Enhancing Distributional Stability among Sub-populations

Jiashuo Liu[1,*], **Jiayun Wu**[1], **Jie Peng**[1], **Xiaoyu Wu**[2], **Yang Zheng**[2], **Bo Li**[3], **Peng Cui**[1,†]
[1]Department of Computer Science and Technology, Tsinghua University
[2]RAMS Lab, Huawei Technologies Co Ltd
[3]School of Economics and Management, Tsinghua University
[*]liujiashuo77@gmail.com, [†]cuip@tsinghua.edu.cn

Abstract

Enhancing the stability of machine learning algorithms under distributional shifts is at the heart of the Out-of-Distribution (OOD) Generalization problem. Derived from causal learning, recent works of invariant learning pursue strict invariance with multiple training environments. Although intuitively reasonable, strong assumptions on the availability and quality of environments are made to learn the strict invariance property. In this work, we come up with the "distributional stability" notion to mitigate such limitations. It quantifies the stability of prediction mechanisms among sub-populations down to a prescribed scale. Based on this, we propose the learnability assumption and derive the generalization error bound under distribution shifts. Inspired by theoretical analyses, we propose our novel stable risk minimization (SRM) algorithm to enhance the model's stability w.r.t. shifts in prediction mechanisms ($Y|X$-shifts). Experimental results are consistent with our intuition and validate the effectiveness of our algorithm. The code can be found at https://github.com/LJSthu/SRM.

1 INTRODUCTION

Traditional machine learning algorithms with empirical risk minimization (ERM) are vulnerable when exposed to data drawn out of the training distribution. In order to mitigate the failures in the out-of-distribution (OOD) generalization, invariant learning methods (Arjovsky et al., 2019; Ahuja et al., 2020, 2021; Peters et al., 2016) are proposed to learn prediction mechanisms that are strictly invariant across given multiple environments. Such strict invariance property enables models to generalize under distributional shifts (Peters et al., 2016; Rojas-Carulla et al., 2018; Arjovsky et al., 2019; Koyama et al., 2020).

To fulfill the promise of invariant learning, *environment labels* have to be provided to achieve strict invariance. Moreover, the concept of strict invariance even assumes the access of *all possible environments*. However, such requirement is unrealistic in real-world applications where modern datasets are often constructed by amalgamating data from various sources, thus significantly limiting the applicability of invariant learning techniques. Recent efforts, such as EIIL (Creager et al., 2021) and HRM (Liu et al., 2021a,b), have focused on generating pseudo environment labels to facilitate invariant learning. Nonetheless, the characteristics of these pseudo environments, the extent of invariance they enable, and even the validity of the problem framework itself, remain unclear and inadequately justified.

To address these limitations, our research shifts focus towards developing models that generalize out-of-distribution within contexts of latent heterogeneity, where the training data is gathered from multiple sources, but lacking explicit source labels. In this setting, the training data exhibits sub-population structures, with probably distinct prediction mechanisms varying across sub-populations. To tackle this problem, we introduce an approach that extends strict invariance to the concept of "distributional stability". This metric assesses the consistency of prediction mechanisms across sub-populations. Unlike the binary nature of strict invariance, which is either yes or no, distributional stability provides a continuous measure that quantifies the degree of predictive mechanism stability across varying contexts. This nuanced approach allows for a more refined assessment of model robust-

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

ness in handling distribution shifts.

In Section 2, we formally define the distributional stability, and introduce its properties as well as relationships with strict invariance. And we also demonstrate its relationship with the distributional robustness from the distributionally robust optimization literature (Duchi et al., 2018, 2019). Then in Section 3, we characterize the *learnability* of the problem to rationalize the problem setting itself and clarify what kind of target distributions could be generalized to. Then we derive the *OOD generalization error bound* for this problem based on the distributional stability. Inspired by the theoretical results, we find that models with strong distributional stability could generalize well with respect to shifts on prediction mechanisms ($Y|X$-shifts). Thus, we propose an empirical algorithm named *Stable Risk Minimization* (SRM) in Section 4, and experimental results on both simulation and real-world data validate the effectiveness of our method.

Notations Throughout this paper, we let $X \in \mathcal{X}$ denote the covariates, $Y \in \mathcal{Y}$ denote the target. $f_\theta(\cdot) : \mathcal{X} \to \mathcal{Y}$ is the predictor parameterized by $\theta \in \Theta$. $\mathcal{E}$ is the random variable taking values in all possible environments. The random variable of data points is denoted by $Z = (X, Y) \in \mathcal{Z}$. $\mathbb{P}^e(Z)$ abbreviated with $\mathbb{P}^e$ denotes the joint distribution in environment e, and for environments $e_1, e_2 \in \text{supp}(\mathcal{E})$, the data distribution can be quite different. $\mathbb{P}_{\text{train}}(Z)$ and $\mathbb{P}_{\text{test}}(Z)$ abbreviated with $\mathbb{P}_{\text{tr}}$ and $\mathbb{P}_{\text{te}}$ respectively represent the joint training distribution and test distribution. Denote the feature extractor $\Phi_\theta(X)$ parameterized by θ, and the predicting function $\hat{Y} = h_\eta(\Phi_\theta(X))$ parameterized by η (not restricted to linear $h(\cdot)$), which gives the whole prediction model $f_{\eta,\theta}(X) = h_\eta(\Phi_\theta(X))$. For simplicity, we omit the subscripts θ, η without causing misunderstanding. Denote the sample size by n and the vector of sample weights $\mathbf{w} \in \mathbb{R}^n_+ = [w_1, \ldots, w_n]^T$ with $\mathbf{w} \geq 0$ and $\mathbf{w}^T\mathbf{1} = 1$.

2 DISTRIBUTIONAL STABILITY

In this section, we first introduce the *strict invariance* property as well as its limitations. Then we propose the *distributional stability* property, a relaxed alternative under latent heterogeneity.

2.1 Strict invariance

Inspired by causal inference literature, strict invariance (Arjovsky et al., 2019; Ahuja et al., 2020; Koyama et al., 2020; Creager et al., 2021; Liu et al., 2021a,b) requires that the prediction mechanism $Y|X$ remains the same among environments, which has two typical formulations.

Definition 1 (Strict Invariance)**.** *Denote the random variable taking values of all possible environments as $\mathcal{E}$. A representation Φ is strictly invariant if condition 1 or condition 2 holds:*
Condition 1 (Arjovsky et al., 2019; Ahuja et al., 2020; Creager et al., 2021): for any $e_1, e_2 \in supp(\mathcal{E})$,

$$\mathbb{E}[Y|\Phi, \mathcal{E} = e_1] = \mathbb{E}[Y|\Phi, \mathcal{E} = e_2]. \tag{1}$$

Condition 2 (Koyama et al., 2020; Liu et al., 2021a,b): for any $e_1, e_2 \in supp(\mathcal{E})$,

$$\mathbb{P}(Y|\Phi, \mathcal{E} = e_1) = \mathbb{P}(Y|\Phi, \mathcal{E} = e_2). \tag{2}$$

Invariant learning methods use strict invariance as a constraint during the model learning procedure. Arjovsky et al. (2019) prove that a linear model only uses invariant features under condition (1), and Koyama et al. (2020) prove the resultant model under condition (2) is optimal for OOD generalization. Despite the promising theoretical results, one major concern in defining the strict invariance as Definition 1 is the access to *all possible environments* $\mathcal{E}$.

