Distributionally Robust Learning With Stable Adversarial Training

Jiashuo Liu, Zheyan Shen, Peng Cui, Senior Member, IEEE, Linjun Zhou, Kun Kuang, and Bo Li

Abstract—Machine learning algorithms with empirical risk minimization are vulnerable under distributional shifts due to the greedy adoption of all the correlations found in training data. There is an emerging literature on tackling this problem by minimizing the worst-case risk over an uncertainty set. However, existing methods mostly construct ambiguity sets by treating all variables equally regardless of the stability of their correlations with the target, resulting in the overwhelmingly-large uncertainty set and low confidence of the learner. In this paper, we propose a novel Stable Adversarial Learning (SAL) algorithm that leverages heterogeneous data sources to construct a more practical uncertainty set and conduct differentiated robustness optimization, where covariates are differentiated according to the stability of their correlations with the target. We theoretically show that our method is tractable for stochastic gradient-based optimization and provide the performance guarantees for our method. Empirical studies on both simulation and real datasets validate the effectiveness of our method in terms of uniformly good performance across unknown distributional shifts.

Index Terms—Stable adversarial learning, spurious correlation, distributionally robust learning, wasserstein distance

1 INTRODUCTION

Traditional machine learning algorithms which optimize the average empirical loss often suffer from the poor generalization performance under distributional shifts induced by latent heterogeneity, unobserved confounders or selection biases in training data[1], [2], [3]. However, in high-stake applications such as medical diagnosis[4], criminal justice[5], [6] and autonomous driving [7], it is critical for the learning algorithms to ensure the robustness against potential unseen data. Therefore, robust learning methods have recently aroused much attention due to its favorable property of robustness guarantee[8], [9], [10]. Instead of optimizing the empirical cost on training data, robust learning methods seek to optimize the worst-case cost over an uncertainty set and can be further separated into two main branches named adversarially and distributionally robust learning. In adversarially robust learning, the uncertainty set is constructed point-wisely[9], [10], [11], [12]. Specifically, adversarial attack is performed independently on each data point within a \( L_2 \) or \( L_\infty \) norm ball around itself to maximize the loss of current classification model. In distributionally robust learning, on the other hand, the uncertainty set is characterized on a distributional level[13], [14], [15]. A joint perturbation, typically measured by Wasserstein distance or \( f \)-divergence between distributions, is applied to the entire distribution entailed by training data. These methods can provide robustness guarantees under distributional shifts when testing distribution is captured in the built uncertainty set. However, in real scenarios, to contain the possible true testing distribution, the uncertainty set is often overwhelmingly large, and results in learned models with fairly low confidence, which is also referred to as the over pessimism or the low confidence problem[16], [17]. That is, with an overwhelmingly large set, the learner optimizes for implausible worst-case scenarios, resulting in meaningless results (e.g., the classifier assigns equal probability to all classes). Such a problem greatly hurts the generalization ability of robust learning methods in practice.

The essential problem of the above methods lies in the construction of the uncertainty set. To address the over pessimism of the learning algorithm, one should form a more practical uncertainty set which is likely to contain the potential distributional shifts in the future and meanwhile is as small as possible. More specifically, in real applications, we observe that different covariates may be perturbed in a non-uniform way, which should be considered when building a
practical uncertainty set. Taking the problem of waterbirds and landbirds classification as an example[18]. There exist two types of covariates where the stable covariates (e.g., representing the bird itself) preserve immutable correlations with the target across different environments, while those unstable ones (e.g., representing the background) are pretty likely to change (e.g., waterbirds on land). Therefore, for the example above, the construction of the uncertainty set should be anisotropic which mainly focuses on the perturbation of those unstable covariates (e.g., background) to generate more practical and meaningful samples.

Further, we illustrate the anisotropic uncertainty set in Fig. 1, where blue points denote the observed training distribution \( \mathcal{N}(0, I) \). And we sample data points from all distributions in the uncertainty set captured by an isotropic Wasserstein ball around the observed distribution, which are colored orange. We can see from Fig. 1 that the original distribution is perturbed equally along both the stable and unstable direction. With the intuition above, we propose that the ideal uncertainty should be like green points, which only perturb the training distribution along unstable directions. Following this, there are several work[19], [20] based on the adversarial attack which focus on perturbing the color or background of images to improve the adversarial robustness. However, these methods mainly follow a step by step routine where the segmentation is conducted first to separate the background from the foreground and cannot theoretically provide robustness guarantees under unknown distributional shifts, which greatly limits their applications on more general settings.

In this paper, we propose the Stable Adversarial Learning (SAL) algorithm to address this problem in a more principled and unified way, which leverages heterogeneous data sources to construct a more practical uncertainty set. Specifically, we adopt the framework of Wasserstein distributionally robust learning(WDRL) and further characterize the uncertainty set to be anisotropic according to the stability of covariates across the multiple environments, which induces stronger adversarial perturbations on unstable covariates than those stable ones. A synergistic algorithm is designed to jointly optimize the covariates differentiating process as well as the adversarial training process of model’s parameters. Compared with traditional robust learning techniques, the proposed method is able to provide robustness under strong distributional shifts while not hurting much confidence of the learner. Theoretically, we prove that our method constructs a more compact uncertainty set, which as far as we know is the first analysis of the compactness of adversarial sets in WDRL literature. Empirically, the advantages of our SAL algorithm are demonstrated on both synthetic and real-world datasets in terms of uniformly good performance across distributional shifts.

2 RELATED WORK

In this section, we investigate several strands of related literature more thoroughly, including domain adaptation, domain generalization, stable learning and distributionally robust learning.

Domain adaptation methods[21] leverage the data from target domain to assist the model training on source domain. Therefore the resulted model could capture the possible distributional shift in testing. Shimodaira[22] proposes to assign each training data a new weight equal to the density ratio between source and target distribution, and therefore guarantees the optimality of learned model on test distribution. Then several techniques have been proposed to estimate the ratio more accurately, such as discriminative estimation[23], kernel mean matching[24] and maximum entropy[25]. Apart from reweighting methods, deep learning based methods[26], [27] learn a transformation in feature space to characterize both source and target domain. However, the deployment of domain adaptation methods in real applications, where one can hardly access data from target domain, is quite limited.

Compared with domain adaptation, domain generalization techniques do not require the availability of target domain data and become more and more popular these years due to its practicability. Different from domain adaptation, domain generalization methods propose to learn a domain-invariant classifier with multiple training domains. Muandet et al. [28] propose a kernel-based optimization algorithm to learn an invariant latent space of data across training domains. Through the lens of causality[29], [30], M. Arjovsky et al. [31] propose Invariant Risk Minimization to learn invariant representation with theoretical guarantee of the optimality of out-of-distribution generalization, which gains the most attention recently. Also, stable learning methods [3], [32], [33] propose to decorrelate the covariates via sample reweighting to estimate the real causal effects, which enhances the stability under distributional shifts. However, they only deal with the covariate shift problem and do not apply to other kinds of distributional shifts (e.g., concept shifts brought by anti-causal variables).

Distributionally robust learning (DRL), from the optimization literature, proposes to optimize for the worst-case cost over an uncertainty distribution set, so as to protect the model against the potential distributional shifts in the uncertainty set, which is constrained by moment or support conditions [34], [35], or \( f \)-divergence [17], [36]. As the uncertainty set formulated by Wasserstein ball is much more flexible, Wasserstein Distributionally Robust Learning (WDRL) has been...
widely studied [13], [14], [37]. WDRL for logistic regression was proposed by Abadeh et al. [37]. Sinha et al. [13] achieved moderate levels of robustness with little computational cost relative to empirical risk minimization with a Lagrangian penalty formulation of WDRL. Esfahani and Kuhn [14] formulated the distributionally robust optimization problems over Wasserstein balls as finite convex programs under mild assumptions. Although DRL offers an alternative to empirical risk minimization for robust performance under distributional perturbations, there has been work questioning its real effect in practice. Hu et al. [38] proved that when the DRL is applied to classification tasks, the obtained classifier ends up being optimal for the observed training distribution, and the core of the proof lies in the over-flexibility of the built uncertainty set. And Fronger et al. [16] also pointed out the problem of overwhelmingly-large decision set, and they used large number of unlabeled examples to further constrain the distribution set.

### 3 Problem Setting

As mentioned above, the uncertainty set built in WDRL is often overwhelmingly large in wild high-dimensional scenarios. To demonstrate this over pessimism problem of WDRL, we design a toy example in 6.1.1 to show the necessity to construct a more practical uncertainty set. Indeed, without any prior knowledge or structural assumptions, it is quite difficult to design a practical set for robustness under distributional shifts.

