Appendices

Rethinking Distribution Shifts: Empirical Analysis and Modeling for Tabular Data

Tianyu Wang, Jiashuo Liu, Peng Cui, Hongseok Namkoong

A Other Relevant Work **40**

B Details in Section 2 **41**

B.1 Details in Section 2.1 . . . 41

B.2 Details in Section 2.2 . . . 42

C Details in Section 3 **51**

C.1 Detailed Results of 10 Settings in Table 1 . . . 51

C.2 Detailed Results of Feasibility in 10 Settings . . . 58

C.3 Detailed Results of Controlled Shift Settings . . . 58

C.4 Difference between Shift Settings in C.3 and Settings in Table 1 . . . 60

D Details in Section 4 **61**

D.1 Definitions of Variables . . . 61

D.2 Scaling of Ambiguity Size . . . 62

D.3 Linear Regression Results for Other Settings . . . 63

D.4 Fine-grained Comparison between the Model Class and the Choice of the Ambiguity Set 63

D.5 (Non-)Linear Regression Results for Ablation Studies . . . 67

D.6 Training Details in Section 4.2 . . . 75

D.7 More Results in Section 4.2 . . . 76

E Details in Section 5.1 **77**

E.1 Feature Attributions in Section 5.1 . . . 77

E.2 Additional Results in other Settings besides Sections 5.1.1 and 5.1.2 . . . 78

E.3 Additional Results in Section 5.1.3 . . . 79

F Details in Section 5.2 **79**

F.1 Further Discussions for Algorithm 1 . . . 79

F.2 Alternative Approach of Identifying Covariate Region . . . 81

F.3 Other Results . . . 82

A Other Relevant Work

Model classes in DRO methods. In Table 7, we summarize the model class of recent distance-based DRO methods that are considered in literature:

Table 7. Model classes of distance-based DRO methods that are explicitly considered in literature for classification ML models, where the models are exact reformulation unless specified by "(Apx)".

Reference	Ambiguity Set	Model class
[37]	1-Wasserstein	Linear SVM
[91], [19]	1(2)-Wasserstein	LR
[92]	1-Wasserstein	Kernel SVM, NN (Apx)
[97]	p-Wasserstein	NN (Apx)
[93]	General (OT) Wasserstein	Linear SVM, LR
[14], [11]	(standard) f-divergence	Linear SVM, LR
[56] / [85] / [52] / [33]	TV / CVaR / KL / χ^2-divergence	Linear SVM, LR
[65]	CVaR, χ^2-divergence	NN (Apx)
[101]	Sinkhorn distance	Linear SVM, LR, NN (Apx)
[15]	Holistic Robust	Linear SVM
[16]	Holistic Robust	NN (Apx)
[22]	OT-Discrepancy	Linear SVM

Active Learning. Active learning aims to improve model performance by acquiring a limited number of labels from the target distribution. See the survey [90] for a detailed reference. The challenge here is to quantify the value of unlabeled data so that we can select better samples. Selection criteria of existing approaches include estimated variance [29], influence on the model performance [68], Shapley value [42, 55], or distance metrics through the importance weighting [27]. Active learning has also been studied under joint distribution shifts and covariate shifts [103, 98]. These works usually assume restricted shift structures between the source and target distributions where only X-shifts [83, 98] or label shifts [111] occur, which may not hold in tabular data when $Y|X$-shifts occur. In contrast, we assume access to labeled target data but impose no restrictions on the two distributions. Then we integrate an efficient algorithm that identifies the covariate region suffering the largest $Y|X$-shifts between two distributions.

Difference with [45]. Compared with the benchmark paper [45], we highlight three major differences in our work:

1. Datasets and Domains: We focus on tabular datasets instead of the image datasets in [45]. The distribution shift patterns in the tabular datasets are more complicated due to $Y|X$-shifts

from unobserved variables. The existence of $Y|X$-shifts leads to more complex and volatile algorithmic behavior, as shown in Sections 3.1 and 3.2, whereas the shift patterns in image datasets are relatively simple since most of them are covariate shifts.

2. Methodologies: In terms of the training procedure, we consider DRO methods since they focus on worst-case distributions that may incorporate $Y|X$-shifts, and therefore, are of potential utility in the tabular data; In contrast, Gulrajani and Lopez-Paz [45] consider various domain generalization methods that build on neural networks and assume invariance across domains, and therefore, these methods are best suited to covariate shifts in the image datasets. In terms of the model classes, Gulrajani and Lopez-Paz [45] limit their analysis to deep networks with 9 methods in all, whereas our benchmark spans 45 methods – including linear models, tree-based ensembles, and neural networks – to reflect the reality that simpler models often outperform very deep architectures on tabular data.

3. Results and Insights: By including this broader range of model classes, we can disentangle the relative impact of algorithmic interventions (such as ambiguity-set design) from model-class effects. While we corroborate findings in [45] regarding which design components matter most, our expanded benchmark also uncovers simple, shift-aware interventions – both algorithmic and data-centric – that yield additional performance gains.

B Details in Section 2

B.1 Details in Section 2.1

DISDE to X-shifts and $Y|X$-shifts. When facing performance degradation under distribution shifts, one direct idea is to figure out the reasons why the performance drop. To this end, Cai et al. [26] propose DIstribution Shift DEcomposition (DISDE) to attribute the total performance degradation to $Y|X$-shifts and X-shifts. Specifically, given samples (X, Y) from distributions P and Q, to quantify the discrepancy between $P_{Y|X}$ and $Q_{Y|X}$, they first control the marginal distribution on $\mathcal{X}$ by introducing the shared distribution S_X. From that, we can estimate the performance degradation caused by $Y|X$-shifts and that caused by X-shifts could also be estimated by comparing S_X with P_X and Q_X, respectively. Note that DISDE could be used in image datasets (see Section 4.2 in [26]). The official code for DISDE could be found at `https://github.com/namkoong-lab/disde`. We specify the formula of DISDE as follows:

$$\begin{aligned}\mathbb{E}_Q[\ell(f_P(X), Y)] - \mathbb{E}_P[\ell(f_P(X), Y)] = \quad & \mathbb{E}_{S_X}[R_P(X)] - \mathbb{E}_P[R_P(X)] && \text{(I)}\\ & + \mathbb{E}_{S_X}[R_Q(X) - R_P(X)] && \text{(II)}\\ & + \mathbb{E}_Q[R_Q(X)] - \mathbb{E}_{S_X}[R_Q(X)], && \text{(III)}\end{aligned}$$

where $R_\mu(x) := \mathbb{E}_\mu[\ell(f_P(X), Y)|X = x]$ for $\mu = P, Q$ is denoted as the conditional risks on P and Q. S_X is the share distribution with support contained in both P_X and Q_X. *Then we see the sum of the two terms (I) and (III) as the performance drop attributed to X-shifts and the term (II) as the performance drop attributed to $Y|X$-shifts.* Besides, we also implement DISDE in our released package named WhyShift, which could be found at `https://github.com/namkoong-lab/whyshift`.

Attribution of Performance Drop. We use XGBoost classifier and calculate the decomposition of performance degradation via DISDE. We calculate the **$Y|X$-shift ratio** from the source distribution P to the target distribution Q:

$$Y|X\text{-shift ratio} = \frac{\mathbb{E}_{S_X}[R_Q(X) - R_P(X)]}{\mathbb{E}_Q[\ell(f_P(X), Y)] - \mathbb{E}_P[\ell(f_P(X), Y)]},$$

where $R_\mu(X)$ and S_X are defined above.

B.2 Details in Section 2.2

In this section, we provide the details of the methods in our benchmark.

B.2.1 Further Details in DRO Methods

Recall we consider the general DRO problem in (2.4) in Section 3 with the ambiguity set $\mathcal{P}(d, \epsilon)$. Here, unless specified, $\mathcal{P}(d, \epsilon) := \{P : d(P, \widehat{P}) \leq \epsilon\}$ denotes the ambiguity set around the training empirical distribution $\widehat{P}$ and $d(\cdot, \cdot)$ is a notion of distance between probability measures.

We describe the DRO methods into three main categories of distances used in DRO models: Wasserstein, Generalized f-divergence, and mixed distances due to their wide usage.

Wasserstein Distance. In the standard **Wasserstein-DRO** method [20], we apply $d(P, Q) = W_c(P, Q)$, where W_c is the Wasserstein distance:

$$W_c(Q, P) = \min_{\pi \in \Pi(Q,P)} \mathbb{E}_{Z_1, Z_2 \sim \pi} \left[c(Z_1, Z_2)\right], \tag{B.1}$$

where the induced cost function inside the distance is given by:

$$c((X_1, Y_1), (X_2, Y_2)) = (X_1 - X_2)^\top \Lambda (X_1 - X_2).$$

Unless specified, Λ is taken as the unit matrix and $c(\cdot, \cdot)$ then becomes squared Euclidean distance. That is to say, we only allow the perturbation of X but not Y, where Theorem 1 in [20] provides the corresponding tractable reformulations.

For the **Augmented Wasserstein-DRO** (Aug.Wass.-DRO) method [92], we still use $d(P, Q)$ being the Wasserstein distance W_c in (B.1) but now we allow the changes in y. More specifically, we

now set:

$$c((X_1, Y_1), (X_2, Y_2)) = (X_1 - X_2)^\top \Lambda (X_1 - X_2) + \kappa |Y_1 - Y_2|,$$

where κ (set as 1) is another hyperparameter to control the change in Y. Note that when $\kappa = \infty$, it reduces to the previous standard case, where [92] provides various loss functions to reformulate the problem.

For the **Satisficing Wasserstein-DRO** (Satis.Wass.-DRO) method [69], we solve the following constrained optimization problem, where DRO is set in the constraint counterpart:

$$\max \left\{ \|\theta\|_{\Sigma^{-1/2}}, \quad \text{s.t. } \mathbb{E}_{(X,Y)\sim P}[\ell_{tr}(\theta; (X, Y))] \leq \tau + \epsilon W_c(P, \widehat{P}), \forall P \right\}.$$

In the formulation above, we do not set ϵ as the hyperparameter but as an optimization goal such that the minimized worst-case performance is still no worse than some target quantity τ, which is a new hyperparameter and selected as the multiplication (i.e., the so-called *target ratio*, > 1) of the best empirical performance with $E_{(X,Y)\sim\widehat{P}}[\ell(\widehat{\theta}; (X, Y))]$ with $\widehat{\theta}$ obtained in the corresponding (2.3). To solve this optimization problem, we prefix $\epsilon \in [0, 100]$ and compute the corresponding left-hand objective value at each binary search to see if it reaches the goal τ and then reduce the potential range of ϵ half by half.

Generalized f-divergence. When d is set as the generalized f-divergence (including CVaR), all distances there can be formulated as follows:

$$d(P, Q) = E_Q \left[f\left(\frac{dP}{dQ} \right) \right].$$

For the **KL-DRO** method [52], we apply $f(x) = x \log x - (x - 1)$;

For the **χ^2-DRO** method [33], we apply $f(x) = (x - 1)^2$;

For the **TV-DRO** method [56], we apply $f(x) = |x - 1|$;

In each of the three methods above, the hyperparameter is ϵ as the uncertainty set size.

For the (joint) **CVaR-DRO** problem [85], we apply $f(x) = 0$ if $x \in [\frac{1}{\alpha}, \alpha]$ and ∞ otherwise (an augmented definition of the standard f-DRO problem). Here the hyperparameter α denotes the worst-case ratio. And we denote such induced d as the so-called *CVaR distance*.

Besides these standard (generalized) f-divergence DRO models, we also include DRO models modeling partial distribution shifts on (X, Y) based on CVaR metrics. Here, we directly use $\mathcal{P}(\alpha)$ as the ambiguity set where α is the CVaR parameter.

If we only consider the shifts in the marginal distribution X, we formulate the **Marginal-CVaR-DRO** model:

$$\mathcal{P}(\alpha) = \{Q_0 : P_X = \alpha Q_0 + (1 - \alpha) Q_1, \text{for some } \alpha \geq \alpha_0 \text{ and distribution } Q_1 \text{ and } \mathcal{X}\}.$$

Specifically, we follow the formulation of (27) in [34] to fit the model.

If we consider the shift in the conditional distribution $Y|X$, we formulate the **Conditional-CVaR-DRO** model:

$$\mathcal{P}(\alpha) = \{Q_0 : P_{Y|X} = \alpha Q_0 + (1-\alpha)Q_1, \text{for some } \alpha \geq \alpha_0 \text{ and distribution } Q_1 \text{ and } \mathcal{Y}\}.$$

Specifically, we follow the formulation of Theorem 2 in [87] to fit the model where approximating $\alpha(x) = \theta^\top x$.

Mixed Distances. We include other ambiguity set design $\mathcal{P}$ induced by some mixed distances including **Sinkhorn-DRO** [101], **Holistic-DRO** [15] (Theorem 3.4 there), **Unified-DRO** (with L_2 and L_∞ norm corresponding to $p = 2, \infty$ norm in their Theorem 5.2) [22]. We follow their initial github codebases and hyperparameter selection when implementing these methods. We also consider a class of DORO methods [108], which discards a proportion ϵ of the largest error points in each iteration to mitigate the outliers in DRO. Specifically in DORO, we consider the ambiguity set as:

$$\mathcal{P} = \{P : d(P, Q) \leq f_d(1/\alpha), d_{TV}(Q, \widehat{P}) \leq \epsilon\}. \tag{B.2}$$

Here in DORO, we consider **CVaR-DORO** and **χ^2-DORO** where d in (B.2) is set as CVaR-distance and χ^2-divergence respectively, where $f_d(t) = (t-1)^2/2$ when d is χ^2-divergence and $f_d(t) = 1$ when d is CVaR distance.

Other DRO Setups. Besides these methods with performance reported in the paper, in our codebase, we also implement Bayesian-based DRO [94], Parametric DRO [54] where we replace the reference measure Q to be parametric distribution (Gaussian mixture distribution); MMD-DRO [112], where we set $d(P, Q)$ as the kernel distance with the Gaussian kernel. However, the performance of these methods shows similar behavior compared with the benchmarks we use, so we do not report them here.

B.2.2 Further Details in Other Methods

Basic ERM methods. For LR and SVM, we use the standard implementation in `scikit-learn` [80] and train them on CPUs. For NN, we implement it via `PyTorch` [79] and train it on GPUs.

Tree-based ensemble methods. As shown by Gardner *et al.* [41], several tree-based methods achieve good performances on tabular datasets. And gradient-boosted trees (e.g., XGB, LGBM, GBM) are widely considered as the state-of-the-art methods on tabular data. Therefore, we compare XGB, LGBM, and GBM in this work. Also, we add RF to incorporate the performance of tree bagging methods. For RF and GBM, we use the standard implementations in `scikit-learn` [80]. For

XGB and LGBM, we use the standard implementations in the `xgboost` package[2] and the `lightgbm` package[3]. All these methods are trained on CPUs. For tree-based ensemble DRO methods, we directly apply the DRO-kind loss functions to XGBoost and LightGBM by changing the empirical losses implemented in `xgboost.train` and `lightgbm.train` to distributionally robust losses with CVaR and KL divergences respectively.

Imbalanced learning methods. Recently, some simple data balancing methods [53] have shown good worst-group performances under distribution shifts. In our benchmark, we implement 4 typical balancing methods, namely Sub-Sampling Y (SUBY), Reweighting Y (RWY), Sub-Sampling Group (SUBG), and Reweighting Group (RWG). For these imbalanced learning methods, we use XGB as the backbone model due to its superiority on tabular data and adjust sample weight or training procedure accordingly for each of the methods.

Fairness-enhancing methods. Following Ding et al. [31] and Gardner et al. [41], fairness-enhancing methods have the potential to mitigate the performance degradation under distribution shifts. In our benchmark, we evaluate the in-processing and post-processing intervention methods. The in-processing method [7] minimizes the prediction error subject to some fairness constraints, and in our benchmark, we choose three typical fairness constraints, including demographic parity (DP), equal opportunity (EO), error parity (EP). And the post-processing method [47] randomizes the predictions of a fixed classifier to satisfy equalized odds criterion, and we use exponential and threshold controls in our benchmark. We use the implementations of `aif360` [13] and `fairlearn` [17].

B.2.3 Parameter Search Space

We provide the hyperparameter grids in Table 8. We mainly use the hyperparameter grids proposed in [41], and restrict the grid size of each method in each setting to 200 due to computational costs. In each setting, we randomly pick 200 configurations for each algorithm for a fair comparison. For methods incorporating the underlying model class (e.g., NN2 / NN3 / NN4 / XGB), we choose the top 10 best configurations for that model class to reduce the search space, making the searched best configuration represent its best performance more accurately. Moreover, to accelerate the grid search process, we utilize `Ray` [66] to run experiments in parallel.

[2] `https://pypi.org/project/xgboost/`
[3] `https://pypi.org/project/lightgbm/`

Table 8: Hyperparameter grids for methods in Section 3 used in all experiments. $\diamond$: for methods with the total grid size above 200, we randomly sample *200* configurations for fair comparisons. For methods incorporating backbone models (e.g., NN/XGB), we choose top-10 best configurations for that backbone model to reduce the search space, making the searched best configuration represent its best performance more accurately.

Model	Total Grid Size	Hyperparameter	Value Range
Basic ERM methods			
NN2/NN3/NN4	$324^{\diamond}$	Learning Rate	$\{1e^{-3}, 3e^{-3}, 5e^{-3}, 1e^{-2}\}$
		Batch Size	$\{64, 128, 256\}$
		Hidden Units	$\{16, 32, 64\}$
		Dropout Ratio	$\{0, 1e^{-1}, 5e^{-1}\}$
		Train Epoch	$\{50, 100, 200\}$
SVM	96	C	$\{1e^{-2}, 1e^{-1}, 1, 1e^{1}, 1e^{2}, 1e^{3}\}$
		penalty	$\{L_1, L_2\}$
Kernel-SVM	$324^{\diamond}$	Nystroem components	{50, 100, 200}
		kernel	{rbf, sigmoid, polynomial}
		C	$\{1e^{-1}, 3e^{-1}, 1, 3, 1e^{1}, 3e^{1}, 1e^{2}, 3e^{2}, 1e^{3}\}$
		γ	{0.01, 0.1, 1, scale}
LR	23	L_2 penalty	$\{1e^{-3}, 3e^{-3}, 5e^{-3}, 7e^{-3}, 1e^{-2}, 3e^{-2}, 5e^{-2}, \ldots, 1.3, 1.7, 5\}$ $1e^{1}, 5e^{1}, 1e^{2}, 5e^{2}, 1e^{3}, 5e^{3}, 1e^{4}\}$
Tree-Base Ensemble Methods			
RF	$640^{\diamond}$	Num. Estimators	$\{32, 64, 128, 256, 512\}$
		Max Features	{sqrt, log2}
		Min. Samples Split	$\{2, 4, 8, 16\}$
		Min. Samples Leaf	$\{1, 2, 4, 8\}$
		Cost-Complexity α	$\{0, 1e^{-3}, 1e^{-2}, 1e^{-1}\}$
GBM	$1680^{\diamond}$	Learning Rate	$\{1e^{-2}, 1e^{-1}, 5e^{-1}, 1\}$
		Num. Estimators	$\{32, 64, 128, 256\}$
		Max Depth	$\{2, 4, 8, 16\}$
		Min. Child Samples	$\{1, 2, 4, 8\}$
LGBM	$1680^{\diamond}$	Learning Rate	$\{1e^{-2}, 1e^{-1}, 5e^{-1}, 1\}$
		Num. Estimators	$\{64, 128, 256, 512\}$
		L_2-reg.	$\{0, 1e^{-3}, 1e^{-2}, 1e^{-1}, 1\}$
		Min. Child Samples	$\{1, 2, 4, 8, 16, 32, 64\}$
		Column Subsample Ratio (tree)	$\{0.5, 0.8, 1.\}$
XGB	$1944^{\diamond}$	Learning Rate	$\{0.1, 0.3, 1.0, 2.0\}$
		Min. Split Loss	$\{0, 0.1, 0.5\}$
		Max. Depth	$\{4, 6, 8\}$
		Column Subsample Ratio (tree)	$\{0.7, 0.9, 1\}$
		Column Subsample Ratio (level)	$\{0.7, 0.9, 1\}$
		Max. Bins	$\{128, 256, 512\}$
		Growth Policy	{Depthwise, Loss Guide}
Linear-DRO Methods (Underlying Model Class: SVM)			
CVaR-DRO	117	Worst-case Ratio α	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
Conditional(-CVaR)-DRO	117	Worst-case Ratio α	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
KL-DRO	117	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
χ^2-DRO	117	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
TV-DRO	117	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
Wasserstein-DRO	138	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99, \ldots, 3\}$
Aug.Wass.-DRO	138	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99, \ldots, 3\}$
Marginal(-CVaR)-DRO	162	Worst-case Ratio α	$\{0.1, 0.2, \ldots, 0.8, 0.9\}$
		L	$\{1e^{-2}, 3e^{-2}, 5e^{-2}, 0.1, 0.3, 0.5, 1, 2, 5, 10\}$
		p	$\{1.5, 2\}$
Satis.Wass.-DRO	112	Target Ratio	$\{1.1, \ldots, 2.5, 3\}$
		κ	$\{1e^{-2}, 5e^{-2}, 1e^{-1}, 5e^{-1}, 1, 5, 1e^{7}\}$
Sinkhorn-DRO	$1156^{\diamond}$	Regularizer	$\{1e^{-3}, \ldots, 7e^{-2}, 1e^{-1}, \ldots, 9e^{-1}\}$
		k	$\{2, 3, 4, 5\}$
		λ	$\{1e^{-3}, \ldots, 7e^{-2}, 1e^{-1}, \ldots, 9e^{-1}\}$

Continued on next page

Table 8 – continued from previous page

Model	Total Grid Size	Hyperparameter	Value Range
Holistic-DRO	3969$^{\diamond}$	α	$\{1e^{-1}, \ldots, 9e^{-1}\}$
		r	$\{1e^{-1}, \ldots, 9e^{-1}\}$
		ϵ	$\{1e^{-3}, 1e^{-2}, 1e^{-1}, \ldots, 9e^{-1}\}$
		ϵ'	$\{1e^{-3}, 1e^{-2}, 1e^{-1}, \ldots, 9e^{-1}\}$
Unified-DRO	180	Distance Type	$\{L_2, L_{\text{inf}}\}$
		Uncertainty Set Size ϵ	$\{1e^{-3}, \ldots, 9e^{-1}\}$
		θ_1	$\{1.001, 1.01, 1.1, 1.5, 2, 3, 5, 10, 50, 100\}$
Kernel-DRO Methods (Underlying Model Class: Kernel-SVM)			
CVaR-DRO	4212$^{\diamond}$	Worst-case Ratio α	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
		Underlying model class	Kernel-SVM
KL-DRO	4212$^{\diamond}$	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
		Underlying model class	Kernel-SVM
χ^2-DRO	4212$^{\diamond}$	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99\}$
		Underlying model class	Kernel-SVM
Wasserstein-DRO	4968$^{\diamond}$	Uncertainty Set Size ϵ	$\{1e^{-4}, \ldots, 0.01, \ldots, 0.99, \ldots, 3\}$
		Underlying model class	Kernel-SVM
Tree-based DRO Methods (Underlying Model Class: LGBM / XGB)			
LGBM-CVaR-DRO	15120$^{\diamond}$	Worst-case Ratio α	$\{0.1, 0.2, \ldots, 0.8, 0.9\}$
		Underlying model class	LGBM
LGBM-KL-DRO	15120$^{\diamond}$	Uncertainty Set Size ϵ	$\{0.1, 0.2, \ldots, 0.8, 0.9\}$
		Underlying model class	LGBM
XGB-CVaR-DRO	17496$^{\diamond}$	Worst-case Ratio α	$\{0.1, 0.2, \ldots, 0.8, 0.9\}$
		Underlying model class	XGB
XGB-KL-DRO	17496$^{\diamond}$	Uncertainty Set Size ϵ	$\{0.1, 0.2, \ldots, 0.8, 0.9\}$
		Underlying model class	XGB
NN-DRO Methods (Underlying Model Class: NN)			
CVaR-DRO	1620$^{\diamond}$	Worst-case Ratio α	$\{0.01, 0.1, 0.2, 0.3, 0.5, 1.0\}$
		Underlying Model Class	NN2/NN3/NN4
χ^2-DRO	1620$^{\diamond}$	Uncertainty Set Size ϵ	$\{0.01, 0.1, 0.2, 0.3, 0.5, 1.0\}$
		Underlying Model Class	NN2/NN3/NN4
CVaR-DORO	8100$^{\diamond}$	Worst-case Ratio α	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$
		ϵ	$\{0.001, 0.01, 0.1, 0.2, 0.3\}$
		Underlying Model Class	NN2
χ^2-DORO	8100$^{\diamond}$	Worst-case Ratio α	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$
		ϵ	$\{0.001, 0.01, 0.1, 0.2, 0.3\}$
		Underlying Model Class	NN2
Imbalanced Learning Methods			
SUBY, RWY	1944$^{\diamond}$	Underlying Model Class	XGB
SUBG, RWG	1944$^{\diamond}$	Underlying Model Class	XGB
Fairness-Enhancing Methods			
In-processing	1944$^{\diamond}$	Constraint Type	{DP, EO, Error Parity}
		Underlying Model Class	XGB
Post-processing	1944$^{\diamond}$	Constraint Type	{Exp, Threshold}
		Underlying Model Class	XGB

B.2.4 Training and Validation Details in Section 3

Training. In each setting, we randomly sample 20,000 samples from the source domain for training and 20,000 samples from the target domain for testing. For `US Accident` and `Taxi` datasets, we only randomly sample 8,000 samples for the source/target domain due to fewer samples involved in each setting. In settings where the source domain does not have enough samples, we use 80%

samples from the source domain for training. Experiments of all settings were run on a server using 48 cores from two AMD EPYC 7402 24-Core Processors. NN-DRO methods were trained on GPU, NVIDIA GeForce RTX 3090, while the other methods were trained only on CPUs. Note that all experiments are small-scale and could be run efficiently. Besides, during training, we found that the bi-search in χ^2-DRO (NN) sometimes failed to converge, and therefore we set the maximal iteration number to 2500.