The strict invariance requires all possible environments to examine whether the prediction mechanism $Y|\Phi$ stays invariant. However, in most of the real-world applications, it is impossible to acquire all possible environments, which renders the goal of strict invariance unrealistic to reach in practice. As a result, the learned invariance only holds for the finite training environments, but *whether* it is violated in other agnostic environments and *how much* it is violated remain entirely unknown for machine learning engineers and system users, which brings huge risks in high-stakes applications.

2.2 A relaxed alternative

To mitigate the limitations above, we relax the requirements for multiple environments and instead consider an elaborated setting where the observed data are heterogeneous. More precisely, following Duchi et al. (2019), we assume that

$$X, Y \sim \mathbb{P}_{\text{tr}} := \alpha\mathbb{Q}_0 + (1 - \alpha)\mathbb{Q}_1$$

where the proportion $\alpha \in (0, 1)$ and $\mathbb{Q}_0, \mathbb{Q}_1$ denote the sub-populations in $\mathbb{P}_{\text{tr}}$. Since the sub-population distributions are not pre-defined, it is termed latent heterogeneity. To measure the stability of a machine learning model under potential distributional shifts, inspired by strict invariance among given environments (i.e. explicit heterogeneity), we could examine whether the predicting mechanism holds among all potential sub-populations within $\mathbb{P}_{\text{tr}}$. First, we define the sub-population set for a distribution in Definition 2.

Definition 2 (Sub-population set). *Given distribution $\mathbb{P}(Z)$, for $\alpha_0 \in (0, 1/2)$ as a lower bound on the sub-population proportion α, the set of sub-populations of distribution $\mathbb{P}$ is*

$$\mathcal{P}_{\alpha_0}(\mathbb{P}) := \{\mathbb{Q}_0 : \mathbb{P} = \alpha\mathbb{Q}_0 + (1-\alpha)\mathbb{Q}_1, \text{ for some } \alpha \in [\alpha_0, 1) \text{ and distribution } \mathbb{Q}_1 \text{ on } \mathcal{Z}\}.$$

Remark. *Intuitively, $\mathcal{P}_{\alpha_0}(\mathbb{P})$ contains all sub-populations of $\mathbb{P}$ with proportion $\alpha \geq \alpha_0$. α_0 controls the size of the minimal sub-populations considered, i.e. smaller α_0 corresponds with smaller sub-populations but larger size of the set ($|\mathcal{P}_{\alpha_0}|$).*

Based on this, we introduce a nuanced variant of strict invariance, named α_0-distributional stability. This concept serves to quantify the level of stability in the face of shifts among sub-populations. It represents a more flexible approach that captures the degree to which a model's predictions remain consistent across varying sub-populations, offering a refined metric for evaluating the robustness of models in heterogeneous data environments.

Definition 3 (α_0-distributional stability). *Given data distribution $\mathbb{P}(Z)$, for $\alpha_0 \in (0, 1/2)$, the α_0-distributional stability of the prediction mechanism $Y|X$ is defined as*

$$DS_{\alpha_0}(Y|X;\mathbb{P}) := \sup_{\mathbb{Q}\in\mathcal{P}_{\alpha_0}(\mathbb{P})} \rho_{KL}(\mathbb{Q}(Y|X), \mathbb{P}(Y|X)) \tag{3}$$

where $\rho_{KL}(\cdot,\cdot)$ denotes the KL-divergence between two distributions.

Remark. *Intuitively, α_0-distributional stability measures the* maximal variation *of the prediction mechanism $(Y|X)$ among sub-populations within $\mathbb{P}$ in terms of KL-divergence. It picks the worst sub-population $\mathbb{Q}^\star$ in the set $\mathcal{P}_{\alpha_0}(\mathbb{P})$ and calculates the KL-divergence between $\mathbb{Q}^\star(Y|X)$ and $\mathbb{P}(Y|X)$. The smaller the DS_{α_0} is, the more stable the prediction mechanism $Y|X$ is, since one can hardly find a sub-population that violates $\mathbb{P}(Y|X)$.*

Then we demonstrate some properties of the proposed α_0-distributional stability.

Proposition 1 (Properties of $\mathrm{DS}_{\alpha_0}(\mathbb{P})$). *For observed data distribution $\mathbb{P}(Z)$ and $\alpha_0 \in (0, 1/2)$, we have*

1. Nonnegativity: $DS_{\alpha_0}(Y|X;\mathbb{P}) \geq 0$;

2. Monotonicity: if $\alpha_1 \geq \alpha_2$, we have $DS_{\alpha_1}(Y|X;\mathbb{P}) \leq DS_{\alpha_2}(Y|X;\mathbb{P})$

Remark. *The smaller α_0 is, the larger distribution set $\mathcal{P}_{\alpha_0}(\mathbb{P})$ is, and the larger the stability criterion is, since the mechanism $Y|X$ is examined under more fine-grained sub-populations.*

Proposition 2 (Relationship with strict invariance). *Here we demonstrate the connections and differences between α_0-distributional stability and strict invariance:*

1. Connection with condition (1): replace $\rho_{KL}(\cdot,\cdot)$ with $\mathbb{E}[\|\mathbb{E}_{\mathbb{Q}}[Y|X] - \mathbb{E}_{\mathbb{P}}[Y|X]\|^2]$, and replace the sub-population set $\mathcal{P}_{\alpha_0}(\mathbb{P})$ with $\mathcal{E}$, then we have: $DS_{\alpha_0}(Y|X;\mathbb{P}) = 0$ is equivalent to condition (1).

2. Connection with condition (2): replace the sub-population set $\mathcal{P}_{\alpha_0}(\mathbb{P})$ with $\mathcal{E}$, then we have: $DS_{\alpha_0}(Y|X;\mathbb{P}) = 0$ is equivalent to condition (2).

Remark (Connection with distributional robustness). *Although both terms involve the sub-population set, distributional stability and distributional robustness are inherently different from each other. Distributional robustness (Duchi et al., 2018; Sinha et al., 2018; Duchi et al., 2019) refers to the* worst-case performance *inside the pre-defined uncertainty set $\mathcal{P}$, while distributional stability measures the* maximal variation *of the prediction mechanism $Y|X$. Therefore, distributional robustness reflects the performance at a single point (i.e. the worst-case distribution), but distributional stability measures the variation of the prediction mechanisms (i.e. contrast between two distributions). Such difference leads to a huge discrepancy in the guarantees of the OOD generalization performances. DRO methods to obtain distributional robustness could* only *ensure the performance within the distribution set $\mathcal{P}$, while methods to pursue distributional stability could* generalize to agnostic testing distributions *under the* learnability assumption*, which will be discussed in detail in Section 3.*

3 THEORETICAL ANALYSIS

Based on *distributional stability*, we formally define the OOD generalization problem under latent heterogeneity. Then we provide theoretical analysis of this problem, including the learnability assumption and the generalization error bound.