Therefore, in this work, we consider a dataset $D = \{D^i\}_{i\in\mathcal{E}_r}$, which is a mixture of data $D^i = \{(x^i_e, y^i_e)\}_{e=1}^n$ collected from multiple training environments $e \in \mathcal{E}_r$, $x^i_e \in \mathcal{X}$ and $y^i_e \in \mathcal{Y}$ are the $i$-th data and label from environment $e$ respectively. Specifically, each dataset $D^i$ contains examples identically and independently distributed according to some joint distribution $P_{XY}$ on $\mathcal{X} \times \mathcal{Y}$. Given the observations that in real scenarios, different covariates have different extents of stability, we propose assumption 1.

**Assumption 1.** There exists a decomposition of all the covariates $X = \{S, V\}$, where $S$ represents the stable covariate set and $V$ represents the unstable one, so that for all environments $e \in \mathcal{E}_r$, $E[Y|S^e = s, V^e = v] = E[Y|S = s] = E[Y|S = s]$. Intuitively, assumption 1 indicates that the correlation between stable covariates $S$ and the target $Y$ stays invariant across environments, which is quite similar to the invariance in [31], [39], [40]. Moreover, assumption 1 also demonstrates that the influence of $V$ on the target $Y$ can be wiped out as long as whole information of $S$ is accessible. Under the assumption 1, the disparity among covariates revealed in the heterogeneous datasets can be leveraged for better construction of the uncertainty set. And based on assumption 1, we propose our problem:

**Problem 1.** Given multi-environments training data $D = \{D^i\}_{i\in\mathcal{E}_r}$, under assumption 1, the goal is to build a more practical uncertainty set for distributionally robust learning and achieve stable performance across distributional shifts with respect to low Mean Error defined as

$$\text{Mean Error} = \frac{1}{|\mathcal{E}_r|} \sum_{e\in\mathcal{E}_r} \mathcal{L}^e$$

and low Std Error defined as

$$\text{Std Error} = \sqrt{\frac{1}{|\mathcal{E}_r| - 1} \sum_{e\in\mathcal{E}_r} (\mathcal{L}^e - \text{Mean Error})^2}.$$

### 4 Method

In this work, we propose the Stable Adversarial Learning (SAL) algorithm, which leverages heterogeneous data to build a more practical uncertainty set with covariates differentiated according to their stability.

First, we introduce the Wasserstein Distributionally Robust Learning (WDRL) framework which attempts to learn a model with minimal risk against the worst-case distribution in the uncertainty set characterized by Wasserstein distance:

**Definition 1.** Let $Z \subset \mathbb{R}^{m+1}$ and $Z = \mathcal{X} \times \mathcal{Y}$, given a transportation cost function $c: Z \times Z \rightarrow \{0, \infty\}$, which is nonnegative, lower semi-continuous and satisfies $c(z, z) = 0$, for probability measures $P$ and $Q$ supported on $Z$, the Wasserstein distance between $P$ and $Q$ is

$$W_c(P, Q) = \inf_{M \in \Pi(P, Q)} \mathbb{E}_{(z, z') \sim M}[c(z, z')],$$

where $\Pi(P, Q)$ denotes the couplings with $M(A, Z) = P(A)$ and $M(Z, A) = Q(A)$ for measures $M$ on $Z \times Z$.

Following the intuition above that the uncertainty should not be isotropic along stable and unstable directions, we propose to learn an anisotropic uncertainty set with the help of heterogeneous environments. The objective function of our SAL algorithm is

$$\min_{w, \omega} \sup_{W_{\omega}(Q, P_\theta)} \mathbb{E}_{X \sim q}[\ell(\theta; X, Y)]$$

s.t. $w \in \arg \min_w \mathbb{W} \left\{ \frac{1}{|\mathcal{E}_r|} \sum_{e\in\mathcal{E}_r} \mathcal{L}^e(\theta) + \alpha \max_{w, \omega} \mathbb{E}_{\mathcal{E}_r} \mathcal{L}^\omega - \mathcal{L}^e \right\}$

where $P_\theta$ denotes the training distribution, $W_{\omega}$ denotes the Wasserstein distance with transportation cost function $c_\omega$ defined as

$$c_\omega(z, z') = \|w \odot (z - z')\|^2$$

and $\mathbb{W} = \{w : w \in [1, +\infty)^{m+1} \land \min(w) = 1\}$ denotes the covariate weight space ($\min(w)$ denotes the minimal element of $w$), and $\mathcal{L}^\omega$ denotes the average loss in environment $e \in \mathcal{E}_r$, $\alpha$ is a hyper-parameter to adjust the tradeoff between average performance and the stability.

The core of our SAL is the covariate weight learning procedure in equation 3. In our algorithm, the uncertainty set is built to achieve stable performance across heterogeneous environments. Intuitively, $w$ controls the perturbation level of each covariate and formulates an anisotropic uncertainty set compared with the conventional WDRL methods. The objective function of $w$ (equation 3) contains two parts: the average loss in training environments as well as the maximum margin, which aims at learning such $w$ that the resulting uncertainty set leads to a learner with uniformly good performance across environments. Equation 2 is the objective function of model’s parameters via distributionally robust learning with the learnable covariate weight $w$. During training, the covariate weight $w$ and model’s parameters $\theta$ are iteratively optimized.

Details of the algorithm are delineated below. We first will introduce the optimization of model’s parameter in Section 4.1, then the transportation cost function learning
procedure in Section 4.2. The pseudo-code of the whole Stable Adversarial Learning (SAL) algorithm is shown in Algorithm 1.

**Algorithm 1. Stable Adversarial Training**

**Input:** Multi-environments data $D^1, D^2, \ldots, D^n$, where $D^e = (X^e, Y^e), e \in E$

**Hyperparameters:** $T, T_o, T_w, m_l, \epsilon_c, \epsilon_{th}, \epsilon_{w}, \alpha$

**Initialize:** $w = [1.0, \ldots, 1.0]$

for $i = 1$ to $T$

for $j = 0$ to $T_w - 1$

Initialize $X_0$ as: $X_0 = X$

for $k = 0$ to $m_l - 1$

[Approximate the supremum of $s_l(X)$ for $X^e$ from all $e \in E$]

$\tilde{X}_{k+1} = \tilde{X}_k + \epsilon_c \nabla_{\tilde{X}} \{ \ell(\theta; \tilde{X}_k) - \lambda \epsilon_w (\tilde{X}_k, X_0) \}$

end for

Update $\theta$ as: $\theta^{i+1} = \theta_i - \epsilon_{th} \nabla_{\theta}(\theta_i; (\tilde{X}_m, Y))$

end for

Calculate $R(\theta)$ as: $R(\theta) = \frac{1}{|E|} \sum_{e \in E} \ell^e + \alpha (\text{sup}_{P \in \mathcal{P}}(C^e - \ell^e))$

$w^i = w^0$

for $j = 0$ to $T_w - 1$

Update $w$ as: $w^{i+1} = w^i - \epsilon_w \nabla_w R(\theta)$

end for

Update $w$ as: $w^{i+1} = \text{Proj}_W(w^i)$.

end for

4.1 Tractable Optimization

In SAL algorithm, the model’s parameters $\theta$ and covariate weight $w$ is optimized iteratively. In each iteration, given current $w$, the objective function for $\theta$ is

$$
\min_{\theta \in \Theta} \sup_{Q: W_{\text{tr}}(Q, P_0) \leq \rho} \mathbb{E}_{X,Y \sim Q} [\ell(\theta; X, Y)].
$$

(5)

The duality results in lemma 1 show that the infinite-dimension optimization problem 5 can be reformulated as a finite-dimensional convex optimization problem [14]. Besides, inspired by [13], a Lagrangian relaxation is provided for computation efficiency.

