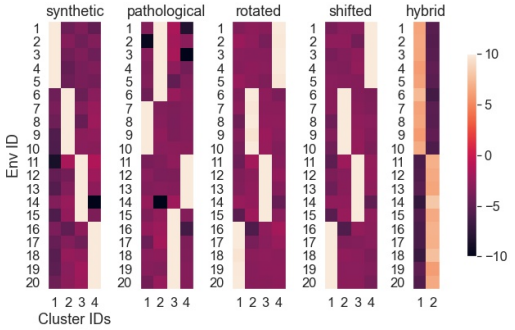


(j) Hybrid (Few-shot on test envs.)

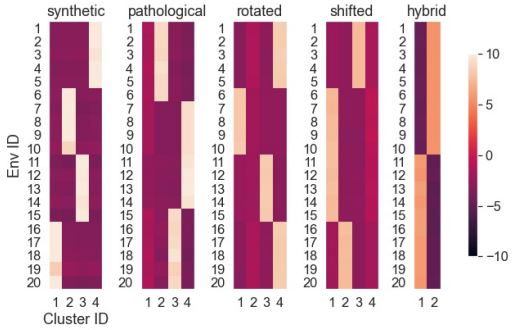
Figure 6: Comparison of test MSE and accuracy on Synthetic and Clustered MNIST datasets respectively for zero-shot learning on training environments and few-shot learning on test environments. “(known)" indicates that the algorithm used the true clustered structure of the environments. “-mean" indicates that the algorithm used the mean operator in the IRM penalty term and “-max" indicates that the algorithm used the max operator in the IRM penalty term.



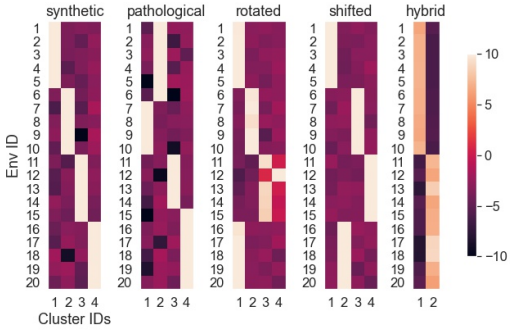
(a) With Gumbel-softmax (random seed 0)



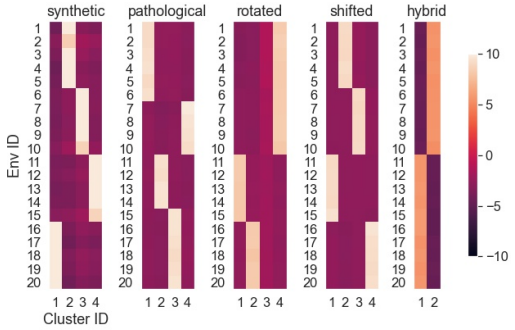
(b) Without Gumbel-softmax (random seed 0)



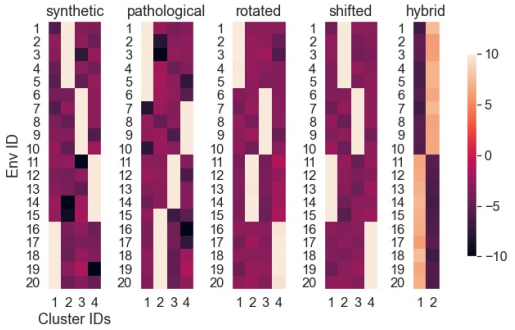
(c) With Gumbel-softmax (random seed 1)



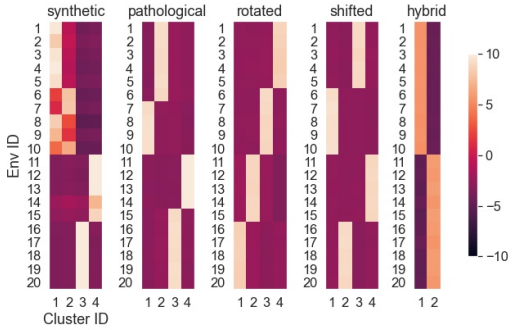
(d) Without Gumbel-softmax (random seed 1)



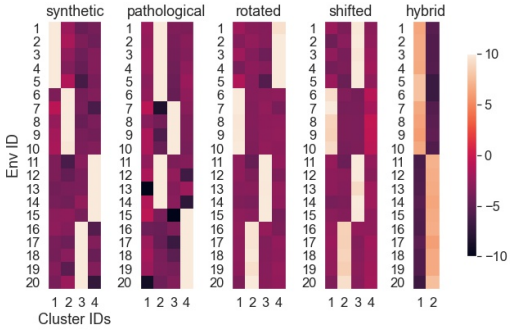
(e) With Gumbel-softmax (random seed 2)



(f) Without Gumbel-softmax (random seed 2)



(g) With Gumbel-softmax (random seed 3)



(h) Without Gumbel-softmax (random seed 3)

Figure 7: The clustering scores $\{\hat{s}_e\}_{e\in\mathcal{E}_{\mathrm{tr}}}$ (each $\hat{s}_e \in \mathbb{R}^4$) at the final epoch of the proposed method on four random seeds. The environments IDs are sorted by the true cluster IDs. **Left**: *without* the Gumbel-softmax function. **Right**: *with* the Gumbel-softmax function.