Problem 1 (Setup). *Given data $Z \sim \mathbb{P}_{tr}(Z)$ collected from **multiple agnostic sources**, the goal is to learn models with good generalization performances on data from agnostic target distribution $\mathbb{P}_{te}(Z)$.*

For traditional machine learning problems, the analysis of the learnability is based on the *i.i.d.* assumption. However, in Problem 1, the target distribution is agnostic and could significantly differ from the training one. Therefore, without any further assumptions, even the learnability itself can *hardly* hold in general. Given this, we characterize the learnability assumption of Problem 1, which makes assumptions on the target distribution. Following Ye et al. (2021), we define the

expansion function as follows:

Definition 4 (Expansion Function). *A function $s : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}^+ \cup \{0, +\infty\}$ is an expansion function, iff the following properties hold: (1) $s(\cdot)$ is monotonically increasing and $s(x) \geq x, \forall x$; (2) $\lim_{x \to 0^+} s(x) = s(0) = 0$.*

Besides, for training distribution $\mathbb{P}_{\text{tr}}(Z)$ and target distribution $\mathbb{P}_{\text{te}}(Z)$, we define the *out-of-distribution stability* of $Y|X$ as:

$$\text{ODS}(Y|X; \mathbb{P}_{\text{tr}}, \mathbb{P}_{\text{te}}) \coloneqq \rho_{\text{KL}}(\mathbb{P}_{\text{te}}(Y|X), \mathbb{P}_{\text{tr}}(Y|X)),$$

which measures the stability of the prediction mechanism between $\mathbb{P}_{\text{tr}}$ and $\mathbb{P}_{\text{te}}$. Note that here $\mathbb{P}_{\text{te}}$ denotes the target distribution, which may not be included in the pre-defined sub-population set.

Then we formally provide the learnability assumption of Problem 1.

Assumption 1 (Learnability of Problem 1). *Problem 1 from $\mathbb{P}_{tr}$ to $\mathbb{P}_{te}$ is (α_0, s)-learnable if there exists an expansion function $s(\cdot)$ such that $ODS(Y|X; \mathbb{P}_{tr}, \mathbb{P}_{te}) \leq s(DS_{\alpha_0}(Y|X; \mathbb{P}_{tr}))$.*

Note that here X could be replaced by some representations $\Phi(X)$. Here we make some remarks.

Remark. ***(1)*** *Assumption 1 assumes that the α_0-distributional stability measure of the training distribution should* approximately hold *in testing, that is, its variation on the target distribution is upper bounded by the expansion function. Intuitively, it requires that the conditional distribution $\mathbb{P}_{te}(Y|X)$ cannot arbitrarily change. If $\mathbb{P}_{te}(Y|X)$ could arbitrarily change, the problem is unlearnable, since the prediction mechanism learned in training may not hold in testing.*
(2) *The steepness of the expansion function reflects the* difficulty *of Problem 1, since the steeper the expansion function is, the less likely the learned distributional stability will hold in testing. As shown in Theorem 1, the expansion function influences the generalization error bound.*

We then derive the OOD generalization bound for Problem 1.

Theorem 1 (Generalization Bound). *Under Assumption 1, assume that $\ell(\cdot, \cdot)$ is upper bounded, the conditional generalization error gap could be bounded by the distributional stability as:*

$$\mathbb{E}_{\mathbb{P}_{te}}\left[\left\|\mathbb{E}_{\mathbb{P}_{te}}[\ell(X,Y)|X] - \mathbb{E}_{\mathbb{P}_{tr}}[\ell(X,Y)|X]\right\|\right] \leq \mathcal{O}(\sqrt{1 - e^{-s(DS_{\alpha_0}(Y|X;\mathbb{P}_{tr}))}}), \tag{4}$$

where $\ell(\cdot, \cdot)$ denotes the loss function.

In Theorem 1, we calculate the *conditional* error gap bound, which excludes covariate shifts by aligning the covariate distribution with $\mathbb{P}_{\text{te}}(X)$. From Equation (4), we can see that controlling the distributional stability $\text{DS}_{\alpha_0}(Y|X; \mathbb{P}_{\text{tr}})$ could decrease the generalization error gap between training and testing. The theoretical results motivate our Stable Risk Minimization (SRM) algorithm in Section 4.

4 METHOD

To enhance the distributional stability, inspired by Theorem 1, we propose our **Stable Risk Minimization** (SRM) algorithm based on the newly-proposed distributional stability. We first introduce the overall objective function, and then derive an approximated optimization method for classification and regression.

Objective function.

To learn models with good distributional stability, we introduce the stability constraints to the general risk minimization and propose our stable risk minimization framework as:

$$\begin{aligned} \theta^*, \eta^* = \arg\min_{\theta, \eta} & \mathbb{E}_{X,Y \sim \mathbb{P}_{\text{tr}}}\left[\ell(h_\eta(\Phi_\theta(X)), Y)\right] \\ \text{s.t.} \quad & \text{DS}_{\alpha_0}(Y|\Phi_{\theta^*}(X); \mathbb{P}_{\text{tr}}) \leq \delta \end{aligned} \tag{5}$$

where α_0 is the pre-defined lower-bound on the sub-population proportion, and $\delta \geq 0$ is the threshold of distributional stability of the prediction mechanism $Y|\Phi_{\theta^*}(X)$. The constraint could help to learn representation $\Phi_{\theta^*}(X)$ that is stable among sub-populations within $\mathbb{P}_{\text{tr}}$.

Following the approximation techniques typically adopted in robust learning (Arjovsky et al., 2019; Sinha et al., 2018), we give up the requirement of a prescribed constraint δ of distributional stability, and instead focus on the Lagrangian penalty problem, which also corresponds with our theoretical results in Theorem 1.

$$\min_{\theta, \eta} \mathbb{E}_{\mathbb{P}_{\text{tr}}}[\ell(h_\eta(\Phi_\theta(X)), Y)] + \lambda \cdot \text{DS}_{\alpha_0}(Y|\Phi_\theta(X); \mathbb{P}_{\text{tr}}) \tag{6}$$

The *key challenge* lies in the calculation of the distributional stability constraint $\text{DS}_{\alpha_0}(Y|\Phi_\theta(X); \mathbb{P}_{\text{tr}})$. Recall that it relies on the worst sub-population $\mathbb{Q}^\star$ in Equation (3). Therefore, the optimization involves a *two-player* game, where a *variation exploiter* keeps picking the worst sub-population $\mathbb{Q}^\star$ from $\mathcal{P}_{\alpha_0}(\mathbb{P}_{\text{tr}})$, and a *stable learner* learns a more stable representation with smaller discrepancy between $\mathbb{Q}^\star(Y|\Phi_\theta(X))$ and $\mathbb{P}_{\text{tr}}(Y|\Phi_\theta(X))$.

Algorithm 1 Stable risk minimization (SRM)

Input: Training Data $D = \{x_i, y_i\}_{i=1}^n$, hyper-parameter λ, epoch number T, prescribed sub-population ratio α_0.
Initialize: $\Phi^{(1)} = X$
for $t = 1$ **to** T **do**
Step 1. Variation explorer: Given $\Phi^{(t)}$, find the worst sub-population $\mathbb{Q}^\star(t)$ characterized by $\mathbf{w}^*$ according to Equation (8).
Step 2. Stable learner: Given the learned worst sub-population $\mathbb{Q}^\star(t)$, perform stable risk minimization on $\{\hat{\mathbb{P}}_{\text{tr}}, \mathbb{Q}^\star(t)\}$ according to Equation (10) to obtain the representation $\Phi^{(t+1)}$.
end for

4.1 Player 1: variation explorer

Given current representation $\Phi_\theta(X)$ (*abbr.* Φ), the α_0-distributional stability takes the form of:

$$\text{DS}_{\alpha_0}(Y|\Phi; \mathbb{P}_{\text{tr}}) = \sup_{\mathbb{Q}\in\mathcal{P}_{\alpha_0}(\mathbb{P}_{\text{tr}})} \mathbb{E}_{\mathbb{Q}}\left[\log \frac{\mathbb{Q}(Y|\Phi)}{\mathbb{P}_{\text{tr}}(Y|\Phi)}\right].$$

The goal of the variation explorer is to find the sub-population $\mathbb{Q}^\star$ that:

$$\mathbb{Q}^\star = \arg \sup_{\mathbb{Q}\in\mathcal{P}_{\alpha_0}(\mathbb{P}_{\text{tr}})} \mathbb{E}_{\mathbb{Q}}\left[\log \frac{\mathbb{Q}(Y|\Phi)}{\mathbb{P}_{\text{tr}}(Y|\Phi)}\right]. \tag{7}$$

For different kinds of tasks, we propose different ways to approximate Equation (7) in the following.