**Lemma 1.** Let $Z = X \times Y$ and $\ell: \Theta \times Z \to \mathbb{R}$ be continuous. For any distribution $Q$ and any $\rho \geq 0$, let $s_l(\theta; (x, y)) = \sup_{\xi \in Z} \{ \ell(\theta; \xi) - \lambda \epsilon_w (\xi, (x, y)) \}$, $\mathcal{P} = \{ Q: W_e(Q, P_0) \leq \rho \}$, we have

$$
\sup_{Q \in \mathcal{P}} \mathbb{E}_Q [\ell(\theta; x, y)] = \inf_{\lambda \geq 0} \{ \lambda \rho + \mathbb{E}_{P_0}[s_l] \},
$$

(6)

and for any $\lambda \geq 0$, we have

$$
\sup_{Q \in \mathcal{P}} \{ \mathbb{E}_Q [\ell(\theta; x, y)] - \lambda W_{\text{tr}}(Q, P_0) \} = \mathbb{E}_{P_0}[s_l].
$$

(7)

The original problem 5 can first be reformulated as equation 6 by duality. However, the infimum with respect to $\lambda$ is also intractable. Therefore, we give up the prescribed amount $\rho$ of robustness in equation (5) and focus instead on the relaxed Lagrangian penalty function for efficiency in equation (7). Notice that there exists only the inner supremum in $\mathbb{E}_{P_0}[s_l(\theta; (x, y))]$ in equation (7), which can be seen as a relaxed Lagrangian penalty function of the original objective function (5). Following lemma 1, we derive the loss function on empirical distribution $\hat{P}_N$ as

$$
\hat{L}(\theta) = \frac{1}{N} \sum_{i=1}^N s_l(\theta; (x_i, y_i)).
$$

(8)

Recall that $s_l(\theta; (x, y)) = \sup_{\xi \in Z} \{ \ell(\theta; \xi) - \lambda \epsilon_w (\xi, (x, y)) \}$, we propose to convert the minimization of $\hat{L}$ over $\theta$ to a minimax procedure as done in [13] to approximate the supremum for $s_l$

$$
\min_{\theta} \max_{\xi \in Z} \mathbb{E}_{\hat{P}_N} \left[ \ell(\theta; \tilde{X}, Y) - \lambda \epsilon_w ((\tilde{X}, Y), (X, Y)) \right].
$$

(9)

Specifically, given predictor $X$, we adopt gradient ascent to obtain an approximate maximizer $\tilde{X}$ of $s_l(\theta; (X, Y))$ and optimize the model’s parameter $\theta$ using $\tilde{X}, Y$. In the following parts, we simply use $\tilde{X}$ to denote $\{X\}_N$, which means the set of maximizers for training data $\{x\}_N$. The convergence guarantee for this optimization can be referred to [13].

4.2 Learning for Transportation Cost $w$

We introduce the learning for transportation cost function $c_w$ in this section. In supervised scenarios, perturbations are typically only added to predictor $X$ and not target $Y$. Therefore, we simplify $c_w: Z \times Z \to [0, +\infty) (Z = X \times Y)$ to be

$$
c_w(x_1, x_2) = c_w(x_1, x_2) + \infty \cdot 1(y_1 \neq y_2)
$$

(10)

$$
= \| w \circ (x_1 - x_2) \|_2^2 + \infty \cdot 1(y_1 \neq y_2)
$$

(11)

and omit ‘$y$-part’ in $c_w$ as well as $w$, that is $w \in [1, +\infty)^m$ in the following parts. Intuitively, $w$ controls the strength of adversary put on each covariate. The higher the weight is, the weaker perturbation is put on the corresponding covariate. Ideally, we hope the covariate weights on stable covariates are extremely high to protect them from being perturbed and to maintain the stable correlations, while weights on unstable covariates are nearly 1 to encourage perturbations for breaking the harmful spurious correlations. With the goal towards uniformly good performance across environments, we come up with the objective function $R(\theta(w))$ for learning $w$ as

$$
R(\theta(w)) = \frac{1}{|E|} \sum_{e \in E} \ell^e(\theta(w)) + \alpha \max_{p \in \mathcal{P}} \left( C^e(\theta(w)) - C^e(\theta(w)) \right),
$$

(12)

where $\alpha$ is the hyper-parameter. $R(\theta(w))$ contains two parts: the first is the average loss in multiple training environments; the second reflects the max margin among environments, which reflects the stability of $\theta(w)$, since it is easy to prove that $\max_{p \in \mathcal{P}} \left( C^e(\theta(w)) - C^e(\theta(w)) \right) = 0$ if and only if the errors among all training environments are same. Here $\alpha$ is used to adjust the tradeoff between average performance and stability.

Given current $\theta^i$, we can update $w$ as

$$
w^{i+1} = \text{Proj}_W \left( w^i - \epsilon_w \frac{\partial R(\theta^i)}{\partial w} \right).
$$

(13)
where \( ProjW \) means projecting onto the space \( W \). And the remaining work is how to calculate the gradient \( \partial R(\theta(w))/\partial w \), which we will introduce in detail in following section 4.2.1.

### 4.2.1 Calculation of \( \partial R(\theta(w))/\partial w \)

In order to optimize \( w \), \( \partial R(\theta(w))/\partial w \) can be approximated as following

\[
\frac{\partial R(\theta(w))}{\partial w} = \frac{\partial R}{\partial \theta} \frac{\partial \theta}{\partial \theta} \frac{\partial X_A}{\partial \theta} \frac{\partial w}{\partial X_A}. \tag{14}
\]

Note that the first term \( \partial R/\partial \theta \) can be calculated easily. The second term can be approximated during the gradient descent process of \( \theta \) as

\[
\theta^{t+1} = \theta^t - c_0 \nabla_{\theta} \hat{L}(\theta^t; \tilde{X}, Y) \tag{15}
\]

\[
\frac{\partial \theta}{\partial \theta} \approx -c \sum_l \nabla_{\theta} \hat{L}(\theta^t; \tilde{X}, Y) \tag{16}
\]

\[
\frac{\partial X_A}{\partial \theta} \approx -c \sum_l \nabla_{\theta} \hat{L}(\theta^t; \tilde{X}, Y) \tag{17}
\]

where \( \nabla_{\theta} \hat{L}(\theta^t; \tilde{X}, Y) \) can be calculated during the training process. The third term \( \nabla_{\theta} X_A \) can be approximated during the adversarial learning process of \( X \) as

\[
\tilde{X}^{t+1} = \tilde{X}^t + c_2 \nabla_{\tilde{X}} \left\{ L(\theta; \tilde{X}, Y) - \lambda c_{\omega}(\tilde{X}^t, X) \right\} \tag{18}
\]

\[
\frac{\partial \tilde{X}^{t+1}}{\partial w} = \frac{\partial \tilde{X}^t}{\partial w} - 2c_2 \lambda \text{Diag}(\tilde{X}^t - X) \tag{19}
\]

\[
\frac{\partial \tilde{X}}{\partial w} \approx -2c_2 \lambda \sum_l \text{Diag}(\tilde{X}^t - X) \tag{20}
\]

which can be accumulated during the adversarial training process.

### 4.2.2 Approximation Precision

We approximate the \( \partial \theta/\partial \tilde{X} \) and \( \partial \tilde{X}/\partial w \) during the gradient descent and ascent process, where we use the average gradient as the approximate value. To better quantify the precision of our approximation, we tested the reliability of our approximation empirically. Since the gradient represents the direction to which the function declines fastest, we compare the \( \Delta R \) after updating by our \( \partial \theta/\partial w \) with that after randomly selected directions with the same step size. Note that the \( \Delta R \) brought by the accurate gradient is largest among any other directions. Therefore, the higher possibility that our \( \Delta R \) is larger than randomly picked direction, the more accurate our approximation is. We perform random experiments for 1000 runs, and the approximation of our SAL outperforms 99.4% of them, which validates the high precision of our approximation.

### 5 Theoretical Analysis

Here we first provide the robustness guarantee for our method, and then we analyze the rationality of our uncertainty set, which also demonstrates the uncertainty set built in our SAL is more practical. And we finally derive the generalization bounds for our method.

#### 5.1 Robustness Guarantee

Recall that the original objective of this work is to optimize for the worst-case error in a distribution set, which is given as \( \min_{\theta, c} \sup_{Q: \mathcal{E}(\theta, Q) \leq \epsilon} \mathbb{E}[\ell(\theta)] \). However, for tractable optimization in section 4.1, we have to give up the prescribed amount \( \rho \) of the distributional robustness and focus on the relaxed Lagrangian penalty function

\[
sup_{Q} \left\{ E_Q[\ell(\theta; (x, y))] - \lambda W_{\omega}(Q, P) \right\} \tag{21}
\]

Note that in equation 21, we do not impose any constraints (e.g., within a Wasserstein ball) on the \( Q \), which we optimize the equation 21 with respect to. Then a natural question is, can the relaxed Lagrangian reformulation, which we actually optimize, provide some kind of robustness guarantee? Or is it just an approximation? In this subsection, we derive the robustness guarantee for the relaxed Lagrangian reformulation to answer this question.