Hyperparameter Tuning. Unless specified, we selected the best configuration according to the performance on 128 samples selected from the target domain, i.e., the so-called **out-of-distribution accuracy**, where the validation set is identically distributed ($i.d.$) with the test set.

Comparison of Evaluation Pipelines. To highlight the suitability of our evaluation pipeline, we compare it with other alternative evaluation pipelines that utilize source and target data for basic ERM methods across different model classes, including LR, linear-SVM, XGB, LGBM:

(a) We train each ERM method on all samples from the source domain and tune the best hyperparameter of each ERM method based on the highest accuracy on samples from the source domain (in-distribution accuracy);

(b) We train each ERM method on all samples from the source domain and tune the best hyperparameter of each ERM method based on the highest accuracy on 128 samples from the target domain;

(c) We train each ERM method on all samples from the source domain and tune the best hyperparameter of each ERM method based on the highest accuracy on all samples from the target domain;

(d) We train each ERM method on 128 samples from the target domain.

where $(a), (b), (d)$ are based on the realistic setting with sufficient samples from the source domain and 128 samples from the target domain. We rename (b) and (d) as (b)-128 and (d)-128 respectively and also consider (b)-32 and (b)-64 ((d)-32 and (d)-64 respectively) that use 32 and 64 samples from the target domain to indicate the robustness of the selection criterion with respect to the number of samples from the target domain.

We evaluate the model performance on the `ACS Income` (Setting 1) dataset in Table 9. Rule (b)—training on the full source dataset and using 128 target samples for hyperparameter selection—outperforms alternative practical rules that train directly on 128 (or 64, 32) samples or the method that is validated using samples from the source domain. Moreover, its selected method performance closely matches that chosen by rule (c), which relies on oracle access to target-domain

Table 9. Accuracy and Radius of ERM methods with different training methods, where the accuracy and radius are computed averaged over all target domains in the setting.

	Method Performance							
	Accuracy				Macro F1-score			
Evaluation Approach	LR	SVM	XGB	LGBM	LR	SVM	XGB	LGBM
(a)	77.3±2.0	77.0±2.1	77.7±2.3	77.8±2.1	75.6±3.2	75.4±3.3	76.2±3.5	76.2±3.5
(b)-128	77.2±1.8	77.1±1.9	77.9±2.8	77.8±2.7	75.5±3.0	75.5±3.2	76.4±4.0	76.2±3.9
(b)-64	77.2±1.8	77.1±2.0	74.5±3.5	74.9±2.9	75.5±3.0	75.4±3.3	72.8±4.5	73.2±3.9
(b)-32	77.2±2.0	77.1±1.9	73.7±3.9	74.3±4.4	75.5±3.2	75.4±3.2	72.0±5.1	72.1±5.3
(c)	**77.6**±1.8	**77.2**±1.9	**78.3**±2.5	**79.2**±2.0	**75.9**±3.1	**75.5**±3.2	**76.5**±3.2	**76.7**±2.9
(d)-128	71.8±3.0	71.1±2.7	72.8±2.9	73.3±2.8	75.5±3.0	75.5±3.2	74.4±4.0	74.2±3.9
(d)-64	68.7±4.7	66.8±5.1	70.7±3.6	71.2±3.2	75.5±3.0	75.4±3.3	72.8±4.5	73.2±3.9
(d)-32	66.9±4.9	65.0±5.2	68.8±4.6	68.9±4.4	75.5±3.2	75.4±3.2	72.0±5.1	72.1±5.3

labels. These results validate our proposed learning scenario. In Appendix C.2, we further compare every method from Section 2 across all settings against the oracle ERM trained on the full target dataset, demonstrating that our experimental design is both challenging and practically achievable.

Comparison of DRO Validation Approaches. Furthermore, to highlight the suitability of our tuning method for DRO approaches, we compare validation approaches in several Wasserstein and f-divergence-based DRO methods under linear models since there are no other hyperparameter configurations and the required radii that contain the target distribution are easy to compute there compared with DRO under hybrid distances.

We compare the following five rules to select the radius in each DRO approach:

(a) We tune the best radius based on the highest accuracy on validation samples from the source domain;

(b) We tune the best radius based on the highest accuracy on 128 validation samples from the target domain, which is how we evaluate models in the main body;

(c) We tune the best radius based on the highest accuracy on all (20,000) samples from the target domain;

(d) We compute the required radius to ensure the target distribution is in the ambiguity set with a high probability using 128 validation samples from the target domain;

(e) We compute the required radius to ensure the target distribution is in the ambiguity set with a high probability using all samples from the target domain.

Above, the set of all the possible radii in (a) and (b) are from a fixed grid in Table 8 of Appendix B.2.3; and validation approaches (c) and (e) are impractical since they require access to all samples from

the target domain, and are included only for comparison. Then we detail how we set the radius to ensure the target distribution is in the ambiguity set with a high probability (i.e., the upper confidence bound principle). We compute the required radius of each method as follows:

- For Wasserstein-DRO, we follow the rule in Section 4 of [64];
- For f-divergence, besides training and validation samples, we assume access to a likelihood ratio function $r((x,y)) := \frac{dP((x,y))}{dQ((x,y))}$, which can be computed in advance using all samples from the source and the target domain. For CVaR, we set $\alpha = \frac{1}{\max_{i\in[m]} r((\tilde{x}_i,\tilde{y}_i))}$. Recall training samples $\{(x_i,y_i)\}_{i\in[n]}$ from the source domain P and $\{(\tilde{x}_j,\tilde{y}_j)\}_{j\in[m]}$ from the target domain Q. For other f-divergences (KL-divergence, χ^2-divergence and TV-Distance), we set the required radius that contains the true target distribution with probability at least $1-\delta$ by the central limit theorem approximation:

$$\epsilon = \frac{1}{m}\sum_{i\in[m]} f(r((\tilde{x}_i,\tilde{y}_i))) + z_{1-\frac{\delta}{2}} \frac{\sqrt{\sum_{i\in[m]}(f(r((\tilde{x}_i,\tilde{y}_i)) - \sum_{i\in[m]} f(r((\tilde{x}_i,\tilde{y}_i)))/m)^2}}{m},$$

  where $z_\cdot$ is the normal quantile. We set $\delta = 0.2$ in our setting to ensure the target distribution is in the ambiguity set with 80% probability.

Since we focus on problems with distribution shifts where a large number of samples are available from the source domain, the radius selection principles for problems without distribution shifts and limited training samples are no longer suitable [37, 19]. Therefore, we only compare the proper validation approaches under distribution shifts.

We report the selected radius and corresponding accuracy of each validation approach in Table 10 across settings with more than 10 target domains (i.e., Settings 1, 2, 4, 5). Unsurprisingly, the validation approach (c) achieves the highest accuracy across each setting. Compared with that, our proposed rule (b) yields a comparable accuracy and similar radius across each setting, indicating the stability of using 128 target samples to select the radius with the highest accuracy. In contrast, the validation approach (d) based on the upper confidence bound principle usually selects overly large radii, which explains its underperformance. Even when full access to samples from the target domain is granted in the approach (e), the overestimation issue persists. This is because choosing the radius to contain the true distribution for worst-case performance guarantees may yield an overly large ambiguity set and thus be overly conservative, particularly in the presence of distribution shifts. Finally, the in-sample validation approach (a) consistently leads to poor accuracy, underscoring the importance of incorporating samples from the target domain in the validation procedure.

Table 10. Accuracy and Radius of Linear DRO with different validation rules, where the accuracy and radius are computed averaged over all target domains in the setting and boldfaced values in each setting denote the one with the highest accuracy and corresponding radius.

	Accuracy					Radius				
Setting / Tuning Approach	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
KL-DRO										
1	76.8	77.9	**78.3**	76.7	76.7	0.060	0.730	**0.782**	3.265	2.773
2	73.3	74.1	**74.6**	73.2	73.2	0.220	0.197	**0.268**	0.669	0.411
4	75.7	76.6	**77.1**	75.0	75.1	0.310	0.289	**0.403**	0.634	0.512
5	72.6	72.9	**73.0**	68.9	69.0	0.002	0.016	**0.016**	6.217	5.023
CVaR-DRO										
1	75.8	78.2	**79.1**	77.2	77.1	0.010	0.087	**0.121**	0.228	0.142
2	73.3	74.5	**75.4**	71.7	60.6	0.310	0.166	**0.178**	0.174	0.035
4	74.4	76.9	**77.9**	73.6	74.0	0.120	0.172	**0.225**	0.149	0.132
5	72.9	73.5	**73.9**	68.8	68.8	0.390	0.342	**0.319**	0.127	0.066
χ^2-divergence-DRO										
1	76.8	77.9	**78.2**	77.2	77.3	0.040	0.954	**1.049**	5.501	5.319
2	73.4	74.6	**75.0**	73.4	73.4	0.780	0.635	**0.739**	1.492	1.583
4	75.8	76.8	**77.3**	75.2	75.2	1.200	0.826	**0.862**	1.486	1.412
5	72.7	72.8	**73.2**	68.5	68.5	0.005	0.485	**0.229**	8.933	8.859
TV-DRO										
1	76.3	78.0	**78.8**	73.0	72.5	0.860	0.426	**0.401**	0.585	0.539
2	73.1	74.2	**74.9**	73.2	73.1	0.840	0.624	**0.586**	0.418	0.404
4	75.2	76.1	**76.6**	72.4	72.7	0.001	0.458	**0.467**	0.451	0.439
5	72.6	73.4	**73.6**	68.4	68.2	0.006	0.193	**0.221**	0.875	0.816
Wasserstein-DRO										
1	77.3	78.5	**79.0**	65.2	65.2	0.000	0.203	**0.200**	7.558	4.927
2	73.2	73.8	**74.0**	72.7	72.7	0.000	0.016	**0.013**	6.466	6.466
4	75.3	75.4	**75.6**	70.1	70.1	0.001	0.001	**0.000**	5.054	3.628
5	72.3	73.0	**73.5**	57.1	57.1	0.001	0.010	**0.016**	6.651	5.047

C Details in Section 3

C.1 Detailed Results of 10 Settings in Table 1

In the main body, due to space limitations, we only visualize the target performances of different methods. Here we provide the detailed results of all algorithms on the 10 selected pairs in Table 1. For each method, we select the top-10 configurations according to the validation set from the corresponding target domain in each setting and report the mean accuracy (or macro F1-score) as well as the standard deviation in Table 11 (or Table 12 respectively) with respect to both *i.d.* and *o.o.d.* (out-of-distribution, i.e., the corresponding target) data.

Table 11. Results (Accuracy) of 10 selected pairs in Section 3.2, where we run each method with its top-10 configurations (according to the target F1-score) and report its mean accuracy and standard deviation.