(1) For regression tasks. Given the representation $\Phi \in \Upsilon$ and the label $Y \in \mathbb{R}$, we parameterize the conditional distribution $\mathbb{P}_{\text{tr}}(Y|\Phi)$ and $\mathbb{Q}(Y|X)$ as:

$$\mathbb{P}_{\text{tr}}(Y|\Phi) \approx \mathcal{N}(f_{\text{tr}}(\Phi), \sigma_{\text{tr}}^2),$$
$$\mathbb{Q}(Y|\Phi) \approx \mathcal{N}(f_{\text{q}}(\Phi), \sigma_{\text{q}}^2),$$

where $f_{\text{tr}} = \mathbb{E}_{\mathbb{P}_{\text{tr}}}[Y|\Phi]$ and $f_{\text{q}} = \mathbb{E}_{\mathbb{Q}}[Y|\Phi]$ denote the prediction functions tailored to fit the data distributions $\mathbb{P}_{\text{tr}}$ and $\mathbb{Q}$, respectively. σ_{tr}, σ_q are noise scale parameters. Based on this approximation, Equation (7) for *regression* tasks becomes:

$$\mathbb{Q}^\star = \arg \sup_{\mathbb{Q}\in\mathcal{P}_{\alpha_0}(\mathbb{P}_{\text{tr}})} \mathbb{E}_{\mathbb{Q}}\left[\frac{(Y - f_{\text{tr}}(\Phi))^2}{\sigma_{\text{tr}}^2} - \frac{(Y - f_{\text{q}}(\Phi))^2}{\sigma_{\text{q}}^2}\right].$$

(2) For classification tasks.. Denote the number of classes by K, the conditional distribution is discrete and can be modeled via a K-dimensional simplex. Given the representation $\Phi \in \Upsilon$ and target variable $Y \in [K]$, $P_{tr}(Y|\Phi)$ and $Q(Y|\Phi)$ are modeled as:

$$\mathbb{P}_{\text{tr}}(Y|\Phi) \approx f_{\text{tr}}(\Phi) \in \Delta_K,$$
$$\mathbb{Q}(Y|\Phi) \approx f_{\text{q}}(\Phi) \in \Delta_K,$$

where $f_{\text{tr}}, f_{\text{q}}$ denote the prediction models that fit the data from distribution $\mathbb{P}_{\text{tr}}$ and $\mathbb{Q}$ respectively. Then Equation (7) for *classification* tasks becomes:

$$\mathbb{Q}^\star = \arg \sup_{\mathbb{Q}\in\mathcal{P}_{\alpha_0}(\mathbb{P}_{\text{tr}})} \mathbb{E}_{\mathbb{Q}}\left[\log \frac{f_{\text{q}}(\Phi)[Y]}{f_{\text{tr}}(\Phi)[Y]}\right].$$

where $f_q(\Phi)[Y]$ denotes the value of Y-th dimension of $f_{\text{q}}(\Phi) \in \Delta_K$, and the same for $f_{\text{tr}}(\Phi)[Y]$.

Now we are ready to derive the empirical objective function from Equation (7) for both regression and classification tasks. Empirically, given dataset $D = \{x_i, y_i\}_{i=1}^n$ drawn from $\mathbb{P}_{\text{tr}}$, $\hat{\mathbb{P}}_{\text{tr}}$ can be represented by

$$\hat{\mathbb{P}}_{\text{tr}} = \frac{1}{n}\sum_{i=1}^n \delta_{(x_i,y_i)},$$

where $\delta_{(x,y)}$ denotes the Dirac distribution that is supported on (x, y). Similarly, the sub-population set $\mathcal{P}_{\alpha_0}(\hat{\mathbb{P}}_{\text{tr}})$ can be modeled as:

$$\mathcal{P}_{\alpha_0}(\hat{\mathbb{P}}_{tr}) = \{\mathbf{w} = [w_1, \ldots, w_n]^T : \mathbf{w} \in \Delta_n, \mathbf{w} \le \frac{1}{\alpha_0 n}\},$$

where the sub-population $Q \in \mathcal{P}_{\alpha_0}(\hat{\mathbb{P}}_{tr})$ is characterized by sample weights, and w_i denotes the weight of the i-th sample. Then Equation (7) can be reformulated as:

$$\begin{aligned} &\mathbf{w}^\star = \arg \max_{\mathbf{w}\in\mathcal{P}_{\alpha_0}(\hat{\mathbb{P}}_{tr})} \sum_{i=1}^n w_i \cdot \boxed{g_i}\ , \\ &\text{s.t.} \quad \mathbf{w} \in \Delta_n \text{ and } \mathbf{w} \le \frac{1}{\alpha_0 n}, \end{aligned} \tag{8}$$

where $\boxed{g_i}$ depends on the task type (regression or classification):

$$\boxed{g_i} := \begin{cases} \frac{(y_i - f_{\text{tr}}(\phi_i))^2}{\sigma_{\text{tr}}^2} - \frac{(y_i - f_{\text{q}}(\phi_i))^2}{\sigma_{\text{q}}^2}, & \text{for regression} \\ \log \frac{f_{\text{q}}(\phi_i)[y_i]}{f_{\text{tr}}(\phi)[y_i]}, & \text{for classification} \end{cases}$$

where $\phi_i = \Phi(x_i)$. To estimate $f_{\text{tr}}, \sigma_{\text{tr}}, f_{\text{q}}, \sigma_{\text{q}}$, through maximal likelihood estimation, we have:

$$\begin{aligned} f_{\text{tr}} &= \arg\min_f \sum_{i=1}^n \ell(f(\phi_i), y_i), \\ f_{\text{q}} &= \arg\min_f \sum_{i=1}^n w_i \ell(f(\phi_i), y_i), \\ \sigma_{\text{tr}}^2 &= \mathbb{E}_{\mathbb{P}_{\text{tr}}}[\ell^2(f_{\text{tr}}(\Phi), Y)] - (\mathbb{E}_{\mathbb{P}_{\text{tr}}}[\ell(f_{\text{tr}}(\Phi), Y)])^2, \\ \sigma_{\text{q}}^2 &= \mathbb{E}_{\mathbb{Q}}[\ell^2(f_{\text{q}}(\Phi), Y)] - (\mathbb{E}_{\mathbb{Q}}[\ell(f_{\text{q}}(\Phi), Y)])^2. \end{aligned}$$

Notably, for f_{tr}, since it fits the empirical training distribution $\hat{\mathbb{P}}_{\text{tr}}$ and is not affected by sample weights, we only train it once and fix it. For f_{q}, one could use bi-level optimization to jointly optimize the sample

weights $\mathbf{w}$ and f_{q}. In this work, we find that adopting an iterative training process yields impressive results in practical, and therefore we did not implement bi-level optimization here. But we refer the readers interested in the bi-level optimization of Equation (8) to (Shu et al., 2019; Shaban et al., 2019).