**Theorem 1 (Robustness Guarantee for Relaxed Lagrangian Reformulation).** For fixed \( \lambda \geq 0 \), define the transportation map \( T_n(\theta; z_0) = \arg \max_{z \in \mathcal{Z}} (\ell(\theta; z) - \lambda c_{\omega}(z_0, z)) \), and the empirical maximizer of the Lagrangian reformulation (equation (21)) is given as

\[
P_n^\lambda = \arg \max_Q \left\{ E_Q[\ell(\theta; (x, y))] - \lambda W_{\omega}(Q, P_n) \right\}. \tag{22}
\]

Then we denote the Wasserstein distance between the worst-case distribution \( P_n^\lambda \) and the training distribution \( P_n \) as \( \rho_n = W_{\omega}(P_n^\lambda, P_n) \), we have

\[
\sup_{P: W_{\omega}(P, P_n) \leq \rho_n} E_P[\ell(\theta; Z)] = E_{P_n}[\rho_n(\theta; Z)] + \lambda \rho_n \tag{23}
\]

**Proof.** By choosing \( \rho_n \) as \( \rho \) in Lemma 1, it is easy to prove under the strong duality. \( \Box \)

Theorem 1 justifies that our relaxed Lagrangian reformulation in optimization can exactly guarantee the distributional robustness inside a \( \rho_n \)-radius ball, which is, given \( \lambda \), our algorithm will find a distribution \( P_n^\lambda \), whose distance from the original \( P_n \) is \( \rho_n \), and we can guarantee that the learned \( P_n^\lambda \) is exactly the worst-case distribution in the \( \rho_n \)-radius ball centered at \( P_n \). The only difference from the direct optimization is that, we cannot guarantee the robustness for a pre-given quantity \( \rho \), while we use the Lagrangian parameter \( \lambda \) as a qualitative factor to control how much robustness to protect.

#### 5.2 Compactness of the Adversarial Set

Then we analyze the rationality of our method in theorem 2, where our major theoretical contribution lies on. As far as we know, it is the first analysis of the compactness of adversary sets in WDRL literature.

**Assumption 2.** Given \( \rho > 0 \), \( \exists Q_0, \in P_0 \) satisfies:

1. \( \forall \epsilon > 0, \| \inf_{\Pi(P_0, Q_0)} \mathbb{E}_{(x, z_0) \sim \Pi}[\ell(z_1, z_2)] \| \leq \epsilon \), we refer to the couple minimizing the expectation as \( M_0 \).
2. \( \mathbb{E}_{M \in \Pi(P_0, Q_0)} \mathbb{E}_{(z_1, z_2) \sim M}[\ell(z_1, z_2)] \geq \rho \), where \( \Pi(P_0, Q_0) \) means excluding \( M_0 \) from \( \Pi(P_0, Q_0) \).
(3) \( Q_{\#S} \neq P_{\#S} \), where \( S = \{ i : w^{(i)} > 1 \} \) and \( w^{(i)} \) denotes the \( i \)-th element of \( w \) and \( P_{\#S} \) denotes the marginal distribution on dimensions \( S \).

Assumption 3. Given \( \rho \geq 0 \) and \( c_w \), there exists distribution \( V \) supported on \( Z \neq \emptyset \) that

\[
W_w(V, P_{\#V}) = \rho. 
\]

Assumption 2 describes the boundary property of the original uncertainty set \( P_0 = \{ Q : W_e(Q, P_0) \leq \rho \} \), which assumes that there exists at least one distribution on the boundary whose marginal distribution on \( S \) is not the same as the center distribution \( P_0 \)'s and is easily satisfied. And Assumption 3 assumes that there exists at least one marginal distribution \( V \) whose distance from the original marginal distribution is \( \rho \), and it is easily satisfied. Based on these assumptions, we come up with the following theorem.

**Theorem 2 (Compactness).** Under Assumption 2, assume the transportation cost function in Wasserstein distance takes form of

\[
c(c(x_1, x_2)) = \|x_1 - x_2\|_1 \text{ or } c(x_1, x_2) = \|x_1 - x_2\|_2^2,
\]

Then, given observed distribution \( P_0 \) supported on \( \mathcal{Z} \) and \( \rho \geq 0 \), for the adversary set \( \mathcal{P} = \{ Q : W_e(Q, P_0) \leq \rho \} \) and the original \( P_0 = \{ Q : W_e(Q, P_0) \leq \rho \} \), given \( c_w \) where \( \min(w^{(1)}, \ldots, w^{(m)}) = 1 \) and \( \max(w^{(1)}, \ldots, w^{(m)}) > 1 \), we have \( \mathcal{P} \subset \mathcal{P}_0 \). Furthermore, under Assumption 3, for the set \( U = \{ |w^{(i)}| = 1 \} \), \( \exists Q_0 \in \mathcal{P} \) that satisfies \( W_w(P_{\#U}, Q_{\#U}) = \rho \).

Theorem 2 proves that the constructed uncertainty set of our method is smaller than the original. Intuitively, in adversarial learning paradigm, if stable covariates are perturbed, the target should also change correspondingly to maintain the underlying relationship. However, we have no access to the target value corresponding to the perturbed stable covariates in practice, so optimizing under an isotropic uncertainty set (e.g., \( P_0 \)) which contains perturbations on both stable and unstable covariates would generally lower the confidence of the learner and produce meaningless results. Therefore, from this point of view, by adding high weights on stable covariates in the cost function, we may construct a more reasonable and practical uncertainty set in which the ineffective perturbations are avoided.

Further, we theoretically analyze the property of learned covariate weights \( w \) in linear regression, including the optimal point of equation 3 and the reason why our method can to some extent mitigate the low confidence problem compared with the original WDRL. To begin with, we make further assumptions on the given multiple environments data.

**Assumption 4 (Data Heterogeneity).** Under Assumption 1, we further assume that \( 3\delta_S \geq 0, \delta_V > 0 \), such that:

1. \( \forall e \in \mathcal{E}, \min_{\theta_S} \mathcal{L}^\theta(e) - \min_{\theta_V} \mathcal{L}^\theta(e_S) \leq \delta_S \)
2. \( \forall \text{ linear model } \psi(X) = \theta_S^T S + \theta_V^T V \) \( \text{ with } \theta_V > 0, \exists e_V, e_S \in \mathcal{E}_V, \text{ such that } \mathcal{L}^\theta(e_S) - \mathcal{L}^\theta(e_V) > \delta_V, \text{ where } \theta_S \text{ denotes the linear parameters on stable covariates and } \theta_V \text{ on unstable covariates.} \)

Actually, Assumption 4 assumes that (1) the predicting performance with stable features or unstable features will not differ much; (2) using unstable features for prediction will hurt the model’s stability across different environments, since \( \mathbb{E}[Y | \mathcal{V}] \) may change greatly.

**Theorem 3 (Optimal \( \theta_w(w^*) \)).** Under Assumption 4, for \( \alpha > \frac{\delta_S}{\delta_V} \), the optimal point \( \theta_w(w^*) \) of equation 3 satisfies that \( \theta_V = 0 \) and \( w^*_V \). Further, choosing \( c(z_1, z_2) = \|z_1 - z_2\|_2 \), with \( \rho \rightarrow \infty, \rho^2/\|w^*_V\| \rightarrow 0 \) and \( w^*_V \rightarrow 1 \), the minimizer \( \theta_w \) of equation 2 will approach \( \theta^* \).

**Proof.** It is easy to prove the parameters of unstable features in \( \theta^* \) is 0 under Assumption 4. We move on to the property of \( w^* \). For \( c_w = \|w \odot (z_1 - z_2)\|_2 \), the equation 2 can be reformulated to (following [14]):

\[
\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \xi_i + \rho \sqrt{(-\theta, 1)}^2 \mathcal{D}(w(-\theta, 1)) \tag{25}
\]

Then with \( \rho \rightarrow \infty, \rho^2/\|w^*_V\| \rightarrow 0 \), \( \rho^2 w^*_V \rightarrow \infty \), it is easy to prove that \( \theta_V \rightarrow 0 \) and \( \theta_S^* = \arg\min_{\theta_S} \mathcal{L} \).

In Theorem 3, we analyze the properties of the optimal points of our method, which verifies that the learned covariate weights will greatly restrict the perturbations on stable features \( w^*_S \rightarrow \infty \) to mitigate the over-pessimism problem. Although the scenario is simple, we can also get inspirations why the original WDRL faces the low confidence problem. From the reformulation in equation 25, we see that WDRL regulates the predictor with \( \|(-\theta, 1)\|_2 \), by letting \( w = 1 \) and the strength of regularization is controlled by the radius \( \rho \) of the ball. As \( \rho \) grows to contain more potential testing distributions, WDRL puts much more penalty on the parameters of both stable features and unstable features, which lowers both \( \theta_S \) and \( \theta_V \) until they are both 0, making in the model refuse to make predictions and only output 0, that is the origin of low confidence or over-pessimism. While in our proposed method, we use the learned covariate weights \( w \) to prevent the parameters \( \theta_S \) of stable features from being affected, and such desired weight can be learned via equation 3 as shown in Theorem 3.