Dataset		`ACS Income`		`ACS Mobility`		`US Taxi`		`ACS Pub.Cov`		`US Accident`		`ACS Time`		`Sub-Sampling`		`diabetes`		`assistments`		`college`	
Shift Pattern		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more		X more		X dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more	
Source → Target Pair		CA→PR		MS→HI		NYC→BOG		NE→LA		CA→OR		2010→2017		Young→Old		White→Others		700→1		Normal → Special	
		i.d.	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*
Basic Methods	LR	80.4±0.6	73.5±0.5	76.8±0.2	70.7±0.5	84.3±0.1	74.9±0.2	83.5±0.1	68.3±0.4	78.6±0.1	77.0±0.2	72.1±0.0	61.9±0.0	91.7±0.0	79.4±0.0	65.0±0.0	59.9±0.1	87.2±1.7	46.5±1.2	93.4±0.0	81.5±0.2
	SVM	80.7±0.0	72.8±0.0	77.9±0.1	71.2±0.1	83.3±0.1	75.6±0.1	83.5±0.1	67.8±0.1	78.8±0.1	77.2±0.1	77.1±0.0	68.7±0.1	91.5±0.1	79.6±0.2	64.9±0.0	59.8±0.0	88.3±0.0	45.7±0.1	93.4±0.0	81.8±0.1
	Kernel-SVM	74.6±4.1	74.1±5.0	65.7±7.3	66.7±5.0	66.5±11.4	69.4±1.7	66.5±7.1	61.0±2.8	79.8±2.6	69.6±4.6	60.4±4.9	58.5±1.9	90.9±1.0	77.3±1.1	64.6±0.5	60.5±0.3	88.3±7.9	61.2±2.0	87.4±3.9	72.5±3.9
	NN2	80.8±0.3	74.1±2.4	72.7±4.6	67.4±5.1	78.5±11.7	72.1±1.9	81.6±1.2	68.5±1.1	82.2±0.8	66.0±1.2	75.3±2.4	68.9±1.2	91.1±0.2	79.1±0.3	63.2±0.9	59.4±0.5	88.3±0.6	58.8±3.4	93.3±0.2	80.4±0.8
	NN3	80.3±0.6	73.0±3.7	75.9±0.5	77.9±0.8	84.6±0.6	73.0±0.9	81.2±1.0	66.3±1.0	82.2±1.7	66.1±1.0	77.1±0.6	69.3±0.9	90.9±0.4	79.1±0.6	63.0±0.9	59.2±0.4	88.6±0.2	59.8±2.9	93.4±0.2	81.5±0.8
	NN4	80.2±2.1	73.2±2.3	75.6±0.5	78.2±0.4	83.9±0.7	72.8±2.1	81.2±0.5	64.3±2.0	81.9±1.9	66.4±0.9	75.1±3.1	68.4±1.5	90.3±0.4	77.7±0.7	62.5±1.3	59.1±1.3	87.4±2.8	57.3±2.5	93.2±0.4	80.8±1.3
Tree-based Ensemble Methods	RF	75.5±0.2	76.0±0.8	79.9±0.4	71.7±0.4	85.6±0.1	73.2±0.2	85.4±0.3	68.5±0.2	85.3±0.4	67.9±0.3	78.3±0.9	70.7±0.4	92.1±0.1	80.5±0.2	65.0±0.6	61.3±0.6	88.6±0.5	64.7±1.3	94.3±0.1	84.2±0.4
	XGB	78.6±1.0	69.8±1.1	71.2±1.2	66.1±0.5	83.9±2.9	71.4±0.8	83.9±1.5	69.3±0.7	81.8±0.3	66.8±0.2	76.8±1.5	68.8±1.5	92.2±0.1	80.2±0.1	61.7±1.0	57.8±0.5	89.2±0.4	60.5±0.6	94.7±0.0	84.9±0.4
	GBM	80.2±1.7	73.8±2.6	65.3±8.4	64.9±3.6	84.5±1.1	72.3±0.7	84.9±0.9	70.1±0.7	84.0±3.7	67.0±1.7	77.4±1.6	70.1±1.2	92.1±0.1	80.6±0.1	63.7±0.0	60.9±0.0	89.1±0.4	62.3±1.3	94.6±0.1	84.7±0.5
	LGBM	79.5±0.7	75.7±3.9	74.7±0.8	67.6±1.1	81.2±2.4	72.8±2.9	84.6±0.6	69.9±0.3	79.7±3.2	67.1±1.2	77.0±1.3	69.2±1.1	92.1±0.1	80.2±0.1	62.1±1.7	58.4±1.2	88.1±0.4	62.4±2.1	94.7±0.1	84.4±0.7
Linear-DRO Methods (base: SVM)	KL-DRO	80.1±0.1	73.6±0.1	77.7±0.1	70.9±0.1	83.3±0.3	74.4±0.4	83.4±0.1	68.3±0.3	79.2±0.0	76.3±0.1	77.1±0.1	68.8±0.2	91.4±0.1	79.8±0.1	64.5±0.4	59.5±0.4	88.3±0.0	44.9±0.0	93.6±0.0	81.6±0.2
	CVaR-DRO	78.8±2.4	72.9±1.1	77.6±0.0	70.6±0.0	83.2±1.0	75.2±1.7	82.5±1.0	68.5±0.6	79.3±0.1	76.1±0.1	76.8±0.2	69.1±0.2	91.6±0.0	80.2±0.1	63.7±0.4	58.5±0.3	88.0±0.0	44.9±0.3	93.1±0.4	80.7±1.0
	χ^2-DRO	80.6±0.1	73.3±0.4	77.6±0.1	70.7±0.2	83.2±0.1	76.0±0.1	83.3±0.2	68.8±0.2	79.1±0.0	76.6±0.3	77.1±0.0	68.8±0.0	91.7±0.1	79.7±0.0	64.8±0.2	59.5±0.4	88.3±0.0	44.9±0.0	93.6±0.0	81.8±0.1
	TV-DRO	78.4±2.0	72.9±0.6	77.4±0.3	70.4±0.6	81.4±2.3	73.1±2.7	82.1±0.8	66.5±0.9	79.1±0.4	76.1±0.7	76.3±0.9	68.7±0.3	91.0±0.1	79.9±0.1	63.2±0.5	57.9±0.7	88.3±0.0	44.9±0.0	93.6±0.0	81.6±0.1
	Wasserstein-DRO	76.2±4.3	74.2±2.4	77.5±0.0	71.3±0.1	84.1±0.0	75.3±0.1	82.0±0.3	65.7±0.7	78.5±0.5	76.8±0.3	76.0±0.0	67.8±0.0	91.5±0.0	77.2±0.0	64.0±0.3	59.2±0.2	88.2±0.0	44.9±0.0	93.6±0.1	81.4±0.2
	Aug. Wass.-DRO	78.5±3.9	74.1±2.3	72.8±9.8	69.9±2.8	81.7±0.8	71.8±1.7	77.5±4.6	65.5±2.3	79.9±0.3	75.1±0.4	73.9±4.4	66.6±2.8	91.2±0.2	80.1±0.4	62.6±2.9	58.0±1.9	70.3±10.8	54.9±2.6	93.5±0.1	81.5±0.2
	Satis. Wass.-DRO	73.8±2.8	73.7±0.5	75.8±0.0	78.5±0.0	79.0±1.3	69.4±2.5	80.0±0.0	59.4±0.0	77.1±0.0	77.2±0.0	71.7±0.0	60.8±0.0	91.2±0.2	78.2±1.1	58.5±0.0	50.6±0.0	87.8±0.0	44.8±0.0	92.9±0.2	81.4±0.2
	Sinkhorn-DRO	73.5±1.9	85.5±1.7	68.7±3.9	69.6±5.0	69.7±7.3	75.0±1.3	81.7±1.4	67.8±1.3	79.1±0.4	76.2±0.5	56.7±12.2	58.2±7.2	91.4±0.1	79.7±0.1	62.8±0.4	58.2±0.7	85.8±2.9	44.6±0.3	91.6±1.6	79.1±1.7
	Unified-DRO(L_2)	80.6±0.0	72.7±0.0	77.5±0.0	71.3±0.0	83.3±0.0	75.2±0.1	82.0±0.0	65.8±0.0	77.9±1.0	76.9±0.2	76.0±0.0	67.8±0.0	91.2±0.0	79.7±0.0	62.6±0.0	57.8±0.0	87.6±1.7	44.8±0.2	93.5±0.0	81.5±0.0
	Unified-DRO(L_{inf})	79.9±0.9	72.8±0.1	77.5±0.0	71.3±0.1	83.6±0.0	75.5±0.0	82.2±0.0	66.0±0.0	77.6±0.7	77.2±0.2	76.0±0.0	67.8±0.0	91.9±0.0	79.3±0.0	63.2±0.0	58.7±0.0	86.6±4.6	44.7±0.4	93.6±0.0	81.5±0.0
	Marginal-DRO	80.7±0.1	73.2±0.2	77.9±0.3	70.8±0.2	82.7±0.1	76.0±0.1	83.5±0.3	69.0±0.3	79.2±0.1	76.0±0.2	77.2±0.0	69.0±0.0	91.3±0.1	79.0±0.3	64.7±0.2	60.4±0.4	88.3±0.0	56.9±0.0	93.4±0.0	81.3±0.1
	Conditional-DRO	80.2±0.3	70.0±2.5	77.4±0.2	70.8±0.5	82.4±0.3	76.1±0.1	82.4±0.7	69.2±0.3	82.0±0.6	71.1±1.8	76.4±1.3	69.0±0.5	91.5±0.0	79.0±0.0	64.2±0.5	60.0±0.7	88.2±0.0	44.9±0.0	93.6±0.0	81.6±0.2
	Holistic-DRO	75.6±1.7	80.2±1.0	77.5±0.1	71.3±0.2	79.7±0.7	70.8±0.6	82.6±0.2	67.4±0.3	77.1±0.5	77.2±0.1	76.9±0.1	68.3±0.1	90.6±0.3	76.8±0.3	63.8±0.1	58.6±0.2	76.4±5.1	43.8±0.4	91.9±0.5	79.5±0.7
NN-DRO Methods (base: NN)	NN2-CVaR-DRO	80.8±0.1	70.3±3.3	62.0±4.3	66.5±2.5	78.3±7.9	74.2±0.9	75.9±9.8	66.2±4.6	76.2±2.9	73.3±2.3	75.4±1.4	68.5±1.4	91.4±0.2	80.7±0.3	62.9±1.0	59.5±0.9	75.5±17.6	60.3±4.1	93.7±0.4	81.5±1.1
	NN2-χ^2-DRO	79.9±3.2	75.1±5.1	72.5±4.7	71.7±4.2	77.7±9.6	74.2±1.4	77.4±4.2	67.5±1.7	83.2±0.2	67.6±0.7	76.5±1.0	69.1±0.9	91.7±0.2	79.8±0.2	62.7±1.8	59.2±1.2	87.5±1.6	61.8±3.4	93.4±0.5	81.0±1.4
	NN2-CVaR-DORO	76.1±4.7	79.1±5.9	68.4±8.8	69.0±5.5	80.5±2.7	73.6±0.9	71.3±12.0	63.1±5.3	82.2±0.9	67.5±1.7	68.9±7.5	63.5±4.1	91.6±0.2	79.2±0.3	63.6±0.7	60.3±0.6	87.9±0.7	58.9±3.4	93.3±0.3	80.7±0.8
	NN2-χ^2-DORO	73.2±4.2	79.2±5.9	67.3±1.8	72.9±0.9	71.6±11.6	70.6±3.7	62.5±8.6	57.7±3.6	82.9±0.5	69.9±1.6	62.7±4.2	58.1±3.7	91.3±0.2	78.6±0.3	63.9±0.0	60.6±0.0	68.8±18.4	58.8±6.7	92.7±0.8	80.1±2.3
	NN3-CVaR-DRO	79.3±5.1	73.7±5.8	62.4±8.8	67.4±6.0	78.5±12.3	73.2±1.6	77.2±6.8	66.1±2.4	81.9±2.9	69.6±3.0	76.8±0.4	69.0±1.0	91.5±0.2	79.7±0.4	63.0±1.8	59.4±1.2	81.7±14.3	59.0±3.2	93.4±0.5	81.3±1.0
	NN3-χ^2-DRO	80.6±1.4	74.9±4.0	74.6±4.1	73.6±5.2	76.9±11.0	73.1±1.2	78.3±3.3	66.4±2.0	83.7±0.3	66.9±0.4	76.5±0.9	69.6±1.0	91.5±0.2	79.5±0.3	64.0±1.5	60.0±1.3	88.3±1.0	60.7±3.1	93.2±0.4	81.3±1.0
	NN4-CVaR-DRO	80.9±0.4	72.4±3.6	61.3±7.1	66.2±5.1	72.5±14.6	71.7±1.6	70.2±11.2	63.7±2.9	83.4±1.1	68.2±1.8	76.1±0.8	68.2±1.4	91.4±0.2	79.9±0.2	62.9±1.5	59.3±1.4	78.6±17.4	59.4±5.5	93.3±0.3	81.8±0.5
	NN4-χ^2-DRO	78.0±4.1	80.0±6.2	68.7±4.4	66.0±6.0	79.7±7.6	71.1±1.0	68.9±12.3	60.3±5.1	83.9±0.2	66.9±0.5	74.1±3.9	68.6±2.5	90.9±0.4	79.5±0.3	63.5±1.4	60.0±1.1	88.7±0.1	61.4±4.3	93.0±0.4	80.5±1.0
Tree-DRO Methods (base: XGB/LGBM)	XGB-CVaR-DRO	75.8±3.3	71.5±3.6	66.5±2.9	65.9±3.3	82.1±2.5	72.5±1.6	83.7±2.5	68.3±0.9	80.1±2.2	65.9±1.1	72.4±5.6	66.1±2.6	91.7±0.2	80.0±0.3	62.3±1.9	57.8±1.0	85.5±3.9	59.8±1.3	94.5±0.3	83.7±1.1
	XGB-KL-DRO	76.4±2.4	74.6±3.8	69.5±4.6	66.5±4.7	81.8±4.6	73.0±1.9	84.0±0.6	68.8±0.3	82.4±6.1	66.9±3.3	76.5±1.9	68.3±1.1	91.3±0.3	79.6±0.2	62.5±1.4	58.1±0.5	87.6±0.4	59.6±1.0	94.6±0.2	84.0±1.1
	LGBM-CVaR-DRO	75.2±6.0	76.7±5.5	65.1±3.2	66.3±3.7	78.9±5.0	71.6±2.4	84.2±0.6	68.4±0.4	78.2±2.9	66.8±2.7	77.0±1.3	68.5±1.0	91.9±0.1	79.1±0.1	61.9±1.2	57.5±0.5	89.2±0.6	62.8±1.5	94.6±0.2	83.9±0.6
	LGBM-KL-DRO	73.5±6.1	77.5±5.6	70.6±5.0	69.6±5.0	83.0±3.8	71.1±1.3	83.2±2.1	68.2±1.1	84.1±3.1	67.3±1.3	76.8±1.4	68.8±0.8	91.9±0.2	79.3±0.2	62.5±0.8	57.7±0.5	86.2±5.0	60.2±1.7	94.6±0.2	84.4±0.6
Kernel-DRO Methods (base: Kernel)	Kernel-χ2-DRO	81.2±0.5	71.0±0.5	73.1±2.1	67.0±1.9	85.3±0.2	70.9±1.7	84.7±0.2	69.8±0.4	81.3±0.7	69.9±1.9	76.0±1.6	69.6±1.1	91.7±0.1	79.6±0.2	64.8±0.5	60.2±0.3	88.6±0.1	57.6±2.3	93.6±0.2	81.3±0.9
	Kernel-CVaR-DRO	81.1±0.6	71.0±0.6	72.7±5.3	67.4±3.5	86.3±0.3	71.7±0.9	83.8±0.5	69.4±0.5	81.6±1.4	68.6±2.1	77.0±1.3	69.1±0.7	91.6±0.2	80.1±0.2	63.9±2.0	59.6±0.7	88.5±0.4	61.2±1.5	93.6±0.2	81.8±0.7
	Kernel-KL-DRO	81.0±0.5	71.2±0.8	67.3±5.1	64.7±4.0	84.7±0.3	73.0±1.2	84.1±0.3	69.9±0.4	81.2±0.7	69.8±2.3	74.9±3.1	69.1±1.9	91.5±0.1	80.0±0.2	65.0±0.4	60.3±0.3	87.8±2.0	60.5±3.3	93.4±0.3	80.5±1.2
	Kernel-Wasserstein-DRO	78.7±3.7	72.7±2.7	77.4±0.2	71.3±0.8	83.2±0.9	73.9±1.2	84.5±0.3	67.0±0.4	79.8±1.2	74.2±2.1	77.1±0.4	68.5±0.7	91.5±0.1	79.3±0.2	63.8±0.9	59.1±0.4	88.6±0.2	58.4±3.7	93.3±0.3	81.3±1.3
Imbalanced Learning & Fairness Methods (base: XGB)	SUBY	78.2±3.2	64.9±1.4	72.0±1.0	67.8±2.7	80.8±2.2	73.4±1.0	73.2±2.3	67.2±1.6	85.5±0.3	64.6±0.6	72.0±1.7	67.6±1.7	91.8±0.1	80.6±0.1	62.3±2.1	60.3±1.7	87.5±0.4	60.1±1.5	88.2±2.0	72.3±2.9
	RWY	80.2±1.2	66.7±0.5	71.5±2.6	66.4±2.5	80.2±4.8	73.5±1.9	81.1±0.7	70.4±0.3	84.3±1.8	65.4±0.9	73.7±0.7	68.7±1.5	91.8±0.1	80.6±0.1	62.2±1.7	59.3±1.6	86.9±0.4	58.0±1.8	94.0±0.2	82.5±1.1
	SUBG	77.4±6.0	74.4±7.0	71.9±3.6	66.3±2.5	-	-	83.9±1.0	69.5±0.5	-	-	75.7±2.8	68.2±3.3	91.8±0.1	80.6±0.1	54.7±2.8	53.5±1.8	-	-	94.0±0.2	82.9±0.9
	RWG	80.5±0.9	70.8±0.6	72.9±2.4	67.7±0.9	-	-	82.5±2.1	68.7±1.3	-	-	78.2±1.4	70.4±1.2	91.8±0.1	80.6±0.1	62.5±1.6	59.2±1.3	-	-	94.5±0.3	83.2±0.6
	In-processing	77.9±3.6	71.2±1.5	70.1±0.7	65.0±1.5	-	-	83.7±0.9	69.5±0.7	-	-	78.9±1.6	71.3±1.1	92.3±0.0	80.4±0.1	64.7±1.8	61.3±1.6	-	-	94.4±0.3	84.0±0.6
	Post-processing	74.5±4.4	76.3±6.5	74.0±2.3	68.4±1.9	-	-	82.5±1.6	68.3±1.3	-	-	79.3±0.3	71.4±0.2	91.8±0.0	80.0±0.1	62.1±1.4	58.9±1.0	-	-	87.7±0.1	83.7±0.7

Table 12. Results (Macro F1-Score) of 10 selected pairs in Section 3.2, where we run each method with its top-10 configurations (according to the target F1-score) and report its mean accuracy and standard deviation.

Dataset		ACS Income		ACS Mobility		US Taxi		ACS Pub.Cov		US Accident		ACS Time		Sub-Sampling		diabetes		assistments		college	
Shift Pattern		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more		X more		X dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more	
Source → Target Pair		CA→PR		MS→HI		NYC→BOG		NE→LA		CA→OR		2010→2017		Young→Old		White→Others		700→1		Normal → Special	
		i.d.	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*
Basic Methods	LR	79.7±0.6	61.3±0.4	53.7±1.0	53.8±0.3	80.9±0.3	74.8±0.2	65.0±0.2	60.2±0.9	73.1±0.2	59.4±1.2	46.8±0.0	43.0±0.0	79.6±0.1	77.7±0.0	61.0±0.1	58.0±0.2	83.6±0.0	41.1±0.6	84.3±0.0	76.2±0.2
	SVM	80.0±0.0	60.7±0.0	59.7±0.7	52.9±0.5	79.6±0.1	75.6±0.1	64.7±0.4	59.6±0.3	73.4±0.1	60.9±0.1	65.2±0.2	59.8±0.2	78.6±0.1	77.4±0.2	60.7±0.0	57.8±0.0	83.5±0.0	35.8±0.0	84.0±0.0	76.2±0.2
	Kernel-SVM	69.8±7.0	66.7±2.6	53.7±3.1	51.7±1.3	63.6±10.3	69.2±1.9	54.3±2.3	57.9±1.4	76.5±2.3	61.0±1.2	53.2±2.1	54.4±1.8	78.2±1.1	75.4±0.7	60.8±1.1	58.9±0.8	66.0±6.7	60.0±1.6	74.7±3.3	69.4±3.2
	NN2	79.8±0.3	61.4±1.5	58.7±4.7	54.4±2.8	75.5±11.8	72.1±1.7	63.6±3.3	62.3±3.6	80.5±0.7	62.1±0.3	66.2±2.1	62.7±3.2	79.0±0.6	77.6±0.3	60.2±0.7	58.5±0.7	83.9±0.9	57.8±4.0	84.0±0.3	76.3±0.8
	NN3	79.4±0.6	60.5±2.7	51.6±2.5	48.3±0.8	82.1±0.7	72.9±1.0	62.3±3.2	59.4±2.9	80.4±1.5	62.1±0.6	66.0±2.0	61.3±2.5	78.6±0.6	77.3±0.5	60.5±1.0	58.6±0.4	84.2±0.3	58.9±3.3	84.0±0.5	77.6±0.8
	NN4	79.2±2.5	60.5±1.2	50.9±1.6	48.4±0.5	81.0±1.3	72.7±2.2	56.7±4.3	54.2±4.5	80.0±1.9	62.2±0.5	65.1±1.7	61.4±1.9	77.4±0.6	76.2±0.7	60.6±1.5	58.6±1.5	83.2±2.4	55.7±3.5	84.3±0.6	77.3±0.7
Tree-based Ensemble Methods	RF	73.3±0.2	62.0±0.6	63.3±1.5	51.0±0.8	83.0±0.1	73.1±0.2	70.7±0.3	61.9±0.4	83.0±0.5	59.8±0.3	69.3±0.7	64.6±1.0	77.1±0.5	78.1±0.2	61.3±0.7	60.2±0.7	85.1±0.3	63.9±1.5	86.0±0.2	79.7±0.5
	XGB	77.9±1.0	57.9±1.0	61.8±1.0	52.5±0.5	81.9±3.0	71.3±0.8	71.9±1.6	65.6±0.5	78.4±0.4	60.4±0.4	69.3±1.1	64.7±0.9	79.6±0.2	78.4±0.1	60.2±0.8	57.7±0.5	86.0±0.4	60.4±0.6	87.7±0.1	81.5±0.4
	GBM	79.1±2.3	61.3±1.7	56.9±5.9	52.5±1.5	82.4±1.2	72.2±0.7	72.0±1.1	66.0±0.8	81.4±4.1	59.8±1.1	69.4±1.3	65.3±1.3	79.9±0.2	79.1±0.1	61.1±0.0	60.5±0.0	85.8±0.4	61.8±1.5	87.1±0.3	80.8±0.6
	LGBM	78.1±1.3	62.4±2.7	63.7±1.0	53.2±0.4	79.5±2.3	72.6±3.0	71.6±0.7	65.4±0.2	76.2±3.7	59.7±1.2	69.3±1.0	65.1±0.7	79.5±0.2	77.8±0.1	60.4±1.5	58.1±1.1	84.9±0.4	62.2±2.1	87.7±0.1	80.6±0.8
Linear-DRO Methods (base: SVM)	KL-DRO	79.3±0.1	61.2±0.1	61.0±0.1	53.2±0.1	79.9±0.7	74.2±0.4	65.3±0.7	60.8±0.9	74.2±0.0	61.8±0.0	65.2±0.2	60.2±0.4	77.6±0.3	77.4±0.1	60.2±0.7	57.3±0.7	83.6±0.0	38.4±0.0	84.1±0.1	76.3±0.2
	CVaR-DRO	77.8±2.7	60.4±1.1	57.5±0.0	52.4±0.0	80.4±2.1	75.1±1.9	66.5±2.4	63.9±1.2	74.5±0.2	61.8±0.2	66.1±0.7	61.5±1.0	79.4±0.0	78.6±0.1	59.0±0.5	56.1±0.6	83.6±0.2	42.7±1.6	83.8±0.6	75.7±0.9
	χ^2-DRO	79.8±0.1	61.1±0.3	58.0±1.1	52.6±0.3	80.5±0.2	76.0±0.1	66.8±0.5	63.8±0.2	74.1±0.1	61.7±0.1	65.3±0.1	60.2±0.1	78.5±0.1	77.0±0.0	60.3±0.6	57.1±0.8	83.5±0.0	39.9±0.3	84.2±0.0	76.5±0.1
	TV-DRO	77.3±2.4	60.2±0.7	58.0±1.2	52.6±0.5	75.9±4.6	72.8±2.9	62.9±1.6	57.8±1.8	74.1±0.6	61.5±0.6	66.3±0.7	62.2±1.3	77.1±0.2	77.4±0.1	58.1±0.9	55.4±1.0	83.3±0.2	34.2±3.4	84.1±0.1	76.2±0.1
	Wasserstein-DRO	74.8±5.1	60.6±0.8	57.1±0.1	51.9±0.0	80.6±0.0	75.3±0.1	63.5±1.3	55.5±1.7	73.0±0.9	59.6±1.7	62.9±0.0	58.8±0.0	77.5±0.1	75.3±0.0	56.9±0.9	55.6±0.4	83.4±0.0	33.1±0.0	84.0±0.2	76.3±0.4
	Aug. Wass.-DRO	77.3±4.8	60.9±0.5	55.2±4.0	51.7±0.4	76.5±1.8	71.2±1.9	65.0±3.2	62.7±2.8	75.4±0.5	62.6±0.5	62.0±3.3	58.6±1.7	78.5±0.3	78.1±0.1	57.4±1.5	55.2±1.6	61.4±8.1	53.7±1.2	83.9±0.2	76.4±0.4
	Satis. Wass.-DRO	72.2±3.2	59.9±0.4	43.1±0.0	44.0±0.0	71.8±2.6	68.5±2.9	44.5±0.0	37.3±0.0	70.0±0.0	56.3±0.0	41.7±0.0	37.8±0.0	75.3±1.2	74.7±1.7	36.9±0.0	33.6±0.0	82.6±0.0	32.9±0.0	81.7±0.9	75.9±0.4
	Sinkhorn-DRO	69.6±2.9	68.9±0.8	55.9±0.8	53.1±0.6	69.2±7.0	74.9±1.3	66.9±2.2	63.5±2.4	74.2±0.6	58.5±0.6	53.4±8.6	56.2±5.5	79.3±0.2	78.1±0.1	53.5±1.6	53.5±1.7	82.4±1.2	33.9±2.9	81.6±2.2	75.2±1.6
	Unified-DRO(L_2)	79.8±0.0	60.5±0.0	57.1±0.1	51.9±0.0	79.9±0.0	75.1±0.1	63.1±0.1	55.6±0.0	71.8±2.1	58.3±2.3	62.9±0.0	58.8±0.0	78.7±0.0	77.5±0.0	52.1±0.0	52.5±0.0	83.4±0.0	33.1±0.0	84.0±0.0	76.4±0.0
	Unified-DRO(L_{inf})	79.1±1.0	60.7±0.1	57.1±0.1	51.9±0.0	80.1±0.0	75.4±0.1	64.4±0.0	56.3±0.0	71.3±1.7	57.3±1.0	62.9±0.0	58.8±0.0	78.9±0.0	77.3±0.0	55.2±0.0	54.8±0.0	83.4±0.0	33.1±0.0	84.1±0.0	76.3±0.1
	Marginal-DRO	79.9±0.1	61.0±0.1	59.2±1.7	53.0±0.6	79.9±0.1	76.0±0.1	67.9±0.6	64.0±1.0	74.4±0.2	61.9±0.2	65.7±0.0	60.6±0.0	78.1±0.2	76.7±0.3	61.8±0.1	59.4±0.5	83.9±0.0	53.8±0.0	84.3±0.0	76.4±0.2
	Conditional-DRO	79.6±0.3	58.6±1.8	59.3±1.3	53.2±0.5	80.0±0.5	76.1±0.1	67.6±1.2	65.3±2.0	79.0±0.9	62.4±0.9	66.5±0.5	62.2±1.3	79.8±0.1	77.4±0.0	60.9±1.0	58.6±1.3	83.4±0.0	33.4±0.0	84.1±0.0	76.3±0.2
	Holistic-DRO	73.2±2.3	65.3±0.3	58.1±2.0	52.0±0.6	73.5±1.3	70.2±0.7	62.4±0.6	58.6±0.9	70.4±1.1	56.5±0.7	64.2±0.2	59.1±0.2	74.8±0.3	73.8±0.1	59.1±0.2	56.1±0.3	83.4±0.0	33.2±0.1	79.2±1.5	72.7±1.0
NN-DRO Methods (base: NN)	NN2-CVaR-DRO	80.1±0.1	58.9±2.2	53.3±1.3	51.1±0.7	75.1±8.0	74.1±0.9	61.3±5.5	61.5±4.1	66.3±7.9	57.3±4.8	65.9±1.3	61.9±2.1	79.2±0.4	78.5±0.2	60.5±0.7	58.9±0.9	72.8±15.9	59.6±4.5	85.0±0.7	78.0±1.0
	NN2-χ^2-DRO	78.3±5.3	61.6±2.4	55.9±5.2	51.9±1.7	75.1±8.6	74.1±1.5	64.2±1.9	64.0±2.1	81.0±0.2	62.5±0.3	66.3±1.5	61.8±2.3	79.7±0.4	78.2±0.2	60.5±1.4	58.5±1.2	83.3±1.6	61.2±3.6	84.5±0.7	77.5±1.2
	NN2-CVaR-DORO	73.0±8.1	63.3±2.4	53.0±4.4	52.0±1.4	77.6±1.9	73.4±1.0	57.6±7.7	59.2±4.3	79.6±1.3	60.9±0.4	60.5±4.0	57.9±2.4	79.2±0.6	76.9±0.4	61.4±0.5	59.6±0.6	83.4±1.0	57.0±4.0	84.3±0.4	76.9±0.9
	NN2-χ^2-DORO	69.3±6.9	62.2±3.0	51.0±1.7	51.3±0.6	68.6±10.9	70.2±4.0	51.6±3.1	54.7±3.7	80.3±0.8	62.1±0.3	56.6±4.0	55.1±3.8	79.0±0.4	77.1±0.3	59.9±0.0	59.3±0.0	65.6±17.3	58.2±6.4	82.1±2.0	76.1±2.1
	NN3-CVaR-DRO	77.1±8.9	59.7±2.0	53.5±3.6	51.6±1.3	74.9±12.1	73.0±1.6	61.0±2.7	60.6±2.8	77.6±6.4	60.6±3.3	65.3±2.2	60.6±2.8	79.5±0.3	78.1±0.3	60.4±1.5	58.7±1.2	78.2±12.7	57.6±3.6	84.4±0.8	77.5±0.8
	NN3-χ^2-DRO	79.4±2.2	61.9±2.7	56.2±3.4	51.2±1.3	74.5±10.0	73.0±1.2	63.2±2.1	61.5±2.3	81.4±0.3	62.1±0.2	67.2±1.7	63.4±2.7	78.9±0.4	77.7±0.4	61.9±1.3	59.4±1.4	84.1±0.9	59.7±3.7	84.2±0.6	77.5±1.2
	NN4-CVaR-DRO	80.0±0.7	60.3±2.6	52.5±2.5	51.0±0.7	70.4±13.9	71.6±1.6	57.1±5.2	59.3±3.8	80.6±1.8	62.0±0.5	64.7±2.6	60.0±3.7	78.8±0.4	78.3±0.1	60.5±1.5	58.7±1.3	75.4±15.6	57.4±6.3	84.8±0.3	78.0±0.6
	NN4-χ^2-DRO	75.2±6.8	64.6±3.2	54.4±2.4	52.3±0.7	77.1±6.7	70.8±1.2	54.0±4.8	55.5±3.9	81.6±0.3	62.1±0.2	66.2±2.6	63.7±1.6	78.9±0.4	77.7±0.3	61.3±1.2	59.4±1.1	84.3±0.3	59.8±5.2	84.0±1.0	76.6±1.0
Tree-DRO Methods (base: XGB/LGBM)	XGB-CVaR-DRO	74.6±3.3	58.5±2.9	54.8±4.3	52.6±1.9	79.5±2.2	72.4±1.7	69.5±2.5	62.8±0.7	76.5±2.7	60.1±0.8	64.7±3.8	61.8±1.5	78.0±0.2	77.4±0.2	58.2±0.9	56.1±0.6	83.0±3.4	59.7±1.3	86.9±0.5	79.9±1.3
	XGB-KL-DRO	74.8±2.7	60.5±2.5	60.4±3.6	53.1±1.6	79.9±3.8	72.8±1.9	70.1±0.7	63.7±0.7	79.2±5.8	59.9±0.9	67.6±0.9	63.0±0.5	78.2±0.3	77.4±0.1	58.4±1.1	56.4±0.7	84.7±0.5	59.6±1.1	87.1±0.5	79.9±1.2
	LGBM-CVaR-DRO	72.9±7.0	61.2±4.4	53.5±4.1	51.8±1.6	76.3±4.5	71.4±2.5	69.8±1.1	62.7±1.7	73.9±3.2	59.2±1.2	66.7±1.6	62.2±1.4	78.4±0.3	76.6±0.2	58.0±0.9	55.9±0.9	86.3±0.6	62.4±1.8	87.1±0.4	79.9±0.9
	LGBM-KL-DRO	70.0±8.9	60.5±3.2	57.3±5.0	53.2±1.8	79.9±3.4	70.8±1.5	70.0±1.3	63.5±0.8	80.9±3.8	60.4±0.4	67.6±1.0	63.1±0.4	78.2±0.4	76.5±0.1	58.1±1.0	55.8±0.8	83.6±4.7	59.7±1.8	87.1±0.3	80.4±0.8
Kernel-DRO Methods (base: Kernel)	Kernel-χ2-DRO	80.5±0.6	59.4±0.3	59.4±0.9	53.0±0.8	82.6±0.2	70.8±1.7	67.9±0.4	63.6±0.9	77.8±1.2	61.5±0.6	67.6±0.7	65.7±0.9	80.1±0.5	78.3±0.2	61.0±0.8	58.5±0.5	84.3±0.3	55.7±2.8	84.3±0.5	77.4±1.0
	Kernel-CVaR-DRO	80.4±0.6	59.4±0.4	57.1±3.3	52.6±1.3	84.3±0.4	71.6±0.9	68.3±0.6	64.0±1.3	78.2±2.2	61.4±0.5	65.7±1.0	61.7±1.5	78.6±0.4	78.3±0.2	60.3±1.6	57.9±1.1	83.9±0.5	59.8±1.5	84.3±0.4	77.6±1.0
	Kernel-KL-DRO	80.3±0.6	59.6±0.5	56.5±1.8	52.9±1.5	81.3±0.4	72.9±1.3	67.8±1.0	64.8±0.8	77.6±1.2	61.2±0.5	66.5±2.2	65.0±1.9	78.7±0.6	78.5±0.2	61.3±0.5	58.6±0.5	83.6±1.9	60.2±3.3	83.9±0.7	76.7±1.2
	Kernel-Wasserstein-DRO	77.6±4.5	60.0±0.9	56.1±2.1	51.0±0.4	79.6±1.6	73.7±1.3	66.2±0.8	56.5±0.8	75.2±2.1	60.1±1.5	63.3±1.7	58.7±2.0	77.8±0.3	77.3±0.3	56.6±1.9	55.6±0.8	84.1±0.3	56.9±3.5	83.3±1.0	76.9±1.4
Imbalanced Learning & Fairness Methods (base: XGB)	SUBY	77.8±3.4	54.8±1.1	60.4±5.0	53.7±1.0	79.5±2.3	73.4±1.0	66.4±2.1	67.0±1.5	84.3±0.3	62.4±0.6	69.0±1.5	67.2±1.6	80.7±0.2	79.2±0.1	62.0±1.9	60.0±1.8	84.8±0.6	60.0±1.5	80.0±2.6	71.3±2.7
	RWY	79.7±1.3	56.2±0.2	61.5±4.3	53.4±1.1	78.6±4.6	73.4±1.9	71.6±0.3	68.8±0.7	82.4±2.4	61.7±0.6	69.7±1.3	67.4±2.2	80.7±0.2	79.2±0.1	61.3±1.8	59.3±1.5	84.2±0.4	57.9±1.7	87.0±0.5	79.7±1.1
	SUBG	74.4±10.6	59.7±1.9	61.0±2.8	52.9±1.0	-	-	70.9±0.9	65.3±0.6	-	-	68.3±2.2	64.0±2.3	80.7±0.2	79.2±0.1	53.4±2.2	53.1±1.8	-	-	85.8±0.4	78.7±1.0
	RWG	79.7±0.9	59.0±0.6	61.8±2.7	52.8±1.0	-	-	70.4±2.0	65.0±1.3	-	-	70.2±0.9	65.3±0.7	80.7±0.2	79.2±0.1	60.5±1.3	58.9±1.2	-	-	87.2±0.5	78.8±0.8
	In-processing	76.7±4.3	58.6±0.8	60.6±0.5	51.8±0.6	-	-	70.8±0.7	65.3±0.5	-	-	67.7±3.0	63.5±2.7	80.1±0.1	78.7±0.1	62.2±1.5	61.1±1.6	-	-	86.9±0.4	80.3±0.7
	Post-processing	71.6±6.7	60.9±3.0	62.1±1.6	52.8±0.9	-	-	69.1±1.7	64.1±1.3	-	-	69.1±2.1	64.6±1.2	79.2±0.1	78.7±0.1	60.2±1.2	58.8±1.0	-	-	82.5±0.2	80.0±0.8