Complexity Analysis Here we analyze the complexity of the variation exploitation stage. *First*, this stage is based on the *representation* $\Phi(X)$ of the input data X. Therefore, the conditional distribution $\mathbb{P}(Y|\Phi(X))$ is easy to fit empirically and typically is chosen as *linear model*, which could be viewed as the last layer of a deep neural network. *Second*, we analyze the additional computation cost, and show that it is similar to *adversarial training*. Denote the sample size as N, dimension of Φ as d_ϕ, and the training epoch as T, the additional cost is $\mathcal{O}(Nd_\phi T)$. Notably, since f_{q} is linear, the convergence is quick, and we set it to 50 in our experiments. To demonstrate that this computation cost is acceptable, we further analyze the additional cost of adversarial training for comparison. Denote the overall number of parameters as D, the attack step as T_a, the additional cost of adversarial training is $\mathcal{O}(NDT_a)$ with $D \gg d_\phi$. Therefore, *the additional computation cost of our method is lower (or no larger) than adversarial training, which is acceptable. Third*, to further lower the computation burden, we perform the variation exploitation stage once every K epochs, and K is set to 20 in our experiments. Therefore, the additional time complexity further reduces to $\mathcal{O}(Nd_\phi T/K)$.

4.2 Player 2: stable learner

Given the worst sub-population $\mathbb{Q}^\star$ in Equation (7), the distributional stability could be simplified to:

$$\mathrm{DS}_{\alpha_0}(Y|\Phi;\mathbb{P}_{\mathrm{tr}}) = \rho_{\mathrm{KL}}(\mathbb{Q}^\star(Y|\Phi)\|\mathbb{P}_{\mathrm{tr}}(Y|\Phi)).$$

Therefore, for the *stable learner* (player 2), the Lagrangian penalty problem in Equation (6) becomes:

$$\begin{aligned}\mathcal{L}(\theta,\eta) =& \mathbb{E}_{\mathbb{P}_{\mathrm{tr}}}[\ell(h_\eta(\Phi_\theta(X)),Y)]+ \\ & \lambda\rho_{\mathrm{KL}}(\mathbb{Q}^\star(Y|\Phi_\theta(X))\|\mathbb{P}_{\mathrm{tr}}(Y|\Phi_\theta(X))).\end{aligned} \tag{9}$$

Following the approximation in (Koyama et al., 2020), we have:

$$\begin{aligned}&\rho_{KL}(\mathbb{Q}^\star(Y|\Phi_\theta(X))\|\mathbb{P}_{\mathrm{tr}}(Y|\Phi_\theta(X))) \approx \mathcal{O}(\alpha^2)+ \\ &\quad \alpha\nabla_{\theta,\eta}\left(\mathcal{R}_{\mathbb{P}_{\mathrm{tr}}}(\theta,\eta)-\mathcal{R}_{\mathbb{Q}^\star}(\theta,\eta)\right)^T\nabla_{\theta,\eta}\mathcal{R}_{\mathbb{Q}^\star}(\theta,\eta)\end{aligned}$$

where α is the learning rate of model parameters θ,η, $\mathcal{R}_{\mathbb{P}_{\mathrm{tr}}} = \mathbb{E}_{\mathbb{P}_{\mathrm{tr}}}[\ell(X,Y)]$ denotes the average prediction error under distribution $\mathbb{P}_{\mathrm{tr}}$, and $\mathcal{R}_{\mathbb{Q}^\star} = \mathbb{E}_{\mathbb{Q}^\star}[\ell(X,Y)]$ denotes the average prediction error under distribution $\mathbb{Q}^\star$.

Given the worst sub-population $\mathbb{Q}^\star$, the overall objective function of the player 2 becomes:

$$\begin{aligned}&\mathcal{L}(\theta,\eta) = \mathbb{E}_{\mathbb{P}_{\mathrm{tr}}}[\ell(h_\eta(\Phi_\theta(X)),Y)]+ \\ &\quad \lambda\nabla_{\theta,\eta}\left(\mathcal{R}_{\mathbb{P}_{\mathrm{tr}}}(\theta,\eta)-\mathcal{R}_{\mathbb{Q}^\star}(\theta,\eta)\right)^T\nabla_{\theta,\eta}\mathcal{R}_{\mathbb{Q}^\star}(\theta,\eta),\end{aligned} \tag{10}$$

which can be efficiently optimized via gradient descent.

5 RELATED WORK

In this section, we discuss the related works in detail. There are mainly two branches of literatures related to our work, including invariant learning (Arjovsky et al., 2019; Ahuja et al., 2020; Koyama et al., 2020; Liu et al., 2021a,c; Ahuja et al., 2021; Creager et al., 2021) and distributionally robust optimization (Duchi et al., 2018; Sinha et al., 2018).

For invariant learning, Arjovsky et al. (2019) first come up with the OOD generalization problem and design a regularizer to learn such representations that the optimal linear classifier remains the same across training environments, and this method is a typical method in invariant learning. And Koyama et al. (2020) theoretically characterize when the invariance will benefit OOD generalization and propose to learn the maximal invariant predictor to achieve OOD optimality. Ahuja et al. (2021) combines invariant learning with information bottleneck for better OOD generalization performance. The proposed invariance definition requires an invariant relationship among all possible environments, termed as the strict invariance. However, whether it exists in real applications remains doubtful, since the noises are likely to change in different environments and therefore violate the strict invariance. Further, the availability of multiple training environments itself is quite hard to meet with in real scenarios, making many invariant learning methods inapplicable in real applications.

In order to mitigate such limitations, recently, some works (Creager et al., 2021; Liu et al., 2021a,c) try to learn pseudo-environments first and then perform invariant learning. Creager et al. (2021) directly maximize the regularizer of IRM with a given biased model to generate environments. Liu et al. (2021a,c) propose to iteratively learn the environment splits and the invariant predictors, although intuitively reasonable, the property of learned environments still remains vague, which renders the proposed framework unstable. Since the property of learned environments cannot be analyzed or guaranteed, whether the invariance can be achieved also remains unclear and cannot be certified. Inspired by this, it is of paramount importance to reformulate the invariant learning problem under latent heterogeneity to a more reasonable one.

Distributionally robust optimization (DRO) methods, typified by f-DRO (Duchi et al., 2018), propose to optimize the worst-case error with respect to a pre-defined distribution set that lies around the training distribution. When the testing distribution lies in the pre-defined distribution set, the OOD generalization performance can be controlled by the worst-case. However, when the target distribution is not captured by the pre-defined set, the performance of DRO depends on the relationship between the target distribution and the worst-case distribution in the pre-defined set, which cannot be guaranteed. This is also reflected in our Figure 1(a) (the curve of f-DRO is quite fluctuant). Unfortunately, such circumstances are quite likely to happen in real scenarios, since the pre-defined set cannot be set too large because of the over-pessimism problem (Hu et al., 2018; Frogner et al., 2019). In this work, we borrow the idea of distribution set from DRO to characterize the sub-population set, based on which we come up with the notion of distributional stability, which is a relaxed alternative of the strict invariance.

6 EXPERIMENTS

Baselines.
We compare our proposed SRM algorithm with the following methods: Empirical Risk Minimization (ERM), Distributionally Robust Optimization (f-DRO, Duchi et al. (2018)), Environment Inference for Invariant Learning (EIIL, Creager et al. (2021)), Kernelized Heterogeneous Risk Minimization (KerHRM, Liu et al. (2021c)) and Invariant Risk Minimization (IRM, Arjovsky et al. (2019)) with environment $\mathcal{E}_{tr}$ labels. Note that IRM requires environment labels, and we provide the ground-truth sub-population labels for IRM.