### 5.3 Generalization Bounds

First, we provide the robustness guarantee in theorem 4 with the help of lemma 1 and Rademacher complexity[41].

**Theorem 4 (Generalization Bounds).** Let \( \Theta = \mathcal{R}_m \), \( x \in \mathcal{X} \), \( y \in \mathcal{Y} \). Assume \( |\ell(\theta; z)| \) is bounded by \( T_i \geq 0 \) for all \( \theta \in \Theta \), \( z = (x, y) \in \mathcal{X} \times \mathcal{Y} \). Let \( F : \mathcal{X} \rightarrow \mathcal{Y} \) be a class of prediction functions, then for \( \theta \in \Theta, \rho \geq 0, \lambda \geq 0 \), with probability at least \( 1 - \delta \), for \( P \in \{ P : W_e(P, P_0) \leq \rho \} \), we have

\[
\sup_{P} \mathbb{E}_{P}[\ell(\theta; Z)] \leq \mathbb{E}_{P} + \mathbb{E}_{P_h}[\lambda_{\Theta}] + \mathbb{R}_{n}(\ell \circ F) + kT_i \mathbb{R}_{n}(1/\delta)/n \tag{26}
\]

where \( \ell \circ F = \{ (x, y) \mapsto \ell(f(x), y) - \ell(0, y) : f \in F \} \) and \( \mathbb{R}_{n} \) denotes the Rademacher complexity[41] and \( k \) is a numerical constant no less than 0.

**Proof.** From lemma 1, for all \( \lambda \geq 0, \rho \geq 0 \), we have

\[
\sup_{P : W_e(P, P_0)} \mathbb{E}_P[\ell(\theta; X, Y)] \leq \rho \mathbb{E}_{P_0}[\lambda_{\Theta}] + \mathbb{R}_{n}(\ell \circ F). \tag{27}
\]

Applying the standard results on Rademacher complexity[41], with probability at least \( 1 - \delta \), we have

**Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.**
\[ E_P[s_i] \leq E_{P_i}[s_i] + R_n(I \circ F) + kT_i \sqrt{\frac{\ln(1/\delta)}{n}} \]  \tag{28} 

then combing with equation 27, the result follows. \(\square\)

Since the Rademacher complexity \(R_n\) also requires the expectation over sample distribution, we further derive the bound of the Rademacher complexity in theorem 4 which only depends on empirical data points. We introduce the definition of \(\epsilon\)-cover and \(\epsilon\)-covering number as follows, which can be used to measure the size of continuous sets.

**Definition 2 (\(\epsilon\)-cover).** \(C \subseteq U\) is an \(\epsilon\)-cover of a functional class \(G \subseteq U\) if and only if for all \(g \in G\), there exists some \(h \in C\) such that \(d_n(g, h) \leq \epsilon\), where \(d_n(\cdot, \cdot)\) is function distance metric defined with respect to a tuple of data points \((z_1, \ldots, z_n) \in \mathbb{R}^d\) as

\[ d_n(g, h) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (g(z_i) - h(z_i))^2}. \]  \tag{29} 

**Definition 3 (\(\epsilon\)-covering number).** The \(\epsilon\)-covering number of a function class \(G\) is defined as

\[ N(G, \epsilon, d_n(\cdot, \cdot)) = \inf \{|C| : C \text{ is an}\ \epsilon\text{-cover of } G\}, \]  \tag{30} 

where \(d_n(\cdot, \cdot)\) denotes the function distance metric as equation 29.

Then we derive the bound of Rademacher complexity \(R_n\) with respect to the \(\epsilon\)-covering number.

**Theorem 5 (\(\hat{R}_n\)).** For the Rademacher complexity in theorem 4, for function set \(G\) and assume that \(\forall g \in G, g : Z \rightarrow \mathbb{R}\) is a function and is bounded by \(T_i \geq 0\), with probability at least \(1 - \delta\), we have

\[ R_n(G) \leq \hat{R}_n(G) + 2T_i \sqrt{\log 1/\delta/2n}. \]  \tag{31} 

**Proof.** Easy to prove with bounded difference inequality. \(\square\)

Finally, we would like to derive the bound for \(\hat{R}_n\) with \(\epsilon\)-covering number.

**Theorem 6.** (Bound of \(\hat{R}_n\)) For function class \(G : Z \rightarrow \mathbb{R}\) containing functions \(G : Z \rightarrow \mathbb{R}\), we have

\[ \hat{R}_n(G) \leq \inf_{\epsilon > 0} \left\{ 4\epsilon + 12 \int_{\epsilon}^{\sup_{G \in G} \sqrt{E[G^2]}} \sqrt{\log N(G, \tau, d_n(\cdot, \cdot))} n d\tau \right\}. \]  \tag{32} 

Specifically, assume that \(\forall G \in G : Z \rightarrow \mathbb{R}, |G|\) is bounded by \(T_i \geq 0\), we have

\[ \hat{R}_n(G) \leq \inf_{\epsilon > 0} \left\{ 4\epsilon + 12 \int_{\epsilon}^{T_i} \sqrt{\log N(G, \tau, d_n(\cdot, \cdot))} n d\tau \right\}. \]  \tag{33} 

**Proof.** Let \(\tau_0 = \sup_{G \in G} \sqrt{E[G^2]}\) and for any \(j \in \mathbb{Z}_+\) let \(\tau_j = 2^{-j}\tau_0\). For each \(j\), let \(C_j\) be a \(\tau_j\)-cover of \(G\) with respect to \(d_n(\cdot, \cdot)\). For each \(G \in G\) and \(j\), pick an \(\tilde{G}_j \in C_j\) such that \(\tilde{G}_j\) is an \(\alpha_j\) approximation of \(G\). Then for \(N \in \mathbb{Z}_+\), \(G\) can be expressed as \(G = G - \tilde{G}_N + \sum_{1}^{N} (\tilde{G}_N - \tilde{G}_N-1)\) where \(\tilde{G}_0 = 0\). Then for any \(N\), we have

\[ \hat{R}_n(G) = \frac{1}{n} \int_{\tau_0}^{\tau_N} \sqrt{\log N(G, \tau, d_n(\cdot, \cdot))} n d\tau. \]  \tag{34} 

Then apply Massart’s finite class lemma to function classes \(\{f - f' : f \in C_j, f' \in C_j\}\) for each \(j\), we have for any \(N\) that

\[ \hat{R}_n(G) \leq \tau_N + 12 \int_{\tau_0}^{\tau_N} \frac{\sqrt{\log N(G, \tau, d_n(\cdot, \cdot))} n d\tau}{n}. \]  \tag{35} 

Then for any \(\epsilon\), choose \(N = \sup\{j : \alpha_j > 2\epsilon\}\). We have \(\alpha_N \leq 4\epsilon\) and

\[ \hat{R}_n(G) \leq 4\epsilon + 12 \int_{\epsilon}^{\sup_{G \in G} \sqrt{E[G^2]}} \sqrt{\log N(G, \tau, d_n(\cdot, \cdot))} n d\tau. \]  \tag{36} 

Since \(\epsilon\) is arbitrarily chosen, we take an infimum over \(\epsilon\). \(\square\)

**Remark 1.** Merging Theorem 4, 5, 6 together, we obtain the final bound as

\[ \sup_{\theta} E_P[\ell(\theta; Z)] \leq \lambda \rho + E_P[s_\lambda(\theta; Z)] + kT_i \sqrt{\frac{\log (1/\delta)}{n}} + \inf_{\epsilon > 0} \left\{ 4\epsilon + 12 \int_{\epsilon}^{\sup_{G \in G} \sqrt{E[G^2]}} \sqrt{\log N(G, \tau, d_n(\cdot, \cdot))} n d\tau \right\}. \]  \tag{37} 

**6 Experiments**

In this section, we validate the effectiveness of our method on simulation data and real-world data.

**Baselines** We compare our proposed SAL with the following methods.

- **Empirical Risk Minimization (ERM)**
  \[ \min_{\theta} E_{P_0}[\ell(\theta; X, Y)]. \]  \tag{38} 

- **Wasserstein Distributionally Robust Learning (WDRL)**
  \[ \min_{\theta} \sup_{Q \in W(Q_0)} E_Q[\ell(\theta; X, Y)]. \]  \tag{39} 

- **Invariant Risk Minimization (IRM)**
  \[ \min_{\theta} \sum_{\epsilon \in \mathcal{Z}} \mathcal{L}^\epsilon + \lambda \| \nabla_w \mathcal{L}^w\|_{1.0}^2. \]  \tag{40} 

For completeness, we also compare with LASSO [42], and Ridge regression [43].
For ERM and WDRL, we simply pool the multiple environments data for training. For fairness, we search the hyper-parameter $\lambda$ in $\{0.01, 0.1, \ldots, 1e0, 1e1, \ldots, 1e4\}$ for IRM and the hyper-parameter $\rho$ in $\{1.5, 10, 20, 50, 80, 100\}$ for WDRL. And we search the hyper-parameters $\lambda$ for LASSO and Ridge in $\{1e-3, 1e-2, \ldots, 1e-1, 1e1\}$. The best hyper-parameter is selected according to the validation set, which is sampled i.i.d. from the training environments.