Analysis. Based on results in Tables 11 and 12, we provide a more detailed analysis compared to our main body.

- Different methods do not exhibit consistent rankings over different distribution shift patterns. The results show that the algorithmic rankings across different settings are quite different. And even the rankings of algorithms within the same method class vary a lot. This further demonstrates the complexity of $Y|X$-shifts.
- Tree-based ensemble methods show competitive performances but do not significantly eliminate the generalization error between source and target data, characterized by the difference between *o.o.d.* and *i.d.* method performance. However, the performance degradation between source and target is still large.
- Imbalance methods and fairness methods show similar performance with the base model class (XGBoost). We find these methods do not have a significant improvement over the basic model class (XGBoost). This is because they are not designed specifically for tree-based ensemble methods.

Additionally, we report the *oracle* accuracy (Macro F1-Score) of each method across all target domains in each setting in Table 14 and Table 15, where we select the best configuration for each method according to the *target* performance. The observed phenomena are consistent with our previous findings.

Table 13. Performance of methods across all 172 source-target pairs. The table presents the algorithmic ranking and accuracy for each method over all pairs. The min, max, mean, standard deviation, and median of these metrics are computed across all pairs.

Category	Method	Rank					Accuracy		
		Min	Max	Mean	Std	Median	Mean	Std	Median
Basic Methods	LR	5	44	28.58	8.20	30	74.86	5.99	75.78
	SVM	7	45	29.68	8.58	32	74.86	5.71	75.56
	Kernel-SVM	2	44	31.13	12.64	36	75.04	5.94	75.78
	NN2	1	43	18.63	8.88	17	76.56	5.09	77.19
	NN3	1	43	23.22	12.58	26	76.19	5.61	77.03
	NN4	3	43	27.23	12.50	34	75.84	5.77	76.54
Tree-based Ensemble Methods	RF	1	39	14.56	8.59	15	76.51	5.21	76.74
	XGB	1	44	15.05	11.40	12	76.70	5.09	76.56
	GBM	1	35	9.53	7.89	6	76.95	4.93	77.15
	LGBM	1	29	8.80	6.55	8	77.23	4.97	77.45
Linear-DRO Methods (base: SVM)	KL-DRO	3	45	29.47	7.66	30	74.98	5.48	75.55
	CVaR-DRO	1	40	23.44	8.65	23	75.39	5.26	75.89
	χ^2-DRO	5	45	26.11	8.35	27	75.18	5.40	75.82
	TV-DRO	4	43	31.14	8.07	32	74.86	5.51	75.38
	Wasserstein-DRO	11	43	33.63	6.07	35	74.66	5.95	75.70
	Aug. Wass.-DRO	2	40	25.45	9.02	25	75.98	5.06	76.30
	Satis. Wass.-DRO	13	45	41.88	5.16	43	72.45	7.05	73.77
	Sinkhorn-DRO	2	44	25.60	12.29	29	75.49	5.37	76.46
	Unified-DRO(L_2)	14	45	35.91	5.86	37	74.44	5.91	75.33
	Unified-DRO(L_{inf})	4	45	34.95	7.06	36	74.46	5.89	75.52
	Marginal-DRO	2	45	24.03	9.58	24	75.46	5.08	76.15
	Conditional-DRO	1	45	20.77	8.82	20	76.10	5.36	76.32
	Holistic-DRO	2	45	31.49	10.17	34	74.52	5.95	75.64
NN-DRO Methods (base: NN)	NN2-CVaR-DRO	1	38	16.51	11.29	13	76.89	5.29	77.28
	NN2-χ^2-DRO	1	36	15.17	9.87	12	76.87	5.25	77.38
	NN2-CVaR-DORO	2	40	21.84	8.79	21	76.34	5.01	76.95
	NN2-χ^2-DORO	2	45	31.12	11.74	34	74.86	5.66	75.41
	NN3-CVaR-DRO	1	41	18.76	11.84	14	76.74	5.70	77.14
	NN3-χ^2-DRO	1	40	18.09	12.47	14	76.69	5.67	77.22
	NN4-CVaR-DRO	1	42	23.99	12.85	22	76.12	5.45	76.25
	NN4-χ^2-DRO	1	45	21.92	14.94	20	76.01	5.78	76.61
Tree-DRO Methods (base: XGB/LGBM)	XGB-CVaR-DRO	1	41	16.48	10.17	15	76.81	5.21	77.31
	XGB-KL-DRO	1	44	17.24	11.83	16	76.98	5.50	77.79
	LGBM-CVaR-DRO	1	45	17.95	13.59	14	76.63	5.33	76.55
	LGBM-KL-DRO	1	43	16.99	11.88	15	76.91	5.33	77.48
Kernel-DRO Methods (base: Kernel)	Kernel-χ^2-DRO	1	36	16.40	8.16	18	76.17	4.97	76.39
	Kernel-CVaR-DRO	1	42	18.08	8.29	19	76.12	4.96	76.41
	Kernel-KL-DRO	1	35	17.15	8.61	17	76.15	4.89	76.32
	Kernel-Wasserstein-DRO	1	44	24.91	8.62	26	75.59	5.65	76.07
Imbalanced Learning & Fairness Methods (base: XGB)	SUBY	1	45	33.69	14.43	42	75.40	5.27	74.50
	RWY	1	45	31.42	16.44	40	75.72	5.43	75.02
	SUBG	1	44	14.49	10.93	10	76.92	5.55	76.70
	RWG	1	45	14.69	11.62	10	76.90	5.37	76.63
	In-processing	1	42	13.19	11.57	8	76.48	4.65	76.54
	Post-processing	1	43	19.19	14.01	13	76.14	4.66	76.08

Table 14. Results (Accuracy) of 10 settings in Section 3.2 with all 172 target domains, where we report the source accuracy, the mean target accuracy and standard deviation across multiple target domains in each setting. We run each method with its best configuration (according to the target F1-score), and boldface the best target performance within each class of methods in each setting.

Dataset		ACS Income		ACS Mobility		US Taxi		ACS Pub.Cov		US Accident		ACS Time		Sub-Sampling		diabetes		assistments		college	
Shift Pattern		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more		X more		X dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more	
Source → Target Pair		CA→ 50 Domains		MS→50 Domains		NYC→3 Domains		NE→50 Domains		CA→13 Domains		2010→3 Domains		Young→Old		White→Others		700→1		Normal → Special	
		i.d.	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*
Basic Methods	LR	80.6	77.3±1.8	75.8	72.7±3.7	84.4	74.4±1.5	84.0	**75.7**±6.6	78.7	71.2±9.5	72.1	62.0±2.6	91.9	79.5±0.0	65.1	60.0±0.0	88.3	49.7±0.0	93.4	81.9±0.0
	SVM	80.7	77.1±1.9	77.8	72.9±3.5	83.3	**74.6**±1.5	83.5	75.4±6.3	78.8	71.3±9.4	77.2	69.1±2.1	91.7	**80.1**±0.0	64.9	59.9±0.0	88.3	46.0±0.0	93.4	81.9±0.0
	Kernel-SVM	80.1	77.6±1.6	77.0	72.8±3.6	79.9	71.3±1.3	78.2	70.7±5.8	78.4	82.8±6.8	72.5	63.5±2.4	91.9	79.0±0.0	64.0	61.1±0.0	66.1	64.7±0.0	88.6	78.3±0.0
	NN2	81.0	78.6±1.7	77.6	**72.9**±3.4	81.5	73.6±0.9	82.9	75.3±6.2	82.3	82.6±6.9	76.8	**70.8**±1.9	91.2	79.5±0.0	64.2	60.4±0.0	88.9	**65.6**±0.0	93.7	82.9±0.0
	NN3	80.0	**79.1**±1.6	76.6	72.8±3.4	85.0	73.1±0.9	82.5	74.0±6.3	82.9	82.7±6.9	77.6	70.2±2.3	91.5	80.0±0.0	64.5	**60.9**±0.0	88.8	65.0±0.0	93.7	**83.1**±0.0
	NN4	81.0	78.5±1.6	75.8	72.7±3.7	83.8	74.3±1.3	82.1	74.0±6.0	84.4	**84.6**±7.2	75.9	69.9±2.1	90.7	78.8±0.0	63.3	**60.9**±0.0	88.6	60.7±0.0	93.7	82.7±0.0
Tree-based Ensemble Methods	RF	81.3	77.8±1.9	80.0	73.6±3.4	83.6	**75.0**±1.5	85.5	76.0±6.5	86.0	78.3±7.6	79.4	71.2±2.3	92.1	**80.8**±0.0	65.6	62.5±0.0	88.1	**66.6**±0.0	94.3	84.8±0.0
	XGB	81.8	77.6±2.2	79.7	73.5±2.9	83.1	73.9±1.1	85.7	76.5±6.3	87.0	**80.2**±7.8	79.5	71.8±2.4	92.3	80.3±0.0	66.3	**62.6**±0.0	89.8	63.1±0.0	94.8	**85.6**±0.0
	GBM	81.0	**78.2**±1.6	80.0	73.6±3.0	66.1	74.2±1.0	85.6	**76.7**±6.0	85.7	79.0±7.3	79.7	**71.9**±2.3	92.1	**80.8**±0.0	63.7	60.9±0.0	72.7	65.4±0.0	94.7	**85.6**±0.0
	LGBM	80.4	78.1±1.3	80.0	**73.8**±3.3	79.3	74.5±1.5	85.9	76.5±6.4	87.0	79.8±7.8	79.6	71.8±2.4	92.3	80.3±0.0	66.1	**62.6**±0.0	87.5	65.9±0.0	94.9	85.8±0.0
Linear-DRO Methods (base: SVM)	KL-DRO	80.7	77.0±2.2	77.8	72.7±3.5	83.4	74.4±1.6	83.6	75.4±5.9	79.2	72.2±8.8	77.0	68.8±2.3	91.6	79.9±0.0	64.8	59.9±0.0	88.2	47.4±0.0	93.6	81.9±0.0
	CVaR-DRO	80.8	77.4±1.9	78.1	72.8±3.4	83.6	75.2±1.4	84.2	75.8±5.9	79.5	72.4±8.8	76.7	69.5±2.1	91.7	**80.4**±0.0	64.9	60.5±0.0	88.3	**50.7**±0.0	93.5	81.8±0.0
	χ^2-DRO	80.7	77.4±1.9	77.9	72.8±3.5	83.3	75.2±1.4	83.7	75.6±5.6	79.2	72.2±8.8	76.5	69.5±2.1	91.7	79.8±0.0	65.1	60.2±0.0	88.3	48.4±0.0	93.6	81.9±0.0
	TV-DRO	80.6	76.8±2.0	77.5	72.6±3.6	82.9	74.0±1.6	82.4	74.8±6.6	79.2	72.3±8.9	76.6	69.4±2.0	91.0	80.1±0.0	64.0	59.3±0.0	88.3	44.9±0.0	93.6	81.7±0.0
	Wasserstein-DRO	80.5	76.9±1.9	75.8	72.7±3.7	84.1	74.5±1.5	82.3	74.8±6.7	79.2	72.3±8.8	76.0	67.7±2.4	91.5	77.3±0.0	64.0	59.3±0.0	88.2	44.9±0.0	93.5	81.9±0.0
	Aug. Wass.-DRO	80.2	77.0±1.6	77.6	73.0±3.6	82.0	73.9±1.4	83.5	75.3±6.8	69.8	79.6±8.5	76.8	68.8±2.3	91.4	**80.4**±0.0	62.9	59.8±0.0	66.4	58.7±0.0	93.5	81.8±0.0
	Satis. Wass.-DRO	79.4	76.3±1.8	75.8	72.7±3.7	79.7	70.1±1.2	80.0	69.9±8.1	77.1	68.9±10.0	71.7	61.0±2.8	91.3	78.7±0.0	58.5	50.6±0.0	87.8	44.8±0.0	93.0	81.5±0.0
	Sinkhorn-DRO	77.5	**78.1**±1.4	75.7	72.7±3.7	82.5	75.2±1.3	83.2	**75.8**±5.0	80.1	72.2±8.6	76.1	**70.3**±1.6	91.5	79.8±0.0	63.6	59.3±0.0	85.5	50.0±0.0	91.8	**82.4**±0.0
	Unified-DRO(L_2)	80.6	76.9±1.9	77.5	72.6±3.6	83.3	74.3±1.5	82.1	74.7±6.6	79.0	71.9±9.0	76.0	67.7±2.4	91.3	79.7±0.0	62.6	57.8±0.0	88.3	44.9±0.0	93.6	81.6±0.0
	Unified-DRO(L_{inf})	80.6	76.9±1.9	76.5	72.9±3.6	83.6	74.6±1.5	82.2	74.8±6.6	78.2	70.8±9.2	76.0	67.7±2.4	92.0	79.3±0.0	64.0	59.3±0.0	88.2	44.9±0.0	93.6	81.7±0.0
	Marginal-DRO	80.8	77.4±1.9	78.2	72.8±3.4	82.7	75.2±1.4	83.5	76.3±5.7	79.5	72.8±8.7	76.2	69.8±1.9	91.1	79.3±0.0	64.7	60.6±0.0	88.3	56.9±0.0	93.6	81.9±0.0
	Conditional-DRO	80.7	77.5±2.3	77.6	**73.0**±3.3	82.6	**75.4**±1.2	83.3	75.7±5.4	76.3	**79.9**±8.1	76.4	69.7±2.1	91.6	79.3±0.0	64.1	**60.8**±0.0	88.2	44.9±0.0	93.5	81.7±0.0
	Holistic-DRO	78.5	76.6±1.4	77.6	72.9±3.6	80.4	70.9±1.4	82.8	74.7±6.5	78.1	69.9±9.8	77.0	68.3±2.5	91.1	77.3±0.0	63.9	58.8±0.0	88.3	45.0±0.0	92.9	80.6±0.0
NN-DRO Methods (base: NN)	NN2-CVaR-DRO	79.2	78.8±1.4	77.3	**73.6**±3.3	83.9	74.4±1.4	80.9	74.3±5.1	80.8	85.1±7.3	77.9	69.8±2.4	91.3	**80.9**±0.0	64.1	61.2±0.0	88.5	67.7±0.0	94.2	83.3±0.0
	NN2-χ^2-DRO	80.5	78.5±1.5	77.6	73.5±3.3	84.1	**74.8**±1.2	81.9	75.2±5.4	82.0	83.8±7.2	76.7	70.2±2.3	92.1	80.2±0.0	64.4	61.0±0.0	88.4	65.4±0.0	93.3	83.4±0.0
	NN2-CVaR-DORO	78.6	78.2±1.4	75.9	72.7±3.7	81.4	74.7±0.9	83.0	**75.8**±5.6	81.1	78.9±7.9	76.9	68.1±2.4	91.5	79.6±0.0	64.8	**61.5**±0.0	88.1	66.1±0.0	93.4	83.3±0.0
	NN2-χ^2-DORO	78.9	78.4±1.3	75.8	72.7±3.7	82.4	73.8±1.0	78.7	71.1±5.9	76.1	78.9±6.7	67.0	62.2±1.8	91.3	79.2±0.0	63.9	60.6±0.0	88.6	68.3±0.0	93.2	83.0±0.0
	NN3-CVaR-DRO	79.8	78.5±1.5	77.4	73.4±3.4	84.4	74.5±1.4	82.1	74.3±6.5	72.2	**86.4**±9.0	76.1	69.8±2.3	91.5	80.4±0.0	64.9	60.8±0.0	88.6	63.8±0.0	94.0	83.1±0.0
	NN3-χ^2-DRO	80.3	78.6±1.6	76.1	73.4±3.5	83.2	73.9±0.7	80.5	74.2±5.2	75.4	85.4±8.3	77.7	**70.5**±2.2	91.7	80.1±0.0	65.3	**61.5**±0.0	88.7	64.1±0.0	93.8	**83.5**±0.0
	NN4-CVaR-DRO	80.8	78.3±1.5	75.8	73.4±3.6	84.5	73.6±1.3	81.6	73.0±6.5	82.2	84.1±7.1	76.8	70.0±2.4	91.5	80.2±0.0	65.0	61.1±0.0	88.7	**68.6**±0.0	93.8	83.1±0.0
	NN4-χ^2-DRO	80.4	**78.9**±1.3	75.8	73.4±3.6	84.0	71.8±0.9	81.5	72.8±6.4	84.4	83.6±7.0	76.9	69.9±2.0	91.1	80.1±0.0	64.6	61.4±0.0	88.8	67.5±0.0	93.3	83.1±0.0
Tree-DRO Methods (base: XGB/LGBM)	XGB-CVaR-DRO	81.1	79.0±1.4	75.8	**73.4**±3.6	82.7	**75.2**±1.0	85.3	**75.8**±6.5	81.4	77.6±6.4	79.1	**70.4**±2.6	92.1	**80.6**±0.0	64.7	59.1±0.0	89.6	62.8±0.0	94.7	85.2±0.0
	XGB-KL-DRO	80.6	79.1±1.3	75.8	**73.4**±3.6	81.3	74.7±0.8	85.1	75.7±6.9	67.6	**80.0**±7.7	79.2	70.2±2.5	92.0	80.0±0.0	62.6	**60.4**±0.0	80.1	64.1±0.0	94.8	**85.9**±0.0
	LGBM-CVaR-DRO	80.7	**79.2**±1.3	78.1	73.0±2.7	75.8	73.7±1.1	85.1	75.6±6.6	85.6	77.2±8.1	78.9	70.3±2.5	92.0	79.4±0.0	64.7	58.7±0.0	89.7	**65.0**±0.0	94.7	85.5±0.0
	LGBM-KL-DRO	80.1	79.2±1.4	75.8	**73.4**±3.6	74.8	73.1±1.0	85.0	75.5±6.7	68.3	78.9±8.6	79.2	70.2±2.5	92.0	79.6±0.0	64.4	58.6±0.0	89.3	64.8±0.0	94.7	85.5±0.0
Kernel-DRO Methods (base: Kernel)	Kernel-χ2-DRO	81.3	77.5±2.1	76.4	73.0±3.6	85.4	74.2±1.6	85.2	76.4±6.7	82.3	75.8±7.8	78.4	**71.3**±1.9	91.7	80.0±0.0	65.6	**61.0**±0.0	88.6	60.3±0.0	93.9	83.3±0.0
	Kernel-CVaR-DRO	81.2	**77.7**±2.0	78.1	**73.0**±3.3	86.3	73.0±1.1	84.3	76.1±6.3	82.8	**76.4**±7.6	78.2	70.5±2.4	91.9	80.4±0.0	64.9	60.8±0.0	88.1	63.9±0.0	93.5	**83.7**±0.0
	Kernel-KL-DRO	81.5	77.5±2.2	76.5	73.0±3.6	84.9	74.0±1.3	84.9	**76.4**±6.3	82.0	75.3±7.7	78.2	70.8±2.2	91.7	**80.5**±0.0	65.3	61.0±0.0	88.4	**64.9**±0.0	93.9	83.1±0.0
	Kernel-Wasserstein-DRO	81.1	77.5±2.0	77.2	72.8±3.6	84.1	**74.4**±1.6	84.5	75.5±7.1	82.4	75.3±7.8	77.7	69.2±2.6	91.5	79.7±0.0	65.3	59.9±0.0	88.6	61.8±0.0	93.7	83.5±0.0
Imbalanced Learning & Fairness Methods (base: XGB)	SUBY	80.7	75.3±2.9	75.8	72.7±3.7	83.7	73.0±1.5	77.4	72.5±3.0	85.4	**86.5**±7.6	73.8	70.8±1.3	92.0	**80.7**±0.0	65.1	**62.8**±0.0	87.2	**63.6**±0.0	88.9	75.9±0.0
	RWY	81.0	75.4±3.0	75.0	72.2±3.7	76.2	**75.1**±0.9	81.3	74.6±4.0	85.3	86.4±7.4	74.9	70.6±1.4	92.0	**80.7**±0.0	65.1	**62.8**±0.0	87.8	63.1±0.0	94.2	84.1±0.0
	SUBG	81.5	**77.7**±2.0	78.9	73.5±3.1	-	-	85.5	**76.5**±6.0	-	-	79.4	71.6±2.3	92.0	**80.7**±0.0	58.4	56.3±0.0	-	-	93.9	85.0±0.0
	RWG	81.7	77.6±2.1	80.2	73.6±3.1	-	-	85.9	76.4±6.5	-	-	79.7	71.6±2.5	92.0	**80.7**±0.0	65.7	62.2±0.0	-	-	94.7	84.7±0.0
	In-processing	81.7	77.6±2.1	80.4	**73.8**±2.9	-	-	86.1	76.5±6.5	-	-	79.4	**71.8**±2.3	92.3	80.5±0.0	66.1	62.7±0.0	-	-	94.7	**85.9**±0.0
	Post-processing	81.3	76.8±2.2	79.7	73.6±3.0	-	-	85.6	76.5±6.4	-	-	79.4	71.7±2.3	91.7	80.1±0.0	65.5	**62.8**±0.0	-	-	87.9	85.5±0.0