Evaluation Metrics.
For experiments with multiple testing distributions, we use Mean Error defined as:

$$\text{Mean Error} = \frac{1}{|\mathcal{E}_{\text{test}}|} \sum_{e \in \mathcal{E}_{\text{test}}} \mathbb{E}_{\mathbb{P}^e}[\ell(X, Y)],$$

and Std Error defined as:

$$\text{Std Error} = \sqrt{\frac{1}{|\mathcal{E}_{\text{test}}| - 1} \sum_{e \in \mathcal{E}_{\text{test}}} (\mathbb{E}_{\mathbb{P}^e}[\ell(X, Y)] - \text{Mean Error})^2},$$

and Max Error defined as:

$$\text{Max Error} = \max_{e \in \mathcal{E}_{\text{test}}} \mathbb{E}_{\mathbb{P}^e}[\ell(X, Y)],$$

which are mean error, standard deviation error, and the worst-case error across testing environments $\mathcal{E}_{\text{test}}$.

6.1 Simulation Data

Regression with Selection Bias
In this setting, the relationships between covariates and the target are perturbed through the selection bias mechanism across sub-populations. We generate the data following the mechanism adopted by Liu et al. (2021c, 2022), where we assume $X = [S, V]^T \in \mathbb{R}^{10}$ and $Y = f(S) + \epsilon = \beta^T S + S_1 S_2 S_3 + \mathcal{N}(0, 0.1)$. To generate different sub-populations, we maintain $\mathbb{P}(Y|S)$ the same across sub-populations and leverage a data selection mechanism to vary $\mathbb{P}(Y|V)$. Specifically, we select data point (x_i, y_i) with probability τ_i according to one certain variable $V_b \in V$ as $\tau_i = |r|^{-5*|y_i - \text{sign}(r) \cdot V_b|}$ where $|r| > 1$. Intuitively, r controls the strengths and direction of the spurious correlation between V_b and Y. The larger value of $|r|$ means the stronger spurious correlation between V_b and Y , and $r > 0$ means positive correlation and vice versa (i.e. if $r > 0$, a data point whose V_b is close to its y is more probably to be selected.). Therefore, we use r to define different sub-populations.

For training data, we mix 2000 data points from different r_1 and 200 points from $r_2 = -1.1$. For different testing scenarios, we sample 1000 data points from $r \in \{-1.9, -2.1, \ldots, -2.9\}$, respectively. For our SRM algorithm and f-DRO, we set $\alpha_0 = 0.1$ (the ground truth is 0.09). Linear models are used in this experiment.

Classification with Spurious Correlation
Following Sagawa et al. (2020), we induce spurious correlations between the label $Y \in \{+1, -1\}$ and a spurious attribute $A \in \{+1, -1\}$ of different strengths and directions. We assume $X = [S, V]^T \in \mathbb{R}^{2d}$, where $S \in \mathbb{R}^d$ is the invariant feature generated from the label Y and $V \in \mathbb{R}^d$ the variant feature generated from the spurious attribute A:

$$S|Y \sim \mathcal{N}(Y\mathbf{1}, \sigma_s^2 \mathbb{I}_d), V|A \sim \mathcal{N}(A\mathbf{1}, \sigma_v^2 \mathbb{I}_d). \quad (11)$$

In this setting, we characterize different groups with the bias rate $r \in (0, 1]$, which represents that for $100 \cdot r\%$ data, $A = Y$, and for the other $100 \cdot (1 - r)\%$ data, $A = -Y$. Intuitively, r controls the spurious correlation between the label Y and spurious attribute A. In training, we generate 2000 data points, where 50% points are from group 1 with $r_1 = 0.9$ and the other from group 2 with varying r_2. In testing, we generate 1000 data points with $r_3 = 0.0$ to simulate strong distributional shifts, since the direction of spurious correlations is reversed from training. We design multiple settings with different bias rates r_2 as well as the dimensions d of features. For our SRM algorithm and f-DRO, we set $\alpha_0 = 0.15$ (the ground truth is 0.17). We use a two-layer MLP for this experiment.

Table 1: Overall results in selection bias simulation experiments with varying bias rates r_1.

Bias Ratio r	$r_1 = 1.5$			$r_1 = 1.9$			$r_1 = 2.3$		
Methods	Mean Error	Std Error	Max Error	Mean Error	Std Error	Max Error	Mean Error	Std Error	Max Error
ERM	2.651(±0.106)	0.119(±0.038)	2.820(±0.140)	3.155(±0.210)	0.147(±0.039)	3.348(±0.184)	3.240(±0.174)	0.136(±0.039)	3.433(±0.197)
f-DRO	1.835(±0.144)	0.070(±0.024)	1.940(±0.169)	1.973(±0.261)	0.096(±0.025)	2.107(±0.274)	2.018(±0.422)	0.100(±0.025)	2.149(±0.425)
EIIL	1.764(±0.402)	0.074(±0.022)	1.864(±0.423)	2.043(±0.600)	0.101(±0.036)	2.185(±0.656)	1.840(±0.347)	0.085(±0.022)	1.962(±0.349)
KerHRM	1.825(±0.354)	0.089(±0.040)	1.978(±0.374)	1.658(±0.472)	0.068(±0.031)	1.788(±0.617)	1.572(±0.504)	0.088(±0.036)	1.677(±0.537)
IRM(with $\mathcal{E}_{tr}$ label)	1.683(±0.201)	0.066(±0.024)	1.780(±0.227)	1.782(±0.134)	0.067(±0.018)	1.886(±0.163)	1.964(±0.276)	0.067(±0.015)	2.057(±0.295)
SRM	**1.288**(±0.344)	**0.059**(±0.024)	**1.367**(±0.365)	**1.323**(±0.223)	**0.054**(±0.020)	**1.402**(±0.233)	**1.382**(±0.283)	**0.059**(±0.018)	**1.457**(±0.299)

Table 2: Overall results in classification simulation experiments with varying bias rates r_2.

Bias Ratio r_2	$r_2 = 0.75$				$r_2 = 0.80$			
Dimension d	$d = 5$		$d = 10$		$d = 5$		$d = 10$	
Methods	Train Acc	Test Acc	Train Acc	Test Acc	Train Acc	Test Acc	Train Acc	Test Acc
ERM	**0.917**(±0.009)	0.388(±0.039)	**0.972**(±0.007)	0.573(±0.026)	**0.931**(±0.005)	0.364(±0.023)	**0.975**(±0.005)	0.526(±0.030)
f-DRO	0.766(±0.012)	0.452(±0.021)	0.920(±0.006)	0.611(±0.028)	0.787(±0.011)	0.427(±0.022)	0.930(±0.005)	0.616(±0.022)
EIIL	0.727(±0.145)	0.544(±0.058)	0.814(±0.160)	0.451(±0.049)	0.743(±0.155)	0.571(±0.050)	0.823(±0.165)	0.406(±0.056)
KerHRM	0.784(±0.035)	0.636(±0.182)	0.834(±0.143)	0.659(±0.205)	0.780(±0.043)	0.665(±0.178)	0.800(±0.097)	0.674(±0.139)
IRM(with $\mathcal{E}_{tr}$ label)	0.855(±0.010)	0.467(±0.046)	0.908(±0.007)	0.529(±0.058)	0.876(±0.005)	0.386(±0.047)	0.914(±0.006)	0.448(±0.056)
SRM	0.781(±0.032)	**0.716**(±0.066)	0.869(±0.023)	**0.684**(±0.052)	0.787(±0.030)	**0.703**(±0.073)	0.871(±0.017)	**0.697**(±0.061)

Better OOD Generalization Performance: We report the results of the regression and classification tasks in Table 1 and 2. From the results, our SRM outperforms all baselines in terms of higher prediction accuracy and better stability among distributional shifts, which validates that our SRM can achieve better OOD generalization performance and is consistent with our theoretical analysis in Theorem 1.