**Kinds of Distributional Shifts.** To demonstrate the superiority of our methods, we design two typical kinds of distributional shifts, including selection bias[32], [33] and anti-causal effects[31]. In our simulation data, we introduce strong distributional shifts, where the spurious correlation between training and testing data varies a lot.

**Evaluation Metrics.** We use Mean_Error defined as $\text{Mean}_{-\text{Error}} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} e(x, y)$ and Std_Error defined as $\text{Std}_{-\text{Error}} = \sqrt{\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} (e(x, y) - \text{Mean}_{-\text{Error}})^2}$ which are the mean and standard deviation error across testing environments $e \in \mathcal{E}_t$.

**Imbalanced Mixture.** In our experiments, we perform a non-uniform sampling among different environments in training set which follows the natural phenomena that empirical data follow a power-law distribution. It is widely accepted that only a few environments/subgroups are common and the rest majority are rare[17], [44], [45].

### 6.1 Simulation Data

First, we design one toy example to demonstrate the over pessimism problem of conventional WDRL. Then, we design two mechanisms to simulate the varying correlations of unstable covariates across environments, named by selection bias and anti-causal effect.

**6.1.1 Toy Example**

In this setting, the goal is to predict $y \in \mathcal{R}$ from $x \in \mathcal{R}^d$, and we use $\ell(\theta; (x, y)) = |y - \theta^T x|$ as the loss function. We take $d = 2$ and generate $X = [S, V]^T$, where $S \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.5)$. We then generate $Y$ and $V$ as following

$$Y = 5 \ast S + S^2 + \epsilon_1, \quad V = \alpha Y + \epsilon_2,$$

where $\epsilon_1 \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1)$ and $\epsilon_2 \overset{\text{iid}}{\sim} \mathcal{N}(0, 1.0)$. In this experiment, the effect of $S$ on $Y$ stays invariant, but the correlation between $V$ and $Y$, i.e., the parameter $\alpha$, varies across environments. In training, we generate 180 data points with $\alpha = 1$ for environment 1 and 20 data points with $\alpha = -0.1$ for environment 2. We compared methods for linear regression across testing environments with $\alpha \in \{-2.0, -1.5, \ldots, 1.5, 2.0\}$.

We first set the radius for WDRL and SAL to be 20.0, and the results are shown in Fig. 2a. We find the ERM induces high estimation error as it puts high regression coefficient on $V$. Therefore, it performs poor in terms of prediction error under distribution shifts. While WDRL achieves more robust performances than ERM across environments, the prediction error is much higher than the others. Our method SAL achieves not only the smallest prediction error, but also the most robust performance across environments.

Furthermore, we train SAL and WDRL for linear regression with a varying radius $\rho \in \{0.0, 0.01, \ldots, 20.0\}$. From the results shown in Fig. 2b, we can see that, with the radius growing larger, the robustness of WDRL becomes better, but meanwhile, its performance maintains poor in terms of high Mean_Error and much worse than ERM ($\rho = 0$). This further verifies the limitation of WDRL with respect to the overwhelmingly-large adversary distribution set. In contrast, SAL achieves not only better prediction performance but also better robustness across environments. The plausible reason for the performance difference between WDRL and SAL can be explained by Fig. 2c. As the radius $\rho$ grows larger, WDRL tends to conservatively estimate small coefficients for both $S$ and $V$ so that the model can produce robust prediction performances over the overwhelmingly-large uncertainty set. Comparatively, as our SAL provides a mechanism to differentiate covariates and focus on the robustness optimization over unstable ones, the learned coefficient of unstable covariate $V$ is gradually decreased to improve robustness, while the coefficient of stable covariate $S$ does not change much to guarantee high prediction accuracy.

To better demonstrate the superiority of our proposed SAL, we further visualize the learned worst-case distribution of WDRL, out SAL compared with the observed data points in Fig. 3. From Fig. 3a, we can see that WDRL (green points) perturbs the observed data greatly along both the stable and unstable direction, while the learned perturbations of our SAL (red points) mainly focus on the unstable direction, which is similar to the ideal case. To better understand why the original distribution set of WDRL is undesirable, we draw Fig. 3b, which shows the relationship between the
stable covariate $S$ and the target $Y$. It show that WDRL (green points) greatly affects such stable relationship, while the proposed SAL does not hurt much, which is analogous to the ideal case. From Fig. 3c, we can see that our proposed SAL greatly perturbs the relationship between the unstable feature $V$ and target $Y$.

### 6.1.2 Selection Bias

In this setting, the correlations between unstable covariates and the target are perturbed through selection bias mechanism. We assume $X = [S, V] \in \mathcal{R}^p$ and $S = [S_1, S_2, \ldots, S_n] \in \mathcal{R}^n$ is independent from $V = [V_1, V_2, \ldots, V_n] \in \mathcal{R}^n$, while the covariates in $S$ are dependent with each other. According to assumption 1, we assume $Y = f(S) + \epsilon$ and $P(Y|S)$ remains invariant across environments while $P(Y|V)$ can arbitrarily change.

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

$$Z_1, \ldots, Z_d \sim \mathcal{N}(0, 1.0), V_1, \ldots, V_n \sim \mathcal{N}(0, 1.0)$$  \hspace{1cm} (45)

$$S_i = 0.8 * Z_i + 0.2 * Z_{i+1} \quad \text{for} \quad i = 1, \ldots, n_0 \hspace{1cm} (46)$$

To induce model misspecification, we generate $Y$ as

$$Y = f(S) + \epsilon = \theta_s * S^T + \beta * S_1 S_2 S_3 + \epsilon, \hspace{1cm} (47)$$

where $\theta_s = [\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3}, \frac{1}{3}, 1], \ldots \in \mathcal{R}^{n_0}$, and $\epsilon \sim \mathcal{N}(0, 0.3)$. As we assume that $P(Y|S)$ remains unchanged while $P(Y|V)$ 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 set $V_i \in V$ as

$$\hat{P} = \Pi_{v_i \in V_i} |r|^\gamma sgn(f(x) - sgn(r)v_i) \sim \text{Uni}(0, 1) \hspace{1cm} (48)$$

$$M(r; (x, y)) = \begin{cases} 1, & \mu \leq \hat{P} \\ 0, & \text{otherwise} \end{cases} \hspace{1cm} (49)$$

where $|r| > 1$ and $V_i \in \mathcal{R}^{n_0}$. 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 $v_i$ is close to its $y$ is more probably to be selected.) Intuitively, $r$ eventually controls the strength and direction of the spurious correlation between $V_i$ and $Y$, which is consistent with our analysis. IRM sacrifices the average performance of the model for the stability across environments, which might owe to its harsh requirements on the diversity of different training environments. Compared with other robust learning baselines, our SAL achieves nearly perfect performance with respect to average performance and stability, which reflects the effectiveness of assigning different weights to covariates for constructing the uncertainty set.