Table 15. Results (Macro-F1 Score) of 10 settings in Section 3.2 with all 172 target domains, where we report the source macro F1-score, the mean target macro F1-score and standard deviation across multiple target domains in each setting. We run each method with its best configuration (according to the target F1-score), and boldface the best target performance within each class of methods in each setting.

Dataset		ACS Income		ACS Mobility		US Taxi		ACS Pub.Cov		US Accident		ACS Time		Sub-Sampling		diabetes		assistments		college	
Shift Pattern		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more		X more		X dominates		$Y\|X$ dominates		$Y\|X$ more		$Y\|X$ more	
Source → Target Pair		CA→ 50 Domains		MS→50 Domains		NYC→3 Domains		NE→50 Domains		CA→13 Domains		2010→3 Domains		Young→Old		White→Others		700→1		Normal → Special	
		i.d.	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*	*i.d.*	*o.o.d*
Basic Methods	LR	79.8	75.6±3.1	55.8	50.2±2.3	81.1	73.0±2.1	66.4	62.7±4.7	73.3	65.4±8.6	46.8	43.0±0.9	79.5	77.7±0.0	61.1	58.1±0.0	83.6	42.4±0.0	84.1	77.0±0.0
	SVM	80.0	75.4±3.2	60.2	53.6±2.2	79.6	73.3±2.2	65.4	61.6±4.2	73.5	65.8±8.5	65.5	60.3±1.1	78.7	77.8±0.0	60.8	57.8±0.0	83.5	35.8±0.0	83.9	76.6±0.0
	Kernel-SVM	79.5	75.3±2.9	58.7	56.8±2.2	77.2	70.8±1.7	58.7	60.1±1.9	77.2	81.1±6.8	55.3	56.3±0.6	80.1	76.9±0.0	63.2	**61.1**±0.0	77.0	62.8±0.0	72.9	74.3±0.0
	NN2	79.7	76.2±2.9	62.7	**61.0**±1.9	78.6	72.8±1.7	68.4	**67.3**±2.3	81.0	80.6±7.0	70.0	**66.8**±1.1	80.2	78.0±0.0	60.4	59.9±0.0	85.1	**65.3**±0.0	84.6	79.1±0.0
	NN3	78.3	76.1±2.5	53.7	50.0±2.6	82.6	72.4±1.6	67.6	64.7±3.1	81.5	80.9±6.9	68.9	64.4±1.5	79.6	**78.2**±0.0	62.0	60.4±0.0	84.7	64.6±0.0	85.2	79.2±0.0
	NN4	80.0	**76.3**±3.1	53.0	46.4±2.2	81.5	**73.6**±1.8	64.0	61.4±3.8	83.2	**83.0**±7.3	68.2	65.2±1.3	78.7	77.6±0.0	63.1	60.6±0.0	84.4	61.2±0.0	84.7	**80.0**±0.0
Tree-based Ensemble Methods	RF	80.5	76.0±3.2	65.1	56.8±1.8	80.6	**73.9**±2.0	71.4	64.2±4.6	84.0	75.5±7.8	69.9	65.5±1.6	77.6	78.4±0.0	62.3	61.7±0.0	84.7	**66.2**±0.0	86.0	80.3±0.0
	XGB	81.1	76.1±3.5	64.9	59.0±2.0	80.5	72.7±1.6	72.6	66.4±4.1	85.1	**78.1**±7.8	71.2	66.9±1.8	79.7	78.5±0.0	63.4	**61.9**±0.0	85.0	62.8±0.0	87.8	82.2±0.0
	GBM	81.2	76.2±3.4	66.3	59.5±2.1	65.5	73.8±1.3	73.7	**66.6**±4.2	83.8	76.3±7.6	71.0	**67.3**±1.9	80.0	**79.2**±0.0	61.1	60.5±0.0	69.4	64.6±0.0	87.3	81.8±0.0
	LGBM	81.2	**76.2**±3.3	66.1	59.0±2.0	78.2	74.2±1.6	73.2	66.3±4.1	85.1	77.7±7.7	71.4	66.7±1.9	79.9	78.1±0.0	63.5	62.0±0.0	84.4	65.7±0.0	87.4	**82.9**±0.0
Linear-DRO Methods (base: SVM)	KL-DRO	79.9	75.3±3.5	61.1	55.3±2.2	80.2	73.0±2.3	66.5	62.6±4.3	74.2	67.3±8.0	65.4	60.7±1.3	78.2	77.6±0.0	60.8	57.9±0.0	83.6	38.5±0.0	84.2	76.6±0.0
	CVaR-DRO	80.0	**75.8**±3.2	62.6	59.3±2.2	81.4	74.4±1.7	68.8	65.3±3.4	74.8	67.5±8.2	66.9	62.9±1.2	79.4	**78.8**±0.0	62.0	59.5±0.0	83.3	45.8±0.0	84.4	76.9±0.0
	χ^2-DRO	79.9	75.7±3.2	60.5	56.8±2.1	80.7	74.3±2.0	67.6	63.8±4.1	74.2	67.3±8.1	66.8	63.1±1.2	78.7	77.1±0.0	61.5	58.6±0.0	83.6	40.3±0.0	84.3	76.6±0.0
	TV-DRO	79.9	75.1±3.3	61.4	56.8±2.3	79.0	72.6±2.2	65.7	61.7±5.9	74.3	67.3±8.1	67.0	64.4±1.2	77.2	77.5±0.0	59.7	58.0±0.0	82.7	44.3±0.0	84.2	76.4±0.0
	Wasserstein-DRO	79.8	75.2±3.3	57.3	48.4±2.9	80.6	73.2±2.1	64.3	61.2±4.6	74.3	67.3±8.1	62.9	58.8±1.5	77.6	75.4±0.0	57.3	56.0±0.0	83.4	33.1±0.0	83.6	77.1±0.0
	Aug. Wass.-DRO	79.8	75.2±3.3	63.0	59.1±1.7	78.9	73.1±2.0	68.5	**66.9**±2.7	69.4	**77.7**±7.9	65.2	61.7±1.2	78.7	78.3±0.0	61.3	59.4±0.0	64.1	**55.2**±0.0	83.7	76.9±0.0
	Satis. Wass.-DRO	78.6	74.5±3.0	43.1	42.1±1.2	73.1	66.9±2.3	44.5	41.0±2.9	70.0	61.9±8.2	41.7	37.9±1.1	75.9	75.6±0.0	36.9	33.6±0.0	82.6	32.9±0.0	82.1	76.1±0.0
	Sinkhorn-DRO	78.1	75.4±2.5	55.9	55.5±1.5	81.0	74.5±1.8	69.9	67.9±2.5	75.7	66.9±8.3	68.3	**66.6**±0.5	79.4	78.3±0.0	56.9	56.3±0.0	78.9	42.6±0.0	82.8	**78.4**±0.0
	Unified-DRO(L_2)	79.8	75.2±3.3	57.3	48.4±2.9	79.9	72.9±2.2	64.3	61.2±4.6	73.9	66.7±8.2	62.9	58.8±1.5	78.7	77.5±0.0	52.1	52.5±0.0	83.4	33.1±0.0	84.0	76.7±0.0
	Unified-DRO(L_{inf})	79.9	75.2±3.3	57.3	48.4±2.9	80.1	73.4±2.1	64.4	61.2±4.6	72.6	64.9±8.3	62.9	58.8±1.5	78.9	77.3±0.0	57.4	56.0±0.0	83.4	33.1±0.0	84.1	76.6±0.0
	Marginal-DRO	80.1	75.7±3.2	62.4	59.3±2.0	80.0	74.3±1.8	69.2	65.9±3.7	74.7	67.9±8.2	67.5	65.2±1.0	78.0	77.1±0.0	63.0	59.9±0.0	83.9	53.8±0.0	84.2	76.7±0.0
	Conditional-DRO	80.0	75.7±3.3	60.6	**60.1**±1.4	80.5	**74.7**±1.6	69.2	66.2±3.7	75.0	77.3±8.7	67.2	65.4±0.8	79.9	77.4±0.0	62.0	**60.3**±0.0	83.4	33.4±0.0	84.1	76.5±0.0
	Holistic-DRO	77.4	74.2±2.5	59.3	52.6±2.4	74.6	68.1±2.5	63.2	60.1±4.6	72.2	63.5±8.9	64.4	59.4±1.5	75.3	73.9±0.0	59.3	56.5±0.0	83.4	33.4±0.0	82.1	74.2±0.0
NN-DRO Methods (base: NN)	NN2-CVaR-DRO	80.1	76.2±3.0	61.7	**61.0**±1.2	77.9	73.9±1.5	66.8	65.7±2.6	79.9	83.5±7.3	67.8	**65.8**±1.5	79.0	**78.7**±0.0	61.0	60.5±0.0	83.9	67.7±0.0	85.9	**79.8**±0.0
	NN2-χ^2-DRO	80.2	**76.4**±3.1	63.0	60.2±2.0	82.1	**74.1**±1.7	68.0	**66.0**±3.7	80.8	82.1±7.2	68.7	64.9±1.6	80.6	78.6±0.0	62.3	60.5±0.0	84.3	65.0±0.0	85.4	79.7±0.0
	NN2-CVaR-DORO	79.0	76.1±2.8	60.5	56.3±2.3	78.9	74.0±1.4	68.5	65.8±3.9	79.4	76.4±8.2	63.4	61.7±1.1	79.5	77.6±0.0	62.5	61.0±0.0	84.6	65.7±0.0	84.7	78.9±0.0
	NN2-χ^2-DORO	78.9	75.6±2.9	52.7	53.7±1.5	79.7	73.0±1.6	58.0	58.1±2.1	75.3	77.0±6.5	62.5	60.2±1.3	79.2	77.7±0.0	59.9	59.3±0.0	84.6	**68.3**±0.0	83.1	79.2±0.0
	NN3-CVaR-DRO	80.3	76.2±3.1	59.5	58.4±1.9	81.7	73.4±2.0	64.9	62.9±3.2	71.8	**85.1**±8.6	69.0	65.4±1.6	79.2	78.7±0.0	61.6	60.0±0.0	84.3	63.3±0.0	85.3	79.3±0.0
	NN3-χ^2-DRO	79.3	76.2±2.7	62.4	56.1±2.0	80.6	73.0±1.6	66.0	64.6±3.2	74.9	84.0±8.0	68.3	65.7±1.4	79.5	78.3±0.0	63.3	61.3±0.0	84.4	64.1±0.0	85.2	79.5±0.0
	NN4-CVaR-DRO	79.6	76.1±2.8	55.1	53.2±1.9	82.1	72.1±1.8	63.0	63.4±2.6	81.0	82.4±7.2	69.1	65.0±1.7	79.1	78.4±0.0	63.5	60.7±0.0	84.6	67.0±0.0	85.0	79.5±0.0
	NN4-χ^2-DRO	79.6	76.2±2.8	57.8	55.0±2.0	80.8	70.5±1.4	58.8	61.4±1.8	80.1	81.9±7.4	67.2	64.7±1.4	78.7	78.1±0.0	63.7	**61.1**±0.0	84.6	66.3±0.0	84.9	79.4±0.0
Tree-DRO Methods (base: XGB/LGBM)	XGB-CVaR-DRO	79.8	**76.6**±2.7	67.2	61.8±2.0	80.5	**74.7**±1.4	70.7	64.7±4.1	78.1	75.1±6.0	68.3	64.2±1.4	78.2	77.9±0.0	59.3	57.4±0.0	86.8	62.6±0.0	87.2	81.7±0.0
	XGB-KL-DRO	79.2	76.5±2.6	66.4	61.8±1.8	79.8	74.4±1.1	71.3	64.5±4.1	67.3	**78.2**±7.3	68.0	**64.3**±1.9	78.3	77.5±0.0	59.1	**59.6**±0.0	77.8	63.7±0.0	87.6	**82.5**±0.0
	LGBM-CVaR-DRO	79.6	76.5±2.7	67.6	63.1±2.0	73.4	72.6±1.7	69.8	**64.8**±3.7	83.2	74.8±7.4	67.8	64.2±1.4	78.0	76.8±0.0	58.8	57.1±0.0	86.5	64.7±0.0	87.2	81.8±0.0
	LGBM-KL-DRO	79.1	76.5±2.5	66.7	**63.3**±1.9	70.6	72.2±1.6	71.6	64.8±4.0	68.0	77.1±7.9	68.0	63.8±1.5	78.4	76.8±0.0	59.1	57.4±0.0	86.6	**64.8**±0.0	87.2	81.9±0.0
Kernel-DRO Methods (base: Kernel)	Kernel-χ2-DRO	80.6	75.9±3.4	61.8	**59.6**±1.6	82.4	73.3±2.2	68.7	64.7±4.8	79.0	72.6±7.6	68.6	**67.3**±0.7	80.4	78.8±0.0	61.0	**59.7**±0.0	84.2	59.6±0.0	85.0	78.9±0.0
	Kernel-CVaR-DRO	80.5	**76.0**±3.4	61.6	58.9±1.7	84.0	71.9±1.9	69.0	**65.3**±3.8	79.9	**73.3**±7.5	67.4	66.8±0.9	78.5	78.6±0.0	62.0	**59.7**±0.0	83.8	63.7±0.0	84.1	**79.5**±0.0
	Kernel-KL-DRO	80.8	75.9±3.5	60.0	57.8±1.8	81.6	72.9±1.9	69.5	65.0±4.7	78.4	71.6±7.9	67.9	66.8±0.4	79.0	**78.9**±0.0	61.1	59.6±0.0	84.0	**64.4**±0.0	83.6	78.8±0.0
	Kernel-Wasserstein-DRO	80.8	75.8±3.5	57.5	49.8±2.6	82.7	**73.3**±2.0	67.4	61.7±5.0	79.6	71.6±8.3	65.0	60.7±1.8	78.0	77.8±0.0	59.4	56.8±0.0	84.0	60.3±0.0	85.0	**79.5**±0.0
Imbalanced Learning & Fairness Methods (base: XGB)	SUBY	80.4	74.3±4.0	64.2	**62.2**±1.9	82.3	72.4±2.0	69.8	69.0±2.5	84.4	**85.1**±7.6	70.4	**69.5**±0.9	81.1	**79.4**±0.0	64.4	**62.8**±0.0	84.1	**63.3**±0.0	81.2	74.5±0.0
	RWY	80.7	74.4±4.0	64.6	61.8±1.8	74.7	**74.4**±1.3	71.6	**70.0**±2.4	84.3	84.9±7.5	71.0	69.4±0.9	81.1	**79.4**±0.0	64.1	**62.8**±0.0	85.1	63.1±0.0	86.8	81.6±0.0
	SUBG	80.7	**76.0**±3.3	65.2	59.8±1.8	-	-	72.4	66.6±3.9	-	-	70.9	67.3±1.4	81.1	**79.4**±0.0	52.2	56.1±0.0	-	-	85.8	81.4±0.0
	RWG	81.0	76.0±3.4	65.2	59.0±1.9	-	-	72.8	66.5±4.1	-	-	70.8	66.4±1.8	81.1	**79.4**±0.0	62.6	61.4±0.0	-	-	86.9	81.2±0.0
	In-processing	81.0	76.1±3.4	65.3	59.1±2.1	-	-	72.2	66.3±4.4	-	-	70.9	66.9±1.7	80.1	78.8±0.0	63.7	62.6±0.0	-	-	87.5	**82.5**±0.0
	Post-processing	80.4	75.1±3.6	65.4	59.0±2.1	-	-	73.2	66.3±4.2	-	-	70.7	66.8±1.6	79.0	78.9±0.0	62.5	62.7±0.0	-	-	62.5	81.8±0.0

C.2 Detailed Results of Feasibility in 10 Settings

We evaluate each method's feasibility across all 10 settings by averaging over its performance on all target domains. Specifically, for a setting with T target distributions $\{Q_t\}_{t\in[T]}$, we define the *feasibility* of model $\hat{f}$ as:

$$\frac{1}{T}\sum_{t\in[T]}\frac{0.5-\mathbb{E}_{Q_t}[\ell(\hat{f}(X),Y)]}{0.5-\min_{\mathcal{F}}\min_{f\in\mathcal{F}}\mathbb{E}_{Q_t}[\ell(f(X),Y)]}.$$

Figure 11 presents these feasibility scores for each method. As before, in Sections 3.3 and 3.4, we find that DRO shows limited improvements and the DRO performance is strongly correlated with its base model. For brevity, we show only the relative feasibility analog of Figure 4, since the rescaling does not affect the conclusions of Findings 1 and 2 (Sections 3.1 and 3.2).

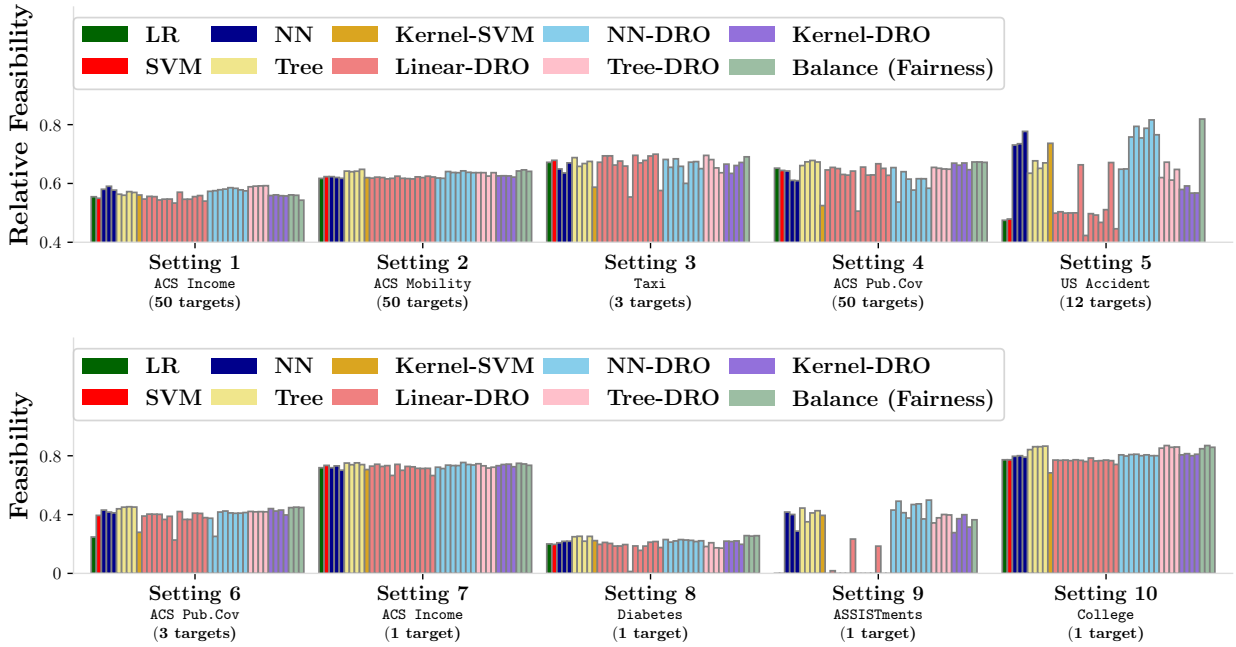


Figure 11. Overall performances (reported as the *feasibility* metric) of all methods on the target data in our 10 settings in Table 1. Here we calculate the average feasibility value among multiple target domains in each setting. Note that for Settings 3, 5 and 9, we do not include fairness-enhancing methods since there are no sensitive attributes.