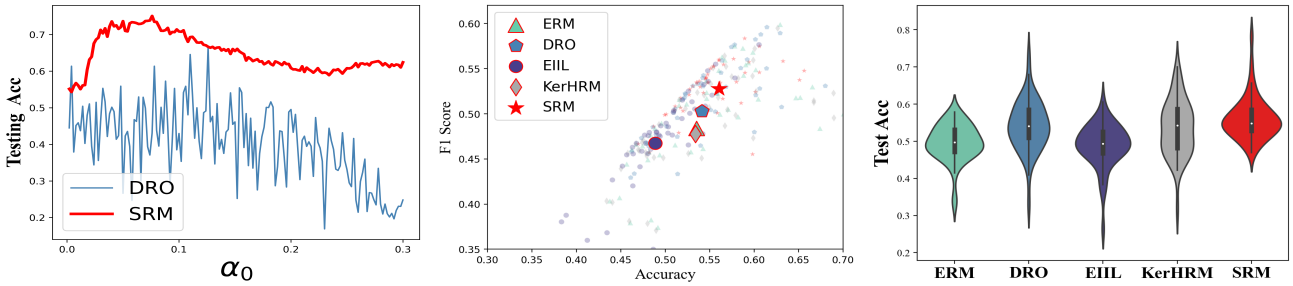
α_0 Controls the Extent of Stability: In the definition of α_0-distributional stability, α_0 controls the range of stability, i.e. smaller α_0 examines more fine-grained stability. To demonstrate the effect of α_0 in our SRM algorithm, for the classification task, we plot the curve of testing accuracy w.r.t. α_0 for our SRM and f-DRO in Figure 1(a). Since the real proportion of the minor sub-population is set to 0.17, we hope SRM is effective when $\alpha_0 \leq 0.17$. From the results, we could see that the performances of SRM maintain at a high level for $\alpha_0 \in [0.05, 0.17]$, which validates our intuitions. For too small α_0, the performances drop due to the insufficient number of samples and stronger noises. Also, the performances of f-DRO are oscillating, which corresponds with our analysis in Remark 2.2 that: since distributional robustness only cares about the worst sub-population performances, when the testing distribution falls out of the pre-defined distribution set, it cannot guarantee the OOD generalization performance. However, for our SRM, the guarantees for the OOD generalization ability in Theorem 1 do not put strong requirements for the testing distributions, since it only requires the learnability of the problem.

6.2 Real-World Data: Retiring Adults

To better validate the effectiveness of the proposed SRM algorithm, we consider a much more challenging scenario on a real-world dataset, named ACSTravelTime (Ding et al., 2021). The task is to predict whether an individual has a commute to work that is longer than 20 minutes. In this task, we have 16 features and 1,428,642 data points in total from all 50 US states. Since there are 50 distinct environments, this dataset contains natural geographic shifts, which makes it suitable for testing the OOD generalization performances. In *training*, we sample 2000 data points from MA and validate on the rest data from MA. In *testing*, we test different methods on all the other 49 states.

In Figure 1(b), we plot the accuracy and F1 score for each method on the 50 states, and in Figure 1(c) we show the overall testing accuracy of different methods. Note that the original code released by KerHRM is too time-consuming to run on this data because of the large amount of data (over 1 million data points), therefore we use HRM (Liu et al., 2021a) here to replace the KerHRM, which can only deal with the raw feature data. Since there is one environment in this experiment and we do not know the underlying sub-populations, we cannot compare with IRM in this setting. And EIIL can be viewed as an alternative to IRM with learned environments from training data.

From the results in Figure 1(b), the average performance of our SRM locates in the top right of the figure, which shows that our methods achieve the best OOD generalization performance w.r.t. testing accuracy and F1 score. Further, in Figure 1(c), for our SRM, the performances of most environments are concentrated at high accuracy, and the variance of different environments is significantly smaller than the baselines. It shows that our SRM algorithm can learn some distributional stability among different sub-populations, which benefits the generalization performances. And



(a) Certified robustness. (b) F1 Score and Testing Accuracy. (c) Overall testing accuracy.

Figure 1: Experimental results. (a): Demonstration of the certified robustness via the classification task (in Section 6.1), where we vary the α_0 and plot the corresponding testing accuracy for f-DRO and our proposed SRM. (b): The F1 score and testing accuracy on all 50 target states of different methods. We highlight the average F1 score and testing accuracy (in Section 6.2). (c): The distribution of testing accuracy of different methods (in Section 6.2).

the good OOD generalization performance also corresponds with our intuition from Theorem 1 that considering the distributional stability could benefit the OOD generalization error.

7 CONCLUSION

In this paper, we propose the distributional stability, which measures the stability of prediction mechanisms among sub-populations. Based on this criterion, we propose an approximated algorithm, termed stable risk minimization, to enhance the model's stability with respect to distribution shifts in prediction mechanisms. Despite the theoretical and empirical results, our work has the following limitations (or potential directions to improve):

Analysis of the approximation. Based on the overall objective function in Equation (5), we make several approximations to derive a tractable optimization algorithm. A notable challenge associated with this approach is the difficulty in thoroughly analyzing the behavior of the approximated algorithm, particularly with regard to its convergence properties and the bounds on its generalization error. A promising avenue for future research lies in the development of improved approximation techniques that come with stronger theoretical guarantees.

Lack of large-scale suitable datasets. In the current version of our study, both simulated and real-world experiments are conducted on a small scale. This limitation is largely due to the nature of datasets commonly employed in large-scale research, which predominantly consist of image data. These datasets usually exhibit shifts in the input space, X, rather than in the conditional distribution ($Y|X$-shifts) that are more pertinent to our investigation into invariant learning.

As the field of invariant learning evolves, a noticeable trend is the application of these methods to complex tasks, particularly image classification datasets. However, a crucial question emerges: Are these image datasets genuinely conducive to invariant learning methods aimed at aligning the $Y|X$ distributions? Research by Gulrajani and Lopez-Paz (2020) reveals that Empirical Risk Minimization (ERM) often outperforms most domain generalization and invariant learning methods tailored for these datasets. This suggests that the prevalent distribution shifts in image datasets are primarily X-shifts, with the primary objective being to model $\mathbb{E}_{\mathbb{P}_{\text{tr}}}[Y|X]$. Additionally, numerous empirical studies, such as those by (Miller et al., 2021), have identified a strong correlation between out-of-distribution (OOD) generalization performance and in-distribution (ID) performance. This correlation further underscores the inadequacy of traditional image classification tasks as a testing ground for invariant learning methods.

In light of these findings, we advocate for a shift in research focus towards understanding the patterns of distribution shifts in real-world applications, as highlighted by (Liu et al., 2023). A promising avenue of exploration involves the creation of real-world, large-scale datasets featuring $Y|X$-shifts. These datasets would likely offer a more fitting and challenging environment for assessing the capabilities of invariant learning methods.

8 Acknowledgements

Peng Cui was supported by National Natural Science Foundation of China (No. 62141607). Bo Li's research was supported by the National Natural Science Foundation of China (No.72171131, 72133002); the Technology and Innovation Major Project of the Ministry of Science and Technology of China under Grants 2020AAA0108400 and 2020AAA0108403.