### 6.1.3 Illustration of the Confidence Problem

As mentioned above, WDRL is faced with the low confidence problem, which is also called the over-optimism problem. We conduct a classification experiment to directly show the confidence problem of WDRL as well as the superiority of our SAL. We make a slight modification to the selection bias setting and turn it into a classification problem. Specifically, we modify the generation of $Y$ as

$$Y = \text{sign}(\theta_s * S^T + \beta * S_1 S_2 S_3 + \epsilon), \hspace{1cm} (50)$$

where $\text{sign}(x) = 1_{x \geq 0}$. In this experiment, we set $n = 2000$, $\kappa = 0.95$, $p = 10$, $n_b = 1$ and compare the SAL with WDRL under radius of $\{1e - 2, 1e - 1, 1e 0, 1e 1\}$. The confidence of a binary classifier $f_b(.)$ is defined as the maximal prediction
TABLE 1
Results in Selection Bias Simulation Experiments of Different Methods With Varying Selection Bias $r$, Ratio $\kappa$, Sample Size $n$ and Unstable Covariates’ Dimension $n_b$ of Training Data, and Each Result is Averaged Over Ten Times Runs

Scenario 1: varying selection bias rate $r$ ($n = 2000, p = 10, \kappa = 0.95, n_b = 1$)

| Methods | $r = 1.5$ | $r = 1.7$ | $r = 2.0$ |
|---------|-----------|-----------|-----------|
| ERM     | 0.484     | 0.651     | 0.561     |
| LASSO   | 0.482     | 0.561     | 0.561     |
| Ridge   | 0.483     | 0.124     | 0.124     |
| WDRL    | 0.482     | 0.200     | 0.114     |
| IRM     | 0.475     | 0.015     | 0.477     |
| SAL     | 0.450     | 0.019     | 0.452     |

Scenario 2: varying ratio $\kappa$ and sample size $n$ ($p = 10, r = 1.7, n_b = 1$)

| $\kappa, n$ | $\kappa = 0.90, n = 500$ | $\kappa = 0.90, n = 1000$ | $\kappa = 0.975, n = 4000$ |
|-------------|--------------------------|---------------------------|-----------------------------|
| Methods     | Mean_Error | Std_Error | Mean_Error | Std_Error | Mean_Error | Std_Error |
| ERM         | 0.580      | 0.103     | 0.562      | 0.068     | 0.513      | 0.055     |
| LASSO       | 0.562      | 0.110     | 0.514      | 0.078     | 0.515      | 0.078     |
| Ridge       | 0.561      | 0.107     | 0.517      | 0.080     | 0.516      | 0.082     |
| WDRL        | 0.563      | 0.101     | 0.527      | 0.083     | 0.536      | 0.080     |
| IRM         | 0.460      | 0.014     | 0.464      | 0.015     | 0.459      | 0.014     |
| SAL         | 0.454      | 0.015     | 0.451      | 0.015     | 0.448      | 0.014     |

Scenario 3: varying ratio $\kappa$ and sample size $n$ ($p = 10, r = 2.0, n_b = 3$)

| $\kappa, n$ | $\kappa = 0.9, n = 1000$ | $\kappa = 0.95, n = 2000$ | $\kappa = 0.975, n = 4000$ |
|-------------|--------------------------|---------------------------|-----------------------------|
| Methods     | Mean_Error | Std_Error | Mean_Error | Std_Error | Mean_Error | Std_Error |
| ERM         | 0.440      | 0.069     | 0.466      | 0.097     | 0.489      | 0.133     |
| LASSO       | 0.433      | 0.059     | 0.460      | 0.095     | 0.482      | 0.124     |
| Ridge       | 0.434      | 0.061     | 0.457      | 0.095     | 0.481      | 0.124     |
| WDRL        | 0.433      | 0.058     | 0.459      | 0.095     | 0.481      | 0.122     |
| IRM         | 0.458      | 0.007     | 0.458      | 0.008     | 0.458      | 0.008     |
| SAL         | 0.415      | 0.019     | 0.411      | 0.015     | 0.411      | 0.016     |

TABLE 2
Results of the Classification Problem Under Selection Bias Setting

Classification under selection bias ($n = 2000, p = 10, \kappa = 0.95, n_b = 1$)

| Radius | 1e-2 | 1e-1 | 1e0 | 1e1 |
|--------|------|------|-----|-----|
| Methods | Acc | Conf | Acc | Conf | Acc | Conf | Acc | Conf |
| WDRL   | 0.765 | 0.702 | 0.581 | 0.585 | 0.377 | 0.529 | 0.361 | 0.504 |
| SAL    | 0.799 | 0.759 | 0.812 | 0.785 | 0.818 | 0.811 | 0.824 | 0.817 |

Accuracy denotes the average accuracy and Conf the confidence.

possibility assigned to classes

$$
\text{Conf} = E[\max(f_{\theta}(x), 1 - f_{\theta}(x))].
$$

(51)

We report the accuracy and confidence of SAL and WDRL in Table 2. As the radius of the uncertainty set increasing, the confidence of a WDRL classifier decreases sharply to 0.5, which means that the binary classifier cannot make a decision and it just randomly guess the answer.

6.1.4 Anti-Causal Effect

Inspired by [31], in this setting, we introduce the spurious correlation by using anti-causal relationship from the target $Y$ to the unstable covariates $V$. Assume $X = [S, V]^T \in \mathbb{R}^n$ and $S = [S_1, \ldots, S_n]^T \in \mathbb{R}^{n_b}$, $V = [V_1, \ldots, V_n]^T \in \mathbb{R}^n$, and the data generation process is as following

$$
S \sim \sum_{i=1}^{k} z_i \mathcal{N}(\mu_i, I), Y = \theta_i^T S + \beta S_1 S_2 S_3 + \mathcal{N}(0, 0.3)
$$

(52)

$$
V = \theta_i Y + \mathcal{N}(0, \sigma^2(\mu_i))
$$

(53)

where $\sum_{i=1}^{k} z_i = 1$ & $z_i > 0$ is the mixture weight of $k$ Gaussian components, $\sigma(\mu_i)$ means the Gaussian noise added to $V$ depends on which component stable covariates $S$ belong to and $\theta_i \in \mathbb{R}^{n_b}$. Intuitively, in different Gaussian components, the corresponding correlations between $V$ and $Y$ are varying due to the different value of $\sigma(\mu_i)$. The larger the $\sigma(\mu_i)$ is, the weaker correlation between $V$ and $Y$ is.

We use the mixture weight $Z = [z_1, \ldots, z_k]^T$ to define different environments, where different mixture weights represent different overall strength of the effect $Y$ on $V$. In this experiment, we set $\beta = 0.1$ and build 10 environments with varying $\sigma$ and the dimension of $S, V$, the first three for
In this experiment, we use a real-world dataset from Kaggle, a house sales price dataset from King County, USA, which includes the houses sold between May 2014 and May 2015. The target variable is the transaction price of the house and each sample contains 17 predictive variables such as the built year of the house, number of bedrooms, and square footage of home, etc. We normalize all predictive covariates to get rid of the influence by their personal details. We split the dataset into 10 environments randomly sampled from the original scales.

Scenario 1: \( n_s = 5, n_v = 5 \)

| Methods | \( e_1 \) | \( e_2 \) | \( e_3 \) | \( e_4 \) | \( e_5 \) | \( e_6 \) | \( e_7 \) | \( e_8 \) | \( e_9 \) | \( e_{10} \) |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| ERM     | 0.281   | 0.305   | 0.341   | 0.461   | 0.555   | 0.636   | 0.703   | 0.733   | 0.765   | 0.824   |
| LASSO   | 0.277   | 0.305   | 0.341   | 0.470   | 0.569   | 0.648   | 0.722   | 0.752   | 0.795   | 0.843   |
| Ridge   | 0.258   | 0.306   | 0.347   | 0.483   | 0.588   | 0.673   | 0.751   | 0.783   | 0.828   | 0.879   |
| IRM     | 0.287   | 0.293   | 0.329   | 0.345   | 0.382   | 0.420   | 0.444   | 0.461   | 0.478   | 0.504   |
| WDRL    | 0.282   | 0.331   | 0.399   | 0.599   | 0.750   | 0.875   | 0.983   | 1.030   | 1.072   | 1.165   |
| SAL     | 0.324   | 0.329   | 0.331   | 0.358   | 0.381   | 0.403   | 0.425   | 0.435   | 0.446   | 0.458   |

Scenario 2: \( n_s = 9, n_v = 1 \)

| Methods | \( e_1 \) | \( e_2 \) | \( e_3 \) | \( e_4 \) | \( e_5 \) | \( e_6 \) | \( e_7 \) | \( e_8 \) | \( e_9 \) | \( e_{10} \) |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| ERM     | 0.272   | 0.280   | 0.298   | 0.526   | 0.362   | 0.411   | 0.460   | 0.504   | 0.534   | 0.580   |
| LASSO   | 0.309   | 0.312   | 0.327   | 0.360   | 0.397   | 0.425   | 0.457   | 0.461   | 0.473   | 0.494   |
| Ridge   | 0.309   | 0.313   | 0.330   | 0.367   | 0.408   | 0.439   | 0.474   | 0.479   | 0.493   | 0.517   |
| IRM     | 0.306   | 0.312   | 0.325   | 0.328   | 0.343   | 0.358   | 0.365   | 0.374   | 0.377   | 0.394   |
| WDRL    | 0.299   | 0.314   | 0.332   | 0.545   | 0.396   | 0.441   | 0.483   | 0.529   | 0.555   | 0.596   |
| SAL     | 0.290   | 0.284   | 0.288   | 0.293   | 0.287   | 0.288   | 0.287   | 0.290   | 0.284   | 0.294   |

The average prediction errors of 15 runs are reported.