C.3 Detailed Results of Controlled Shift Settings

We construct a distribution shift instance with controlled perturbations, i.e., by perturbing features and labels from the original database. We focus on the ACS Income dataset in Setting 1 and set

both the source and target domains as data from California while constructing the target domains by perturbing features and (or) labels of the original ACS Income dataset. Denote $x = (x_{\mathcal{N}}, x_{\mathcal{C}})$, where $x_{\mathcal{N}}$ denotes the numerical feature and $x_{\mathcal{C}}$ denotes the categorical feature. We generate each perturbed row of target domains following [58]. More specifically, we allow:

- feature shifts: For each numerical feature $x_j, j \in \mathcal{N}$, in the target domain, we perturb x_j with $x_j + \epsilon_j$, where $\epsilon_j \sim N(0, \text{Var}[x_j]/10)$; for each categorical feature $\ell \in \mathcal{C}$, in the target domain, we perturb x_ℓ with $\tilde{x}_\ell \in \text{domain}(X_\ell)$ with $P(\tilde{x}_\ell = x_\ell) = 0.9$ and $P(\tilde{x}_\ell = u) = \frac{0.1}{|\text{domain}(X_\ell)|-1}$ for $u \in \text{domain}(X_\ell) \backslash \{x_\ell\}$. We perturb each sample independently.
- label shifts: We flip the output of each sample with probability 0.2.

By including at least one shift from the feature or label, we create a "feature perturbation" setting with the following four target domains:

0. No shifts;
1. Only label shifts;
2. Only feature shifts;
3. Both feature and label shifts.

For each target domain, we follow our evaluation pipeline in Section 2, i.e., tune each method based on 128 samples from the target domain, and report results in Table 16. Across all four domains, variants of Wasserstein-DRO (Wasserstein-DRO and Unified-DRO) enjoy relatively good performance, often outperforming the base model since such shifts are captured in corresponding ambiguity sets. When the shifts are severe (e.g., Scenarios 2 or 3), then Unified-DRO achieves better than basic ERM methods across all model classes. This indicates that existing DRO captures such synthetic shifts.

Table 16. Results under the synthetic dataset with 4 target domains, where we run each method with its top-10 configurations (according to the *target* performance and report its mean and standard deviation. We boldface the best target performance within each class of methods in each domain.

		Linear-DRO methods				Basic Methods			
Domain	Metric	Wasserstein-DRO	Unified-DRO-L_2	KL-DRO	χ^2-DRO	SVM	LR	XGB	NN2
0	Accuracy	**80.6**±0.1	**80.6**±0.0	**80.6**±0.0	80.5±0.1	**80.7**±0.0	**80.7**±0.2	80.4±0.6	80.1±0.5
	F1-score	79.8±0.1	**79.9**±0.0	79.8±0.0	79.8±0.2	**80.0**±0.0	79.9±0.2	79.7±0.6	79.5±0.5
1	Accuracy	**79.9**±0.0	**79.9**±0.0	79.6±0.0	79.7±0.0	79.9±0.0	79.6±0.0	**80.1**±0.7	79.2±0.1
	F1-score	79.0±0.0	**79.1**±0.0	78.7±0.0	78.8±0.0	79.0±0.0	78.8±0.0	**79.2**±0.8	78.5±0.2
2	Accuracy	78.6±0.1	**78.9**±0.0	78.4±0.0	78.4±0.0	78.3±0.2	78.2±0.0	**78.7**±1.0	77.0±0.8
	F1-score	77.6±0.1	**77.9**±0.1	77.3±0.0	77.4±0.0	77.2±0.2	77.1±0.0	**77.5**±1.2	76.2±0.8
3	Accuracy	77.0±0.0	**77.7**±0.1	76.2±0.0	76.4±0.0	76.4±0.2	76.4±0.2	**77.0**±0.8	74.6±0.3
	F1-score	75.5±0.1	**76.4**±0.1	74.8±0.1	74.9±0.0	75.0±0.2	74.9±0.2	**75.5**±1.0	73.6±0.5

C.4 Difference between Shift Settings in C.3 and Settings in Table 1

Besides the difference in the selection criterion, we emphasize that the shifts constructed in Appendix C.3 are inherently easier to address. Specifically, for each setting (including both the setting in Appendix C.3 and our spatiotemporal shifts studied in Table 1, where we only choose Settings 1,2,4,5 since there are more than 10 domains in each setting), and for each domain – whether target domains and source domains, we fit a logistic regression model on the data in that domain and obtain the corresponding coefficient vector. This yields a collection $\{\theta^{(j)}\}_{j\in\mathcal{S}}$ over all domains $\mathcal{S}$ for each setting. We then perform principal component analysis (PCA) on these coefficients and compute the proportion of variance explained by the first k components, for $k = 1, 2, \ldots, 5$.

We present the results in Figure 12, where we also compare another "bootstrap" setting over the ACS Income dataset while bootstrapping the training data. Unsurprisingly, the coefficients in the "bootstrap" setting retain substantial variance even when reduced to one or two components. Interestingly, the coefficients in the setting from Appendix C.3 exhibit similar variance explained, approximately 10% higher than our constructed shifts, namely, the "income" setting and related settings such as "accident". For the same "income" setting (Setting 1 and the setting in Appendix C.3), we also compute the covariance matrix of $\{\theta^{(j)}\}_{j\in\mathcal{S}}$ for each setting and find that the largest eigenvalue in Setting 1 is significantly larger than that in the synthetic setting ($0.6 > 0.1$). This highlights the greater difficulty of learning under our constructed shifts compared to the setting in Appendix C.3 with controlled perturbations.

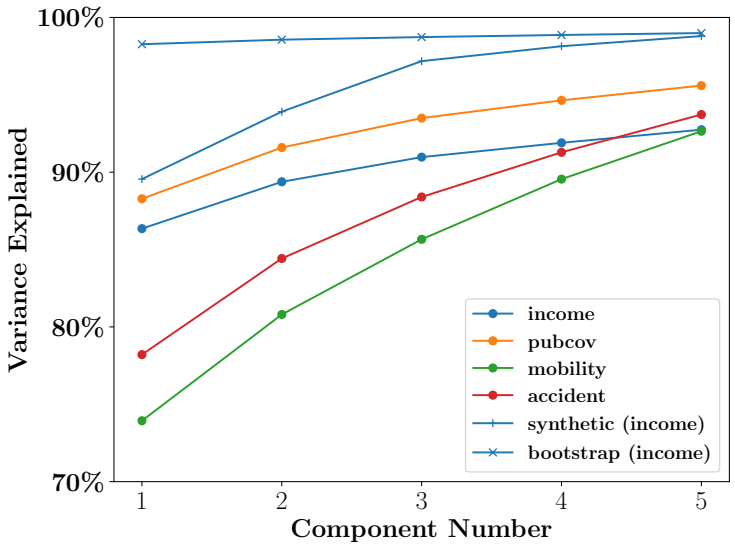


Figure 12. Variance explained by components across different settings, where "income", "pubcov", "mobility", "accident" correspond to Settings 1, 4, 2, 5 in Table 1 and "synthetic (income)" correspond to the setting in Appendix C.3

D Details in Section 4

D.1 Definitions of Variables

Here in Table 17, we provide the exact definitions of each variable in the linear regression used in Section 4.1 and further augmented variables that will be used in Appendix D.3.

Table 17: Definition of independent variables used in (4.1) of Section 4.1

Type	Name	Definition
Model Class $X_{i,j,s}$	LGBM	A dummy variable that takes value one if the underlying model class of the method is LightGBM
	NN-2	A dummy variable that takes value one if the underlying model class of the method is the neural network with 2 layers
	NN-3	A dummy variable that takes value one if the underlying model class of the method is the neural network with 3 layers
	NN-4	A dummy variable that takes value one if the underlying model class of the method is the neural network with 4 layers
	Kernel	A dummy variable that takes value one if the underlying model class of the method is the Kernel method
Ambiguity Set $D_{i,j,s,t}$	Wasserstein	A dummy variable that takes value one if the method belongs to DRO and uses Wasserstein-type metric
	Chi-squared	A dummy variable that takes value one if the method belongs to DRO and uses χ^2-divergence metric
	Kullback-Leibler	A dummy variable that takes value one if the method belongs to DRO and uses KL-divergence metric
	Total Variation	A dummy variable that takes value one if the method belongs to DRO and uses TV-distance metric
	OT-Discrepancy	A dummy variable that takes value one if the method belongs to DRO and uses the optimal transport-discrepancy with conditional moment constraints
	Radius	The rescaled ambiguity size if the method belongs to DRO and equal to zero if the method does not belong to DRO
Shift Pattern $Z_{i,j}$	$Y\|X$-ratio	The $Y\|X$-shift percentage calculated by DISDE from the source domain to the target domain
Validation Type $V_{i,j}$	Worst-case	A dummy variable that takes value one if the best configuration is obtained by maximizing the accuracy on target data from the worst-performing target domain
	Average-case	A dummy variable that takes value one if the best configuration is obtained by maximizing the accuracy on target data across all target domains, averaged on each target domain

Table 18: Additional (Interaction) Variables in Appendix D.3

Name	Definition
Radius^2	The square of the rescaled ambiguity size if the method belongs to DRO methods and zero if the method does not belong to DRO
LGBM-Radius	An interaction variable that takes value to be the robustness radius if the underlying model class of the method is LightGBM
NN2-Radius	An interaction variable that takes value to be the robustness radius if the underlying model class of the method is the neural network with 2 layers
NN3-Radius	An interaction variable that takes value to be the robustness radius if the underlying model class of the method is the neural network with 3 layers
NN4-Radius	An interaction variable that takes value to be the robustness radius if the underlying model class of the method is the neural network with 4 layers
Kernel-Radius	An interaction variable that takes value to be the robustness radius if the underlying model class of the method is kernel-SVM
LGBM-$Y\|X$-ratio	An interaction variable that takes value to be the Y|X-shift percentage if the underlying model class of the method is LightGBM
NN2-$Y\|X$-ratio	An interaction variable that takes value to be the Y|X-shift percentage if the underlying model class of the method is the neural network with 2 layers
NN3-$Y\|X$-ratio	An interaction variable that takes value to be the Y|X-shift percentage if the underlying model class of the method is the neural network with 3 layers
NN4-$Y\|X$-ratio	An interaction variable that takes value to be the Y|X-shift percentage if the underlying model class of the method is the neural network with 4 layers
Kernel-$Y\|X$-ratio	An interaction variable that takes value to be the Y|X-shift percentage if the underlying model class of the method is Kernel SVM

D.2 Scaling of Ambiguity Size

We discuss the choice of rescaled ambiguity size across different DRO methods in Section 4.1. Note that we consider (squared) Wasserstein Distance and OT-Discrepancy metric (i.e. the distance proposed in Unified-DRO [22]), TV, CVaR, KL-divergence, χ^2-divergence in the linear regression. Denote their original ambiguity size hyperparameter in the method training as $\epsilon_{was}, \epsilon_{ot}, \epsilon_{tv}, \alpha_{cvar}, \epsilon_{kl}, \epsilon_{\chi^2}$ and the rescaled standard one as $\epsilon_{was'}, \epsilon_{ot'}, \epsilon_{tv'}, \epsilon_{cvar'}, \epsilon_{kl'}, \epsilon_{\chi^{2'}}$

The unsquared Wasserstein Distance, OT Discrepancy metric, and TV distance already form a distance notion, so we do not distinguish the difference between them. Since we use the squared Wasserstein Distance in the DRO formulation, we set $\epsilon_{was'} = \sqrt{\epsilon_{was}}$ in the variable "Radius" the linear regression model. And $\epsilon_{ot'} = \epsilon_{ot}, \epsilon_{tv'} = \epsilon_{tv}$.

For the KL and χ^2-divergence, although χ^2, KL does not form a distance, we can symmetrize these two divergences such that the augmented version becomes a distance. For example, $\tilde{d}_{KL}(P,Q) = KL(P,Q) + KL(Q,P)$. We do not distinguish the important sizes between KL-divergence and the reversed version. Therefore, we set $\epsilon_{kl'} = 2\epsilon_{kl}, \epsilon_{\chi^{2'}} = 2\epsilon_{\chi^2}$.

For the CVaR metric, denote $D_\infty(P||Q) = \operatorname{ess\,sup} \frac{dP}{dQ}$. Then the uncertainty region becomes the distribution set that CVaR operates on. That is,

$$\mathcal{P} := \left\{ P | D_\infty(P||\widehat{P}) \leq \log \frac{1}{\alpha} \right\} = \left\{ P | \text{there exists } Q, \beta \in [\alpha, 1], \text{s.t. } \widehat{P} = \beta P + (1-\beta)Q \right\}.$$

Therefore, we take $\epsilon_{cvar'} = \log \frac{1}{\alpha_{cvar}}$.

D.3 Linear Regression Results for Other Settings

We include linear regression results with respect to `Best Config` and `Worst Domain` in other settings in Table 19, where similar patterns hold as in Section 4.1.

D.4 Fine-grained Comparison between the Model Class and the Choice of the Ambiguity Set

In this subsection, we conduct a fine-grained comparison of accuracy improvements resulting from incremental changes in model class versus algorithmic design of the ambiguity set.

We examine accuracy variations across different methods that involve two types of interventions:

- Model class changes: Increasing incremental model complexity from linear models to neural networks with the number of hidden layers in $\{2, 3, 4\}$, and to LightGBM with the number of base estimators in $\{64, 128, 256\}$.
- DRO ambiguity set designs: Incorporating distributional robustness beyond standard ERM training, through χ^2-DRO and CVaR-DRO in NNs, and KL-DRO and CVaR-DRO in LGBM, using the optimal ambiguity size that maximizes the accuracy in each corresponding target domain. We limit our comparison to these DRO variants with optimal radii, as they are the formulations supported for the respective model classes in our implementation and yield the best accuracies among the considered methods.

Table 19. Regression results on algorithmic design components on the method performance in all other settings besides in Table 4

		Dependent variable: Target accuracy							
		`Best Config`				`Worst Domain`			
Variable Name		Setting 3	Setting 4	Setting 2	Setting 6	Setting 3	Setting 5	Setting 1	Setting 6
Model Class	LGBM	-0.0234^{***}	-0.0189^{***}	0.0025^{***}	0.0224^{***}	-0.0435^{***}	0.0142^{***}	0.0054^{*}	0.0061^{***}
		(.0039)	(.0017)	(.0006)	(.0034)	(.0049)	(.0030)	(.0029)	(.0014)
	NN2	-0.0108^{***}	-0.0105^{***}	0.0039^{***}	0.0122^{***}	-0.0934^{***}	0.0540^{***}	-0.0221^{***}	-0.0288^{***}
		(.0040)	(.0018)	(.0006)	(.0034)	(.0057)	(.0038)	(.0036)	(.0023)
	NN3	-0.0097^{**}	-0.0249^{***}	0.0022^{***}	0.0112^{***}	-0.1868^{***}	0.0516^{***}	-0.0175^{***}	-0.0201^{***}
		(.0042)	(.0018)	(.0006)	(.0034)	(.0057)	(.0038)	(.0037)	(.0025)
	NN4	-0.0215^{***}	-0.0367^{***}	-0.0005	0.0070^{**}	-0.1839^{***}	0.0573^{***}	-0.0158^{***}	-0.0248^{***}
		(.0044)	(.0019)	(.0006)	(.0035)	(.0059)	(.0037)	(.0036)	(.0030)
	Kernel	-0.0165^{***}	-0.0101^{***}	-0.0015^{***}	-0.0063^{**}	-0.0789^{***}	0.0098^{***}	-0.0058^{**}	-0.0010
		(.0033)	(.0016)	(.0005)	(.0030)	(.0049)	(.0030)	(.0029)	(.0015)
Ambiguity Set	Wasserstein	0.0016	0.0046^{**}	-0.0005	0.0051	-0.0382^{***}	-0.0140^{**}	0.0000	-0.0070^{**}
		(.0043)	(.0020)	(.0006)	(.0037)	(.0064)	(.0064)	(.0050)	(.0030)
	Chi-squared	-0.0006	0.0005	0.0015^{***}	0.0075^{***}	0.0053	0.0040^{*}	0.0159^{***}	0.0103^{***}
		(.0029)	(.0014)	(.0004)	(.0024)	(.0037)	(.0023)	(.0023)	(.0014)
	Kullback-Leibler	-0.0015	0.0047^{***}	-0.0031^{***}	-0.0002	0.0188^{***}	-0.0255^{***}	-0.0006	0.0011
		(.0037)	(.0015)	(.0005)	(.0035)	(.0043)	(.0027)	(.0025)	(.0013)
	Total Variation	-0.0067	-0.0098^{***}	-0.0030^{***}	0.0032	-0.0727^{***}	-0.0410^{***}	-0.0291^{***}	-0.0026
		(.0059)	(.0026)	(.0008)	(.0051)	(.0091)	(.0055)	(.0053)	(.0025)
	OT-Discrepancy	-0.0039	-0.0046^{*}	-0.0015^{*}	-0.0055	-0.0040	-0.0126^{**}	0.0220^{***}	-0.0073^{***}
		(.0059)	(.0024)	(.0009)	(.0050)	(.0084)	(.0052)	(.0048)	(.0024)
	Radius	0.0001	0.0005	0.0000	0.0051^{***}	-0.0213^{***}	-0.0101^{***}	-0.0048^{***}	-0.0030^{***}
		(.0028)	(.0016)	(.0004)	(.0011)	(.0013)	(.0009)	(.0009)	(.0004)
Shift Pattern	$Y\|X$-ratio	0.1976^{***}	-0.1218^{***}	-0.0052^{***}	0.1967^{***}	-	-	-	-
		(.0019)	(0.0007)	(.0001)	(.0010)				
Validation Type	Worst-case	0.0114^{***}	-0.0229^{***}	-0.0002	-0.0018	-	-	-	-
		(.0026)	(.0011)	(.0004)	(.0022)				
	Average-case	0.0134^{***}	0.0028^{***}	0.0015^{***}	0.0012	-	-	-	-
		(.0026)	(.0011)	(.0004)	(.0022)				
Fixed Effect	Setting	No	No	No	No	No	No	No	No
	Domain	Yes	Yes	Yes	Yes	No	No	No	No
Overall	N	287	3671	3671	287	5056	4500	4321	3352
	Adjusted R^2	0.878	0.841	0.929	0.868	0.356	0.251	0.053	0.135

Notes. ***,** and * show statistical significance at the 1%, 5%, and 10% levels using two-tailed tests, respectively.

Neural Network Analysis. In Figure 13, we report the accuracy gains comparing these two interventions across all distribution shift pairs. Panels $(a)(d)$ in Figure 13 show that when the baseline is a linear model (e.g., SVM), changing the model class to NN2 yields larger accuracy improvements than imposing DRO on the linear model. However, when the base model class is already a neural network, the DRO effect becomes relatively more significant, as shown in panels $(b)(c)(e)(f)$.

We further corroborate these patterns in Figure 14, which aggregates accuracy differences across all settings as well as representative configurations (Settings 1 and 5), and shows that model class upgrades dominate accuracy gains for linear baselines, while DRO effects remain secondary and model-dependent for neural networks.

This pattern indicates that DRO interventions yield significant improvements when applied across incremental changes within the neural network model class. However, this finding must be interpreted carefully: while DRO applied to NN2 shows substantial relative gains, neural network-based methods overall achieve lower absolute accuracy in our problem setup, indicating that neural networks are not well-suited to our semi-synthetic shifts. Consequently, despite the apparent effectiveness of DRO within the neural network family, these methods do not achieve the best overall performance due to the inherent limitations of the neural network model class for our problems. This interpretation is further supported by panels (g) and (h) in Figure 13, which demonstrate that transitioning from NN2 to LightGBM produces substantially larger accuracy gains than any DRO intervention, highlighting the primacy of model class selection over robustness design when models are mismatched to the problem structure.

LightGBM Analysis. To better isolate the effects of fine-grained model class changes from ambiguity set design, we conduct additional experiments focusing on interventions of LightGBM, which proves to be a better model class with high accuracy in Section 4. We represent fine-grained model class changes by controlling the number of base estimators and tuning over other hyperparameters.

As shown in all panels of Figure 15, when using a well-suited model class like LightGBM, regardless of the specific base model class, fine-grained changes in model complexity consistently outperform the effects of DRO ambiguity set design. The aggregated results in Figure 16 further support this finding. Across all settings, incorporating distributional robustness does not even provide improvements when the base model class is LightGBM.