References

Ahuja, K., Caballero, E., Zhang, D., Bengio, Y., Mitliagkas, I., and Rish, I. (2021). Invariance principle meets information bottleneck for out-of-distribution generalization. *CoRR*, abs/2106.06607. 1, 6

Ahuja, K., Shanmugam, K., Varshney, K. R., and Dhurandhar, A. (2020). Invariant risk minimization games. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 145–155. PMLR. 1, 2, 6

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *CoRR*, abs/1907.02893. 1, 2, 4, 6, 7

Creager, E., Jacobsen, J., and Zemel, R. S. (2021). Environment inference for invariant learning. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2189–2200. PMLR. 1, 2, 6, 7

Ding, F., Hardt, M., Miller, J., and Schmidt, L. (2021). Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*. 8

Duchi, J. C., Hashimoto, T., and Namkoong, H. (2019). Distributionally robust losses against mixture covariate shifts. *Under review*, 2. 2, 3

Duchi, J. C., Namkoong, H., and Namkoong, H. (2018). Learning models with uniform performance via distributionally robust optimization. *CoRR*, abs/1810.08750. 2, 3, 6, 7

Frogner, C., Claici, S., Chien, E., and Solomon, J. (2019). Incorporating unlabeled data into distributionally robust learning. *CoRR*, abs/1912.07729. 7

Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. In *International Conference on Learning Representations*. 9

Hu, W., Niu, G., Sato, I., and Sugiyama, M. (2018). Does distributionally robust supervised learning give robust classifiers? In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2034–2042. PMLR. 7

Koyama, M., Yamaguchi, S., and Yamaguchi, S. (2020). Out-of-distribution generalization with maximal invariant predictor. *CoRR*, abs/2008.01883. 1, 2, 6

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. (2021a). Heterogeneous risk minimization. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6804–6814. PMLR. 1, 2, 6, 8

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. (2021b). Integrated latent heterogeneity and invariance learning in kernel space. *Advances in Neural Information Processing Systems*, 34:21720–21731. 1, 2

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. (2021c). Kernelized heterogeneous risk minimization. *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*. 6, 7

Liu, J., Wang, T., Cui, P., and Namkoong, H. (2023). On the need for a language describing distribution shifts: Illustrations on tabular datasets. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 9

Liu, J., Wu, J., Pi, R., Xu, R., Zhang, X., Li, B., and Cui, P. (2022). Measure the predictive heterogeneity. In *The Eleventh International Conference on Learning Representations*. 7

Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y., and Schmidt, L. (2021). Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR. 9

Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012. 1

Rojas-Carulla, M., Schölkopf, B., Turner, R. E., and Peters, J. (2018). Invariant models for causal transfer learning. *J. Mach. Learn. Res.*, 19:36:1–36:34. 1

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. (2020). An investigation of why overparameteri-

zation exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8346–8356. PMLR. 7

Shaban, A., Cheng, C.-A., Hatch, N., and Boots, B. (2019). Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR. 6

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32. 6

Sinha, A., Namkoong, H., and Duchi, J. C. (2018). Certifying some distributional robustness with principled adversarial training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. 3, 4, 6

Ye, H., Xie, C., Cai, T., Li, R., Li, Z., and Wang, L. (2021). Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*. 3

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]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

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]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (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]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

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]
   (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]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

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 PROOF

Proof. Denote the upper bound of $\ell(\cdot,\cdot)$ as $M > 0$. For any $e \in \text{supp}(\mathcal{E})$, denote $\mathbb{P}_e^{'}(Y,\Phi) = \mathbb{P}^e(Y|\Phi)\mathbb{P}(\Phi)$ and $\mathbb{P}_{\text{tr}}^{'}(Y,\Phi) = \mathbb{P}_{\text{tr}}(Y|\Phi)\mathbb{P}(\Phi)$, and then we have

$$\begin{aligned}
& \mathbb{E}\left[\mathbb{E}_{\mathbb{P}^e}[\ell(f(\Phi),Y)|\Phi] - \mathbb{E}_{\mathbb{P}_{\text{tr}}}[\ell(f(\Phi),Y)|\Phi]\right] && (12)\\
\leq\ & 2M \cdot \text{TV}(\mathbb{P}_e^{'}, \mathbb{P}_{\text{tr}}^{'}) && (13)\\
\leq\ & 2M \cdot \sqrt{\frac{1}{2}\rho_{\text{KL}}(\mathbb{P}^e(Y|\Phi)\|\mathbb{P}^{\text{tr}}(Y|\Phi))} && (14)\\
\leq\ & \mathcal{O}(\sqrt{1-e^{-s(\delta)}}) && (15)
\end{aligned}$$

□

B Experimental Details

In this section, we demonstrate the details of our simulated experiments.

Regression In this setting, the correlations among covariates are perturbed through a selection bias mechanism. We assume $X = [S,V]^T \in \mathbb{R}^{10}$ with $S \in \mathbb{R}^5$ and $V \in \mathbb{R}^5$. We assume $Y = f(S) + \epsilon$ and $\mathbb{P}(Y|S)$ remains invariant across environments while $P(Y|V)$ can arbitrarily change.

Therefore, we generate training data points with the help of auxiliary variables $Z \in \mathbb{R}^6$ as following:

$$\begin{aligned}
& Z_1,\ldots,Z_6 \overset{iid}{\sim} \mathcal{N}(0,2.0) && (16)\\
& V_1,\ldots,V_5 \overset{iid}{\sim} \mathcal{N}(0,2.0) && (17)\\
& S_i = 0.8 * Z_i + 0.2 * Z_{i+1} \qquad for\ \ i = 1,\ldots,5 && (18)
\end{aligned}$$

To induce model misspecification, we generate Y as:

$$Y = f(S) + \epsilon = \theta_s(S)^T + S_1S_2S_3 + \epsilon \qquad (19)$$

where $\theta_s = [\frac{1}{2}, -1, 1, -\frac{1}{2}, 1]$, and $\epsilon \sim \mathcal{N}(0,1.0)$. As we assume that $\mathbb{P}(Y|S)$ remains unchanged while $\mathbb{P}(Y|V)$ can vary across environments, we design a data selection mechanism to induce this kind of distribution shifts. For simplicity, we select data points according to a certain variable $V_b \in V$:

$$\begin{aligned}
& \tau = |r|^{-5*|y-\text{sign}(r)*v_b|} && (20)\\
& \mu \sim Uni(0,1) && (21)\\
& M(r;(x,y)) = \begin{cases} 1, & \mu \leq \tau \\ 0, & \text{otherwise} \end{cases} && (22)
\end{aligned}$$

where $|r| > 1$. Given a certain r, a data point (x,y) is selected if and only if $M(r;(x,y)) = 1$ (i.e. if $r > 0$, a data point whose V_b is close to its Y is more probably to be selected.)

Classification We set $\sigma_s^2 = 3.0$ and $\sigma_v^2 = 0.3$ to let the model more prone to use spurious V since it is more informative.

As for the hyper-parameter for SRM and f-DRO, for regression data, we set $\alpha_0 = 0.1$ (the true minor subpopulation ratio is 0.09); for classification data, we set $\alpha_0 = 0.15$ (the true minor subpopulation ratio is 0.17). As for the validation data, we sample *i.i.d* data as training data and compare both the worst-case performance of two subpopulations. As for IRM, we select the parameter of the regularizer $\lambda \in \{0.1, 0.3, \ldots, 0.9, 1.5, 5.0, 10.0\}$ according to the validation performance. As for EIIL, we set the epochs for splitting environments to 1e4 for good convergence, and other parameters are the same as IRM. As for KerHRM, we set the cluster_num to be the ground-truth 2. All experiments are run on a GPU server with one NVIDIA GeForce RTX 3090.