To test the stability of different algorithms, we simulate different environments according to the built year of the house. It is fairly reasonable to assume the correlations between parts of the covariates and the target may vary along time, due to the changing popular style of architecture. Specifically, the houses in this dataset were built between 1900 ~ 2015 and we split the dataset into 6 periods, where each period approximately covers a time span of two decades. In training, we train all methods on the first and second decade where built year \( \in [1900, 1910] \) and \( [1910, 1920] \) respectively and validate on 100 data points sampled from the second period.

From the results shown in Fig. 4a, we can find that SAL achieves not only the smallest \textit{Mean Error} but also the lowest \textit{Std Error} compared with baselines. From Fig. 4b, we can find that from period 4 and so on, where large distribution shifts occurs, ERM performs poorly and has larger prediction errors. IRM performs stably across the first 4 environments but it also fails on the last two, whose distributional shifts are stronger. WDRL maintains stable across environments while the mean error is high, which is consistent with our analysis in 6.1.1 that WDRL equally perturbs all covariates and sacrifices accuracy for robustness. From Fig. 4b, we can find that from period 3 and so on, SAL performs better than ERM, IRM and WDRL, especially when distributional shifts are large. In periods 1-2 with slight distributional shift, the SAL method incurs a performance drop compared with IRM and WDRL, while SAL performs much better when larger distributional shifts occur, which is consistent with our intuition that our method sacrifice a little performance in nearly I.I.D. setting for its superior robustness under unknown distribution shifts.

Classification. Finally, we validate the effectiveness of our SAL on an income prediction task. In this task we use the Adult dataset\cite{1} which involves predicting personal income levels as above or below $50,000 per year based on personal details. We split the dataset into 10 environments.
according to demographic attributes, among which distributional shifts might exist. In training phase, we train all methods on 693 data points from environment 1 and 200 points from the second respectively and validate on 100 points sampled from both. We normalize all the predictive covariates to get rid of the influence by their original scales. In testing phase, we test all methods on the 10 environments and report the mis-classification rate on all environments in Fig. 4c. From the results shown in Fig. 4c, we can find that the SAL outperforms baselines on almost all environments except a slight drop on the first. However, our SAL outperforms the others in the rest 8 environments where agnostic distributional shifts occur.

7 CONCLUSION

In this paper, we address a practical problem of overwhelmingly-large uncertainty set in robust learning, which often results in unsatisfactory performance under distributional shifts in real situations. We propose the Stable Adversarial Learning (SAL) algorithm that anisotropically considers each covariate to achieve more realistic robustness. We theoretically show that our method constructs a better uncertainty set and provide the theoretical guarantee for our method. Empirical studies validate the effectiveness of our methods in terms of uniformly good performance across different distributed data.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive suggestions and efforts to improve this paper.

REFERENCES

[1] H. Daume and D. Marcu, “Domain adaptation for statistical classifiers,” J. Artif. Intell. Res., vol. 26, no. 1, pp. 101–126, 2006.
[2] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2011, pp. 1521–1528.
[3] Z. Shen, P. Cui, T. Zhang, and K. Kuang, “Stable learning via sample reweighting,” 2019, arXiv:1911.12580.
[4] M. Kukar, “Transductive reliability estimation for medical diagnosis,” Artif. Intell. Med., vol. 29, no. 1–2, pp. 81–106, 2003.
[5] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, “Fairness in criminal justice risk assessments: The state of the art,” Social. Methods Res., vol. 50, no. 1, pp. 3–44, 2018.
[6] C. Rudin and B. Ustun, “Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice,” Interfaces, vol. 48, no. 5, pp. 449–466, 2018.
[7] B. Huval et al., “An empirical evaluation of deep learning on highway driving,” 2015, arXiv:1504.01716.
[8] A. Ben-Tal and A. Nemirovski, “Robust convex optimization,” Math. Operations Res., vol. 23, no. 4, pp. 769–805, 1998.
[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.
[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2017, arXiv:1706.06083.
[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proc. IEEE Eur. Symp. Secur. Privacy, 2016, pp. 372–387.
[12] N. Ye and Z. Zhu, “Bayesian adversarial learning,” in Proc. 32nd Int. Conf. Neural Inf. Process. Syst., 2018, pp. 6892–6901.
[13] A. Sinha, H. Namkoong, and J. Duchi, “Certifying some distributional robustness with principled adversarial training,” Proc. Int. Conf. Learn. Representations, 2018.
[14] P. M. Esfahani and D. Kuhn, “Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations,” Math. Program., vol. 171, no. 1/2, pp. 115–166, 2018.
[15] J. Duchi and H. Namkoong, “Learning models with uniform performance via distributionally robust optimization,” 2018, arXiv:1810.08750.
[16] C. Froger, S. Cläci, E. Chien, and J. Solomon, “Incorporating unlabeled data into distributionally robust learning,” 2019, arXiv:1912.07729.
[17] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, “Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization,” 2019, arXiv:1911.08731.
[18] C. Wah, S. Branson, L. Welinder, P. Perona, and S. Belongie, “The caltech-UCSD birds-200–2011 dataset,” 2011.
[19] A. Bhattacharjee, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth, “Big but imperceptible adversarial perturbations via semantic manipulation,” CoRR, 2019, arXiv:1904.06347.
[20] P. Vaishnavi, T. Cong, K. Eykholt, A. Prakash, and A. Rahmati, “Can attention masks improve adversarial robustness?,” 2019, arXiv:1911.11946.
[21] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[22] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” J. Statist. Plan. Inference, vol. 90, no. 2, pp. 227–244, 2000.
[23] S. Bickel, M. Brückner, and T. Scheffer, “Discriminative learning under covariate shift,” J. Mach. Learn. Res., vol. 10, pp. 2137–2155, 2009.
[24] M. Dudík, R. E. Schapire, and S. J. Phillips, “Correcting sample selection bias in maximum entropy density estimation,” in Proc. Adv. Neural Inf. Process. Syst., 2005, pp. 323–330.
[25] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, “Correcting sample selection bias by unlabeled data,” in Proc. Adv. Neural Inf. Process. Syst., Proc. 20th Annu. Conf. Neural Inf. Process. Syst., 2006, pp. 601–608.
[26] B. Fernando, A. Haddar, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” in Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 2960–2967.
[27] Y. Ganin and V. I. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in Proc. 32nd Int. Conf. Mach. Learn., 2015, pp. 1180–1189.
[28] K. Muandet, D. Balduzzi, and B. Schölkopf, “Domain generalization via invariant feature representation,” in Proc. 30th Int. Conf. Mach. Learn., 2013, pp. 1–10–1–18.
Zheyan Shen received the BS degree from the Department of Computer Science and Technology, Tsinghua University, in 2017. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Tsinghua University. His research interests include causally-regularized machine learning and social dynamics modeling. He has published more than 100 papers in prestigious conferences and journals in data mining and multimedia. His recent research won the IEEE Multimedia Best Department Paper Award, SIGKDD 2015 Best Paper Finalist, ICDM 2015 Best Student Paper Award, SIGKDD 2014 Best Paper Finalist, IEEE ICME 2014 Best Paper Award, ACM MM12 Grand Challenge Multimodal Award, and MMM13 Best Paper Award. He is PC co-chair of CIKM2019 and MMM2020, SPC or area chair of WWW, ACM Multimedia, IJCAI, AAAI, etc., and associate editors of IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Big Data, ACM Transactions on Intelligent Systems and Technology, and ACM Transactions on Multimedia Computing, Communications, and Applications, etc. He received ACM China Rising Star Award, in 2015, and CCF-IEEE CS Young Scientist Award, in 2018. He is now a distinguished member of ACM and CCF.

Linjun Zhou received the BE degree from the Department of Computer Science and Technology, Tsinghua University, in 2016. He is currently working toward the PhD degree with the Lab of Media and Network, Department of Computer Science and Technology, Tsinghua University. His research interests include few-shot learning and robust learning.

Kun Kuang received the PhD degree with the Department of Computer Science and Technology, Tsinghua University, in 2019. He is an associate professor with the College of Computer Science and Technology, Zhejiang University. He was a visiting scholar with Stanford University. His main research interests include causal inference and causally-regularized machine learning. He has published more than 30 papers in major international journals and conferences, including SIGKDD, ICML, ACM MM, AAAI, IJCAI, IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Knowledge Discovery from Data, Engineering, and CIDM, etc.

Bo Li received the bachelor’s degree in mathematics from Peking University, and the PhD degree in statistics from the University of California, Berkeley. He is currently an associate professor with the School of Economics and Management, Tsinghua University. His research interests are business analytics and risk-sensitive Artificial Intelligence. He has published widely in academic journals across a range of fields including statistics, management science and economics.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.