Summary. Our analysis suggests that when the model class is appropriately suitable to the problem (e.g., LightGBM in our case), incremental improvements in model complexity tend to

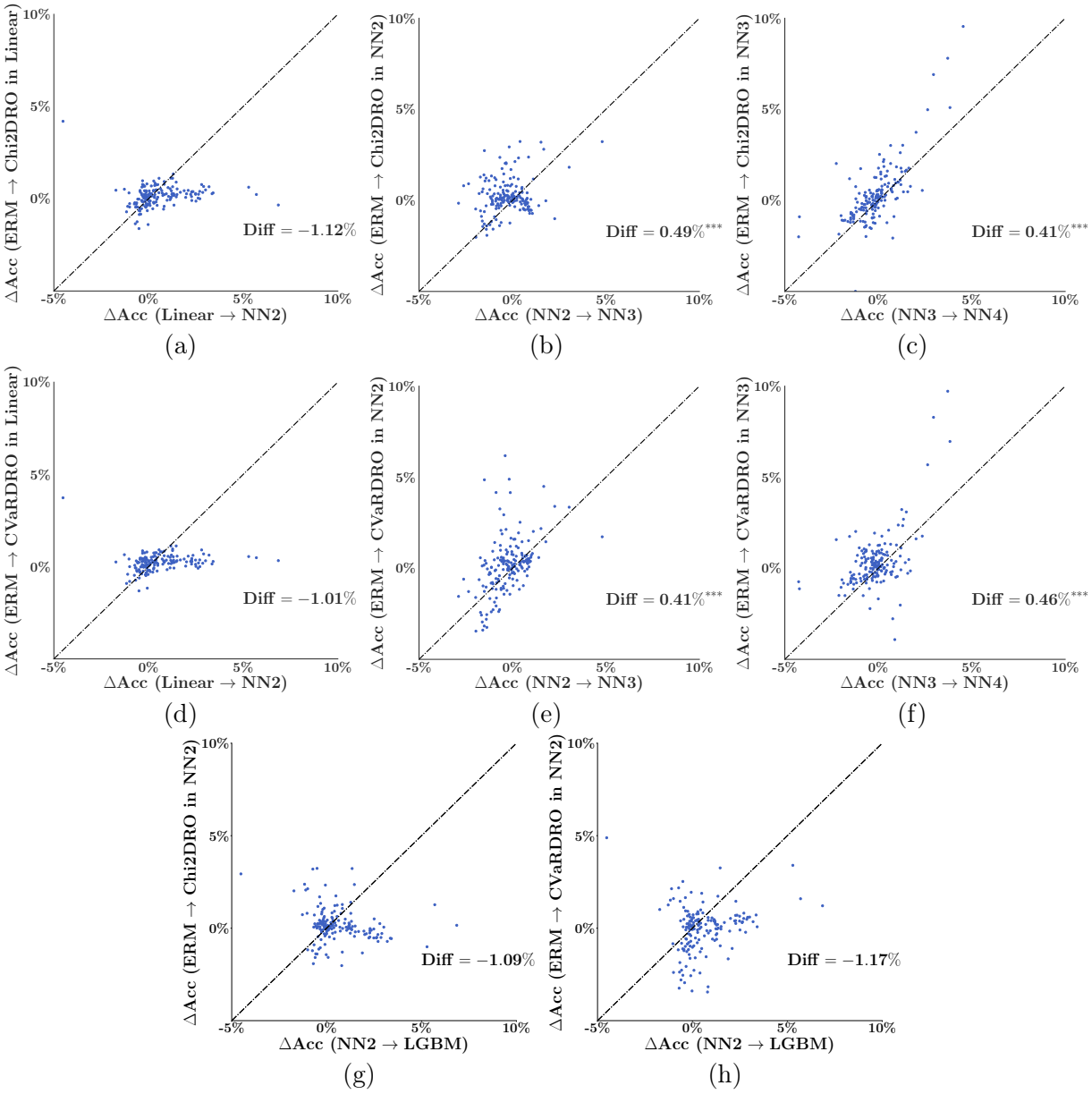
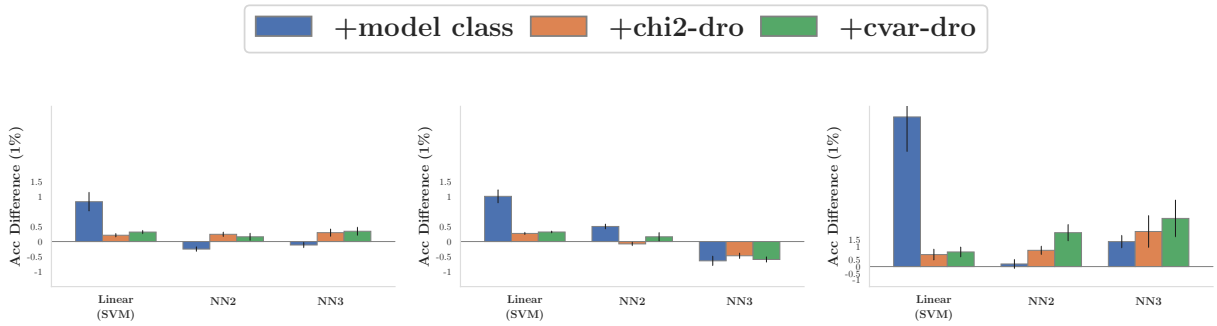


Figure 13. Scatter plot comparing accuracy gains from interventions across distribution pairs and settings. The horizontal axis shows the accuracy improvement from model-class changes, while the vertical axis shows the accuracy improvement from DRO ambiguity set designs. Each point corresponds to one source–target pair. The difference between the two interventions (vertical minus horizontal) is reported within each panel. Superscripts *,**, and *** indicate statistical significance at the 1%, 5%, and 10% levels, respectively, based on one-sided tests of the null hypothesis that the accuracy gain from model-class changes exceeds that from DRO ambiguity design.

provide larger accuracy gains than distributional robustness interventions. We acknowledge the limitations of this empirical analysis and leave further theoretical quantification of the trade-offs between fine-grained model class selection and robustness design as future work.



(a) All settings (b) Setting 1 (c) Setting 5

Figure 14. Average accuracy gains (and the standard deviations) from increasing model class (linear to neural networks varying the number of layers versus DRO ambiguity set design among all domains in each setting. Here, "+model class" denotes incremental increases in model complexity (e.g., linear to NN2, NN2 to NN3, NN3 to NN4). Each bar group corresponds to a fixed base model class.

D.5 (Non-)Linear Regression Results for Ablation Studies

We consider the interaction effects of model class and shift patterns, and the nonlinear effects of the radius in the ambiguity set in the linear regression (4.1) and provide results in Tables 20 and 21. We provide full definitions of variables in Table 17 and include three categories of variables: Radius^2 as the nonlinear effect of the radius in the ambiguity set, interaction terms between the model class and shift patterns, and interaction terms between the model class and robustness radius in Table 18. We also change the dependent variable to the performance gap (2.2) in (4.1) while keeping all these independent variables in Table 17 to understand the effects of robust models against distribution shifts and report results in Table 23.

Nonlinear effects of the radius and interaction effects of the model class and radius. In Tables 20 and 21, we introduce the nonlinear effect of the radius in the ambiguity set and interaction effects between model class and radius. This enlarges the coefficient size of the nonlinear and linear components in all settings. When aggregating all settings, the coefficient size of the radius for linear models ("Radius") is significantly positive in Table 20 compared with the negative coefficient size of the radius for nonlinear models ("Radius" + "XX-Radius"). This has two implications: (i) Adding robustness does not improve performance for nonlinear models; (ii) For linear models, it may provide slight improvements when the optimal ambiguity radius is not large. Taking into account the scaled size (with Radius <0.5 in general DRO models and Radius <0.1 in Wasserstein DRO observed from the data), the ultimate aggregated performance improvement is approximately 0.005, which is still smaller than the improvement attributable to the model class. Therefore, the relative improvement incorporating DRO for linear models is not large compared to changing the model class.

Table 20. Regression results on algorithmic design components (`Best Config`, incorporating nonlinear effects of the radius and additional interaction effects between the model class and robustness radius, between the model class and shift patterns) on the method performance

Variable Name		Dependent variable: Accuracy						
		All	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6
Model Class	LGBM	0.0102***	0.0169***	0.0101***	−0.0162**	−0.0103***	0.0567***	0.0313***
		(.0022)	(.0016)	(.0009)	(.0077)	(.0027)	(.0072)	(.0057)
	NN2	0.0157***	0.0140***	0.0050***	0.0051	0.0055**	0.0725***	0.0299***
		(.0020)	(.0016)	(.0008)	(.0074)	(.0027)	(.0067)	(.0057)
	NN3	0.0131***	0.0119***	0.0000	0.0135*	−0.0044	0.0761***	0.0273***
		(.0022)	(.0016)	(.0009)	(.0072)	(.0028)	(.0067)	(.0057)
	NN4	0.0088***	0.0149***	−0.0027***	0.0227***	−0.0067**	0.0803***	0.0271***
		(.0025)	(.0017)	(.0009)	(.0079)	(.0029)	(.0081)	(.0062)
	Kernel	0.0025	0.0012	−0.0041***	0.0029	−0.0207***	0.0426***	−0.0206***
		(.0018)	(.0014)	(.0006)	(.0062)	(.0022)	(.0055)	(.0050)
Ambiguity Set	Wasserstein	0.0058***	0.0020**	0.0024***	0.0078*	0.0160***	−0.0086	0.0216***
		(.0019)	(.0008)	(.0007)	(.0041)	(.0018)	(.0060)	(.0033)
	Chi-squared	0.0006	0.0015**	0.0032***	−0.0006	−0.0105***	−0.0027	0.0139***
		(.0013)	(.0008)	(.0006)	(.0048)	(.0016)	(.0075)	(.0031)
	Kullback-Leibler	0.0003	0.0021***	0.0040***	0.0013	0.0008	−0.0095*	−0.0011
		(.0015)	(.0008)	(.0007)	(.0046)	(.0019)	(.0052)	(.0030)
	Total Variation	−0.0054**	−0.0006	−0.0027***	−0.0024	−0.0153***	−0.0034	0.0043
		(.0024)	(.0010)	(.0009)	(.0058)	(.0027)	(.0118)	(.0045)
	OT-Discrepancy	−0.0008	−0.0020**	−0.0002	0.0004	−0.0026	−0.0058	0.0029
		(.0024)	(.0010)	(.0009)	(.0058)	(.0030)	(.0081)	(.0045)
	Radius	0.0124***	0.0033	−0.0147***	−0.0005	0.0153**	−0.0221	0.0254***
		(.0021)	(.0024)	(.0014)	(.0109)	(.0077)	(.0473)	(.0053)
	Radius2	−0.0011	−0.0001	0.0175***	0.0048	0.0004	0.0224	−0.0137***
		(.0007)	(.0020)	(.0011)	(.0081)	(.0059)	(.0493)	(.0012)
	LGBM-Radius	−0.0111***	0.0014	−0.0292***	−0.0163**	−0.0238***	−0.0186	−0.0077
		(.0026)	(.0015)	(.0021)	(.0079)	(.0040)	(.0128)	(.0057)
	NN2-Radius	−0.0355***	−0.1920	−0.0049	−0.0164	−0.0478***	0.9804	−0.0753***
		(.0061)	(.1233)	(.0100)	(.0107)	(.0052)	(.1164)	(.0173)
	NN3-Radius	−0.0309***	0.1406	0.0109***	−0.6364	−0.0499***	0.1362	−0.0513***
		(.0048)	(.1233)	(.0017)	(.7382)	(.0053)	(.1255)	(.0173)
	NN4-Radius	−0.0314***	−0.0041	0.0196***	−0.6235**	−0.1044***	0.0109	−0.0906***
		(.0091)	(.0063)	(.0034)	(.7382)	(.0113)	(.0354)	(.0257)
	Kernel-Radius	−0.0028	−0.0017	0.0079***	0.0182	0.0818***	−0.0043	0.0587***
		(.0027)	(.0011)	(.0013)	(.0112)	(.0059)	(.0253)	(.0069)
Shift Pattern	$Y\|X$-ratio	−0.0038***	0.0689***	−0.0056***	0.2029***	0.2029***	−0.0056***	0.1958***
		(.0004)	(.0014)	(.0002)	(.0024)	(.0012)	(.0004)	(.0013)
	LGBM-$Y\|X$-ratio	0.0001	−0.0030	0.0016***	0.0050	0.0123***	0.0008	0.0085
		(.0007)	(.0019)	(.0003)	(.0075)	(.0030)	(.0007)	(.0085)
	NN2-$Y\|X$-ratio	−0.0010	−0.0030	0.0006**	−0.0118	0.0100***	0.0008	−0.0106
		(.0007)	(.0019)	(.0003)	(.0075)	(.0030)	(.0007)	(.0085)
	NN3-$Y\|X$-ratio	−0.0012*	−0.0026	0.0004	−0.0234***	0.0063**	0.0009	−0.0121
		(.0007)	(.0019)	(.0003)	(.0075)	(.0030)	(.0007)	(.0085)
	NN4-$Y\|X$-ratio	−0.0013*	−0.0033*	0.0009***	−0.0406***	0.0097***	0.0009	−0.0120
		(.0007)	(.0019)	(.0003)	(.0075)	(.0030)	(.0007)	(.0085)
	Kernel-$Y\|X$-ratio	−0.0008	0.0016	0.0002	−0.0324***	−0.0037	0.0004	−0.0091
		(.0006)	(.0016)	(.0002)	(.0064)	(.0025)	(.0006)	(.0072)
Validation Type	Worst-case	−0.0056***	0.0050***	−0.0002	0.0114***	−0.0229***	−0.0067**	−0.0018
		(.0010)	(.0004)	(.0003)	(.0023)	(.0009)	(.0033)	(.0017)
	Average-case	0.0054***	0.0078***	0.0015***	0.0134***	0.0028***	0.0202***	0.0012
		(.0010)	(.0004)	(.0003)	(.0023)	(.0009)	(.0033)	(.0017)
Fixed Effect	Setting	Yes	No	No	No	No	No	No
	Domain	No	Yes	Yes	Yes	Yes	Yes	Yes
Overall	N	12527	3671	3671	287	3671	935	287
	Adjusted R^2	0.196	0.775	0.937	0.903	0.873	0.776	0.921

Notes. ***,** and * show statistical significance at the 1%, 5%, and 10% levels using two-tailed tests, respectively.

Table 21. Regression results on algorithmic design components (`Worst Domain`, incorporating nonlinear effects of the radius and additional interaction effects between the model class and shift patterns) on the method performance

Variable Name		Dependent variable: Accuracy						
		All	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6
Model Class	LGBM	0.0004	0.0034	0.0059**	−0.0311***	−0.0120***	0.0241***	0.0151***
		(.0017)	(.0034)	(.0025)	(.0057)	(.0021)	(.0036)	(.0016)
	NN2	0.0022	−0.0068	0.0077*	−0.0509***	−0.0117***	0.0634***	−0.0130***
		(.0023)	(.0042)	(.0041)	(.0066)	(.0035)	(.0044)	(.0026)
	NN3	−0.0253***	0.0004	0.0064	−0.1512***	−0.0299***	0.0586***	−0.0003
		(.0023)	(.0043)	(.0050)	(.0066)	(.0038)	(.0045)	(.0029)
	NN4	−0.0242***	−0.0090**	−0.0304***	−0.1480***	−0.0600***	0.0591***	−0.0097***
		(.0024)	(.0042)	(.0054)	(.0071)	(.0046)	(.0044)	(.0037)
	Kernel	−0.0244***	−0.0113***	−0.0065**	−0.0856***	−0.0346***	0.0018	0.0003
		(.0018)	(.0034)	(.0026)	(.0057)	(.0020)	(.0035)	(.0017)
Ambiguity Set	Wasserstein	−0.0161***	0.0029	0.0019	−0.0342***	−0.0196***	−0.0035	−0.0025
		(.0027)	(.0049)	(.0064)	(.0063)	(.0038)	(.0063)	(.0029)
	Chi-squared	0.0099***	0.0174***	0.0133***	0.0034	0.0001	0.0074***	0.0108***
		(.0013)	(.0023)	(.0022)	(.0037)	(.0017)	(.0023)	(.0014)
	Kullback-Leibler	0.0016	−0.0056**	0.0012	0.0182***	0.0026*	−0.0134***	0.0014
		(.0014)	(.0027)	(.0020)	(.0046)	(.0016)	(.0028)	(.0014)
	Total Variation	−0.0305***	−0.0304***	−0.0087**	−0.0650***	−0.0400***	−0.0365***	0.0011
		(.0027)	(.0052)	(.0036)	(.0089)	(.0032)	(.0054)	(.0025)
	OT-Discrepancy	−0.0021	0.0212***	−0.0043	0.0053	−0.0304***	−0.0083	−0.0014
		(.0025)	(.0048)	(.0034)	(.0084)	(.0029)	(.0053)	(.0024)
	Radius	0.0011	−0.0018	0.0054***	−0.0153***	0.0299***	−0.0108***	0.0076***
		(.0013)	(.0026)	(.0018)	(.0041)	(.0015)	(.0027)	(.0013)
	Radius2	−0.0006***	−0.0007**	−0.0006**	0.0011**	−0.0031***	0.0000	−0.0012***
		(.0002)	(.0004)	(.0003)	(.0005)	(.0002)	(.0004)	(.0002)
	LGBM-Radius	−0.0167***	0.0096***	−0.0127***	−0.0161***	−0.0363***	−0.0345***	−0.0154***
		(.0016)	(.0032)	(.0021)	(.0053)	(.0018)	(.0034)	(.0015)
	NN2-Radius	−0.0410***	−0.0721***	−0.0391***	−0.0474***	−0.0193***	−0.0229***	−0.0290***
		(.0017)	(.0074)	(.0022)	(.0040)	(.0021)	(.0064)	(.0027)
	NN3-Radius	−0.0548***	−0.0883***	−0.0428***	−0.0389***	−0.0112***	−0.0179**	−0.0357***
		(.0019)	(.0076)	(.0029)	(.0040)	(.0029)	(.0081)	(.0031)
	NN4-Radius	−0.0995***	−0.0522***	−0.0541***	−0.0893***	0.0132	0.0004	−0.0696***
		(.0055)	(.0114)	(.0154)	(.0118)	(.0104)	(.0100)	(.0175)
	Kernel-Radius	0.0019**	0.0055***	−0.0078***	0.0099***	−0.0054***	0.0103***	−0.0014
		(.0009)	(.0018)	(.0013)	(.0031)	(.0011)	(.0019)	(.0009)
Fixed Effect	Setting	No	No	No	No	No	No	No
	Domain	No	No	No	No	No	No	No
Overall	N	23776	4321	3212	5056	3330	4500	3352
	Adjusted R^2	0.483	0.113	0.267	0.410	0.340	0.291	0.205

Notes. ***,** and * show statistical significance at the 1%, 5%, and 10% levels using two-tailed tests, respectively.

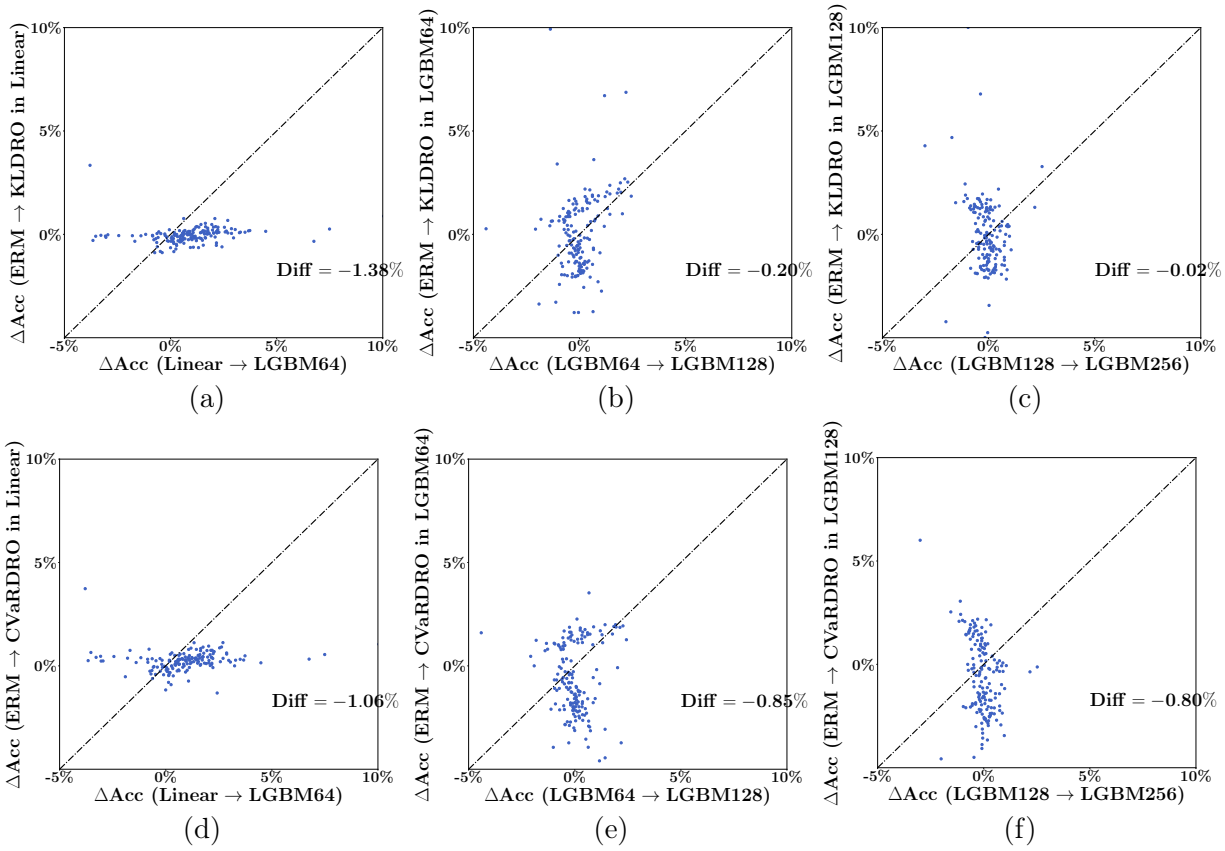


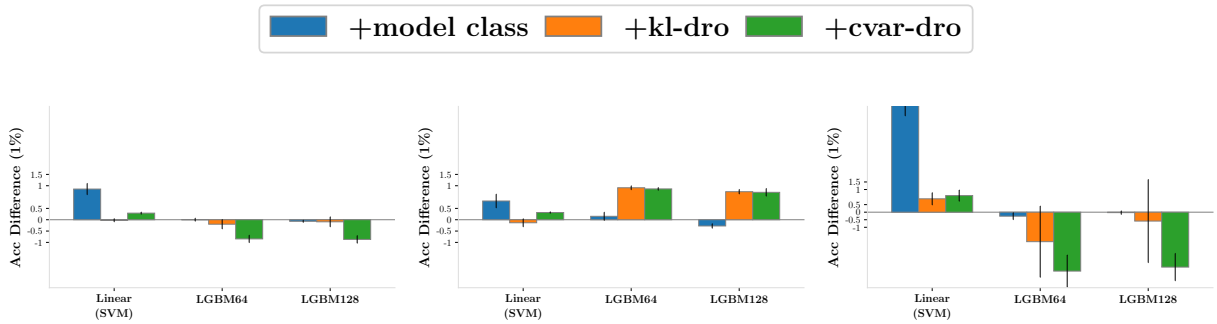
Figure 15. Scatter plot comparing accuracy gains from interventions across distribution pairs and settings. The horizontal axis shows the accuracy improvement from model-class changes, while the vertical axis shows the accuracy improvement from DRO ambiguity set designs. Each point corresponds to one source-target pair. The difference between the two interventions (vertical minus horizontal) is reported within each panel. Superscripts *,**, and *** indicate statistical significance at the 1%, 5%, and 10% levels, respectively, based on one-sided tests of the null hypothesis that the accuracy gain from model-class changes exceeds that from DRO ambiguity design.

For the nonlinear effects of the radius in the ambiguity set, the coefficients of the quadratic and linear components usually have opposite signs. When plugging the usual choices of the model configuration (with Radius < 0.5) and the optimal radius from the coefficient, the effects on the model accuracy are still relatively small across all settings. For specific settings, the approximated largest effect of the radius on the accuracy is $(-0.014\text{Radius}^2 + 0.025\text{Radius})_{\max} = 0.004$ in Setting 6 in Table 20, which is still dominated by the effect of the model class in that setting.

Subgroup regression per model class. In Table 22, we run one regression for methods under each model class and report the coefficients of the ambiguity set and other controlled variables. We observe that for most competitive model classes, e.g., linear, LightGBM, NN2, introducing the distributionally robust counterpart does not yield substantial performance gains. The only notable

Table 22. Regression results on algorithmic design components (`Best Config`) on the method performance for each model class across all six settings

Variable Name		Dependent variable: Accuracy			
		Linear	LGBM	Kernel	NN2
Ambiguity Set	Wasserstein	-0.0030	-	0.0086^{**}	-
		(.0028)		(.0037)	
	Chi-squared	-0.0003	-	0.0080^{***}	0.0013
		(.0025)		(.0028)	(.0026)
	Kullback-Leibler	-0.0013	-0.0076^{***}	0.0107^{**}	-
		(.0026)	(.0028)	(.0025)	
	Total Variation	-0.0065^{***}	-	-	-
		(.0026)			
	OT-Discrepancy	-0.0040	-	-	-
		(.0026)			
	Radius	0.0015	-0.0023	0.0076^{***}	-0.0229^{***}
		(.0030)	(.0018)	(.0022)	(.0058)
Shift Pattern	$Y\|X$-ratio	-0.0049^{***}	-0.0038^{***}	-0.0046^{***}	-0.0041^{***}
		(.0004)	(.0005)	(.0004)	(.0005)
Validation Type	Worst-case	-0.0007	-0.0198^{***}	-0.0008	-0.0016
		(.0020)	(.0028)	(.0024)	(.0027)
	Average-case	0.0008	0.0036	0.0058^{**}	0.0097^{***}
		(.0019)	(.0030)	(.0024)	(.0028)
Fixed Effect	Setting	Yes	Yes	Yes	Yes
	Domain	No	No	No	No
Overall	N	3654	1566	2610	1566
	Adjusted R^2	0.174	0.271	0.178	0.288



(a) All settings (b) Setting 1 (c) Setting 5

Figure 16. Average accuracy gains (and the standard deviations) from increasing model class (linear to LightGBM varying the number of base estimators versus DRO ambiguity set design among all domains in each setting. Here, "+model class" denotes incremental increases in model complexity (e.g., linear to LightGBM with 64 estimators, LightGBM with 64 estimators to 128 estimators). Each bar group corresponds to a fixed base model class.

exception is the kernel-DRO methods, which demonstrate significant improvements over the ERM counterparts (i.e., Kernel-SVM) when averaged across target domains. However, this improvement is partly attributable to the increased variability introduced by the random projections used in kernel methods, and the relatively poor performance of Kernel-SVM compared to linear and tree-based ensemble models. In fact, the gains from switching model classes are generally much larger than those obtained by adding robustness within a given class.

Interaction effects of model class and shift patterns. For interaction effects of the model class and shift patterns, we do not observe significantly consistent coefficients among different settings and the coefficient sizes for them are too small (smaller than ≈ 0.01). This means that when the $Y|X$-shift is larger, no particular model class (e.g., LightGBM or NN or linear model class) suffers a larger performance drop. We observe similar coefficient sizes when we consider the interaction effects of the ambiguity set and shift patterns. This aligns with previous empirical findings in Section 3.2.

Setting the dependent variable to the performance gap. In Tables 23 and 24, we investigate how DRO models can mitigate the performance gap under natural distribution shifts by isolating the effects of the model class on the target model performance. The coefficient of the tree-based ensemble is negative, which is expected, as vanilla tree-based ensembles do not provide reliable improvements under distribution shifts. However, the coefficient sizes across various ambiguity set designs, including different distance types and radii, are not consistently larger than those of the neural network models in both the overall and most specific settings. This suggests that DRO models fail to sufficiently mitigate performance gaps.

Table 23. Regression results on algorithmic design components (`Best Config`, incorporating nonlinear effects of the radius and additional interaction effects between the model class and shift patterns) on the performance gap, i.e, **target accuracy - source accuracy**.

Variable Name		Dependent variable: Performance gap						
		All	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6
Model Class	LGBM	−0.0165***	0.0057***	−0.0212***	−0.0425***	−0.0452***	−0.0201***	−0.0027
		(.0021)	(.0015)	(.0009)	(.0082)	(.0025)	(.0072)	(.0037)
	NN2	0.0064***	0.0050***	0.0045***	−0.0109	−0.0021	0.0213***	0.0099***
		(.0020)	(.0015)	(.0008)	(.0080)	(.0025)	(.0067)	(.0037)
	NN3	0.0025	0.0016	0.0032***	−0.0090	−0.0087***	0.0174***	0.0098***
		(.0021)	(.0015)	(.0009)	(.0077)	(.0025)	(.0067)	(.0037)
	NN4	0.0034	0.0056***	0.0018*	−0.0006	−0.0040	0.0189**	0.0105***
		(.0024)	(.0016)	(.0010)	(.0085)	(.0027)	(.0082)	(.0040)
	Kernel	−0.0017	−0.0003	0.0048***	0.0044	−0.0191***	0.0104*	−0.0046
		(.0017)	(.0013)	(.0007)	(.0067)	(.0020)	(.0055)	(.0032)
Ambiguity Set	Wasserstein	0.0007	−0.0030***	−0.0048***	−0.0107**	0.0051***	−0.0086	0.0059***
		(.0018)	(.0008)	(.0007)	(.0044)	(.0016)	(.0060)	(.0021)
	Chi-squared	0.0001	0.0039***	−0.0075***	0.0003	−0.0022	−0.0011	0.0034*
		(.0013)	(.0008)	(.0006)	(.0052)	(.0015)	(.0076)	(.0019)
	Kullback-Leibler	0.0015	0.0058***	−0.0071***	0.0007	0.0001	−0.0069	−0.0027
		(.0014)	(.0007)	(.0007)	(.0049)	(.0017)	(.0052)	(.0019)
	Total Variation	−0.0031	−0.0026***	0.0000	−0.0002	−0.0058**	0.0002	0.0004
		(.0023)	(.0010)	(.0010)	(.0062)	(.0025)	(.0118)	(.0029)
	OT-Discrepancy	0.0040*	0.0018*	−0.0013	−0.0007	0.0100***	−0.0034	0.0024
		(.0023)	(.0010)	(.0009)	(.0062)	(.0027)	(.0081)	(.0028)
	Radius	0.0004	−0.0063***	0.0123***	−0.0091	−0.0067	−0.0283	0.0056*
		(.0021)	(.0023)	(.0014)	(.0117)	(.0071)	(.0475)	(.0034)
	Radius2	0.0011	0.0037**	−0.0199***	0.0086	0.0042	0.0226	−0.0029***
		(.0007)	(.0019)	(.0012)	(.0087)	(.0054)	(.0495)	(.0007)
	LGBM-Radius	0.0024	0.0088***	0.0382***	−0.0096	0.0014	−0.0086	−0.0038
		(.0025)	(.0014)	(.0022)	(.0084)	(.0037)	(.0129)	(.0036)
	NN2-Radius	−0.0118**	−0.6542***	0.0652***	−0.0092	−0.0181***	0.0714	−0.0205*
		(.0060)	(.1183)	(.0103)	(.0115)	(.0048)	(.1214)	(.0110)
	NN3-Radius	−0.0084*	−0.0834	0.0083***	0.4106	−0.0266***	0.1396	0.0000
		(.0047)	(.1183)	(.0018)	(.7917)	(.0049)	(.1260)	(.0110)
	NN4-Radius	−0.0133	−0.0058	0.0435***	−0.5412*	−0.0908***	0.0176	0.0144
		(.0089)	(.0061)	(.0035)	(.7917)	(.0104)	(.0356)	(.0163)
	Kernel-Radius	−0.0061**	−0.0072***	−0.0041***	−0.0194	0.0282***	−0.0007	0.0129***
		(.0027)	(.0011)	(.0014)	(.0120)	(.0054)	(.0254)	(.0044)
Shift Pattern	$Y\|X$-ratio	−0.0046***	−0.0290***	−0.0074***	−0.0726***	−0.1681***	−0.0059***	0.0145***
		(.0004)	(0.0013)	(.0002)	(.0026)	(0.0007)	(.0004)	(.0008)
	LGBM-$Y\|X$-ratio	0.0016**	−0.0030*	0.0016***	0.0050	0.0124***	0.0008	0.0085
		(.0007)	(.0018)	(.0003)	(.0081)	(.0027)	(.0007)	(.0054)
	NN2-$Y\|X$-ratio	0.0001	−0.0030*	0.0006**	−0.0118	0.0100***	0.0008	−0.0106*
		(.0007)	(.0018)	(.0003)	(.0081)	(.0027)	(.0007)	(.0054)
	NN3-$Y\|X$-ratio	0.0001	−0.0026	0.0004	−0.0234***	0.0063**	0.0009	−0.0121**
		(.0007)	(.0018)	(.0003)	(.0081)	(.0027)	(.0007)	(.0054)
	NN4-$Y\|X$-ratio	0.0001	−0.0033*	0.0009***	−0.0406***	0.0097***	0.0009	−0.0120**
		(.0007)	(.0018)	(.0003)	(.0081)	(.0027)	(.0007)	(.0054)
	Kernel-$Y\|X$-ratio	−0.0001	0.0016	0.0002	−0.0324***	−0.0037	0.0004	−0.0091**
		(.0006)	(.0015)	(.0002)	(.0069)	(.0023)	(.0006)	(.0046)
Validation Type	Worst-case	−0.0056***	0.0050***	−0.0002	0.0114***	−0.0228***	−0.0067**	−0.0018*
		(.0010)	(.0004)	(.0004)	(.0024)	(.0009)	(.0033)	(.0011)
	Average-case	0.0054***	0.0078***	0.0015***	0.0134***	0.0029***	0.0202***	0.0012
		(.0010)	(.0004)	(.0004)	(.0024)	(.0009)	(.0033)	(.0011)
Fixed Effect	Setting	Yes	No	No	No	No	No	No
	Domain	No	Yes	Yes	Yes	Yes	Yes	Yes
Overall	N	12527	3671	3671	287	3671	935	287
	Adjusted R^2	0.241	0.792	0.935	0.896	0.890	0.748	0.963

Notes. ***,** and * show statistical significance at the 1%, 5%, and 10% levels using two-tailed tests, respectively.

Table 24. Regression results on algorithmic design components (`Worst Domain`, incorporating nonlinear effects of the radius and additional interaction effects between the model class and shift patterns) on the performance gap, i.e., **target accuracy - source accuracy**.

Variable Name		Dependent variable: Performance gap						
		All	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6
Model Class	LGBM	−0.0134***	0.0123***	0.0193***	−0.0413***	−0.0302***	−0.0343***	0.0017
		(.0012)	(.0021)	(.0012)	(.0030)	(.0034)	(.0026)	(.0012)
	NN2	0.0056***	−0.0130***	0.0341***	−0.0063*	0.0252***	0.0169***	0.0020
		(.0015)	(.0027)	(.0020)	(.0035)	(.0058)	(.0033)	(.0019)
	NN3	−0.0153***	−0.0051*	0.0292***	−0.0647***	−0.0127**	0.0073**	0.0011
		(.0015)	(.0027)	(.0024)	(.0035)	(.0063)	(.0034)	(.0021)
	NN4	−0.0120***	−0.0066**	0.0107***	−0.0668***	−0.0313***	0.0097***	−0.0024
		(.0016)	(.0027)	(.0027)	(.0038)	(.0076)	(.0032)	(.0027)
	Kernel	−0.0126***	−0.0022	0.0086***	−0.0442***	−0.0392***	−0.0034	0.0040***
		(.0012)	(.0021)	(.0012)	(.0031)	(.0033)	(.0026)	(.0013)
Ambiguity Set	Wasserstein	−0.0112***	0.0162***	−0.0049	−0.0171***	−0.0391***	0.0016	−0.0124***
		(.0018)	(.0031)	(.0031)	(.0033)	(.0063)	(.0047)	(.0021)
	Chi-squared	−0.0036***	0.0054***	0.0046***	−0.0143***	−0.0189***	0.0025	−0.0020*
		(.0009)	(.0014)	(.0011)	(.0020)	(.0028)	(.0017)	(.0010)
	Kullback-Leibler	−0.0030***	−0.0101***	0.0068***	0.0003	−0.0074***	−0.0099***	0.0000
		(.0009)	(.0017)	(.0010)	(.0025)	(.0026)	(.0021)	(.0010)
	Total Variation	−0.0129***	0.0006	−0.0055***	−0.0099**	−0.0369***	−0.0212***	−0.0005
		(.0018)	(.0033)	(.0017)	(.0048)	(.0053)	(.0041)	(.0018)
	OT-Discrepancy	−0.0057***	0.0173***	−0.0042**	−0.0043	−0.0328***	−0.0039	−0.0017
		(.0017)	(.0030)	(.0017)	(.0045)	(.0048)	(.0039)	(.0017)
	Radius	0.0002	−0.0059***	0.0073***	−0.0060***	0.0240***	−0.0085***	0.0019**
		(.0009)	(.0017)	(.0009)	(.0022)	(.0025)	(.0020)	(.0009)
	Radius2	0.0000	0.0005**	−0.0009***	0.0005	−0.0021***	0.0002	−0.0003***
		(.0001)	(.0002)	(.0001)	(.0003)	(.0003)	(.0003)	(.0001)
	LGBM-Radius	−0.0008	0.0288***	0.0063***	0.0076***	−0.0241***	−0.0268***	−0.0055***
		(.0011)	(.0020)	(.0010)	(.0028)	(.0029)	(.0025)	(.0011)
	NN2-Radius	0.0125***	−0.0246***	−0.0074***	0.0111***	0.0165***	0.0060	0.0147***
		(.0011)	(.0047)	(.0011)	(.0022)	(.0034)	(.0047)	(.0020)
	NN3-Radius	0.0177***	−0.0313***	−0.0096***	0.0308***	0.0295***	0.0055	0.0116***
		(.0012)	(.0048)	(.0014)	(.0022)	(.0048)	(.0060)	(.0023)
	NN4-Radius	−0.0086**	−0.0131*	0.0207***	0.0074	0.0656***	0.0115	0.0799***
		(.0037)	(.0072)	(.0075)	(.0063)	(.0171)	(.0074)	(.0128)
	Kernel-Radius	0.0018***	0.0020*	0.0024***	−0.0013	−0.0002	0.0077***	0.0041***
		(.0006)	(.0012)	(.0006)	(.0017)	(.0017)	(.0014)	(.0006)
Fixed Effect	Setting	Yes	No	No	No	No	No	No
	Domain	No	No	No	No	No	No	No
Overall	N	23776	4321	3212	5056	3330	4500	3352
	Adjusted R^2	0.755	0.208	0.338	0.239	0.304	0.328	0.111

Notes. ***,** and * show statistical significance at the 1%, 5%, and 10% levels using two-tailed tests, respectively.

Nonlinear regression to identify critical components Due to the limited fitting power of the linear equation, we generalize the linear equation (4.1) in each setting to a nonlinear model for a better explanation and subsequent attribution results. More specifically, following the same notation as (4.1), we estimate the following equation via fitting a random forest regressor (default setup as `scikit-learn` with 10 estimators using 80% of all data points to avoid overfitting):

$$\text{Accuracy}_{i,j,s,t} = f(X_{i,j,s}, D_{i,j,s,t}, Z_{i,j}, V_{i,j}; \mu_i, \tau_j) + \tilde{\eta}_{i,j,s,t} \tag{D.1}$$

We do not incorporate interaction terms between algorithmic design since $f(\cdot)$ is a nonlinear function that implicitly takes them into account in its specification.

In particular, to quantify the contribution of each algorithmic design component, i.e., model class, validation procedure, and ambiguity-set configuration, we compute the average marginal performance gain over the baseline linear SVM where all design flags are set to zero across the default configuration set $\mathcal{C}$ of that baseline linear SVM. In particular, for each variable $v \in \mathcal{V} := \{X_{i,j,s}, D_{i,j,s,t}, Z_{i,j}, V_{i,j}\}$, for the regression fitted in a given setting, we compute its sensitivity (attribution) result $\mathcal{A}(v)$ averaged over $c \in \mathcal{C}$ as follows:

$$\mathcal{A}(v;\Delta) = \frac{1}{\Delta|\mathcal{C}|} \sum_{c\in\mathcal{C}} \left(f(X_{i,j,s}, D_{i,j,s,t}, Z_{i,j}, V_{i,j}; \mu_i, \tau_j)|_{c+\Delta\cdot e_v} - f(X_{i,j,s}, D_{i,j,s,t}, Z_{i,j}, V_{i,j}; \mu_i, \tau_j)|_c \right), \tag{D.2}$$

where e_v is the unit vector with 1 in the v-th coordinate. This metric corresponds to a form of local sensitivity analysis in [9]. We report the attribution results in Table 25 with $\Delta = 1$ (Specifically, we compute the sensitivity value of the "Radius" parameter as $\max_{\Delta\in\{0.2,0.4,\ldots,1\}} \mathcal{A}(v;\Delta)$ since it is a continuous variable). Note that the random forest regressor produces an R^2 exceeding 0.8 across all settings, implying its adequacy to explain algorithmic behavior. Despite that, our main results in fitting (4.1) remain valid, where the effect of the model class still dominates that of the ambiguity set design.

D.6 Training Details in Section 4.2

We detail how we compute the worst-case distribution of DRO methods under the Linear-SVM based on f-divergence and Wasserstein distances as follows.

- For f-divergence-DRO, recall the training samples $\{(x_i, y_i)\}_{i\in[n]}$, we denote the loss for each sample by $\ell_{tr}(i) = \ell_{tr}(\widehat{f}(x_i), y_i)$ and solve the optimization problem as follows:

$$\max_{p\in\Delta_n} \left\{ \sum_{i\in[n]} p_i \ell_{tr}(i), \text{ s.t. } \frac{1}{n} \sum_{i\in[n]} f(np_i) \le \epsilon \right\}$$

  where $\Delta_n = \{p \in \mathbb{R}^n_+ | \sum_{i\in[n]} p_i = 1\}$. This is a linear optimization problem over a convex

Table 25. Attribution results of random forest regression on algorithmic design components on the method performance compared with Linear ERM method configurations following (D.2) in the `Best Config` design, with boldfaced values denoting the largest value in each category.

Variable Name		All	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6
Model Class	LGBM	0.0061	0.0082	**0.0073**	-0.0153	-0.0099	0.0490	**0.0244**
	NN2	**0.0085**	0.0073	0.0045	-0.0065	-0.0044	0.0688	0.0078
	NN3	0.0032	0.0062	-0.0002	**-0.0039**	-0.0154	**0.0735**	0.0048
	NN4	0.0037	**0.0085**	-0.0023	-0.0133	-0.0150	0.0702	0.0018
	Kernel	-0.0140	-0.0012	-0.0072	-0.0209	-0.0527	0.0515	-0.0450
Ambiguity Set	Wasserstein	-0.0003	-0.0006	-0.0006	0.0006	-0.0010	0.0006	-0.0052
	Chi-square	0.0006	**0.0024**	0.0004	-0.0004	-0.0005	**0.0018**	0.0013
	Kullback-Leibler	0.0000	-0.0006	-0.0006	-0.0002	0.0006	-0.0008	-0.0003
	Total Variation	-0.0010	-0.0019	-0.0009	-0.0019	-0.0019	-0.0013	-0.0001
	OT-Discrepancy	-0.0005	-0.0001	-0.0006	-0.0002	-0.0003	-0.0000	-0.0083
	Radius	**0.0030**	0.0014	**0.0012**	**0.0060**	**0.0038**	0.0003	**0.0042**
Shift Pattern	$Y\|X$-ratio	**0.0155**	**0.0339**	**0.0294**	**-0.0199**	**-0.0155**	**0.0530**	**0.0141**
Validation Type	Worst-case	-0.0004	0.0007	-0.0001	0.0024	-0.0020	-0.0030	0.0004
	Average-case	**0.0012**	**0.0035**	**0.0006**	**0.0020**	**-0.0002**	**0.0024**	**0.0003**
Overall	Out-of-sample R^2	0.929	0.864	0.957	0.936	0.934	0.922	0.939

feasible set.

- For Wasserstein-DRO, we solve the optimization problem from Theorem 20 (ii) in [92] with $\gamma = \frac{1}{n}$. This is a linear optimization problem.

For each optimization problem, we solve it via the interior-point algorithm by calling the MOSEK solver to obtain the worst-case distributions.

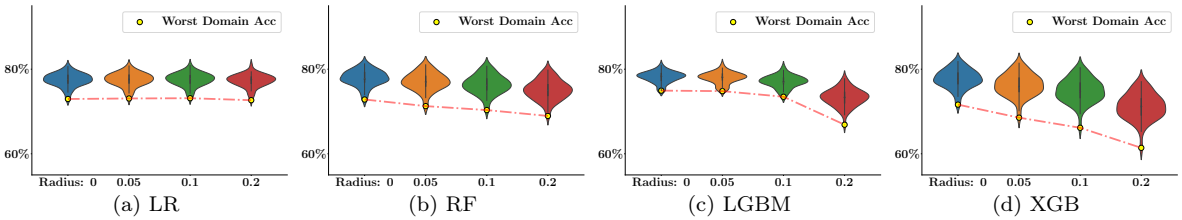
Then we demonstrate the experimental settings in Section 4.2, where we focus on Setting 1 with `ACS Income` dataset and choose CA as the source domain. We sample 20,000 data points from `ACS Income` (domain: CA), and obtain the worst-case empirical distribution of DRO, denoted by $\widehat{P}^\star$. For the other 50 target domains, we sample 20,000 points from each domain (denoted by $\{\widehat{Q}_t\}_{t\in[50]}$). Note that if the number of samples in one domain is less than 40,000, we randomly sample half of the samples from that domain.

For the optimal in-distribution accuracy for each model under the worst-case distribution $\widehat{P}^\star$ in (4.3), we calculate its performance via 4-fold validation. For the 50 target domains $\{\widehat{Q}_t\}_{t\in[50]}$, we calculate the optimal in-distribution accuracy for each model in the same way (via 4-fold validation). For the transfer accuracy in (4.4), we fit models (LR, RF, LGBM, XGB) on $\widehat{P}^\star$ and test them on $\{\widehat{Q}_t\}_{t\in[50]}$, respectively.

D.7 More Results in Section 4.2

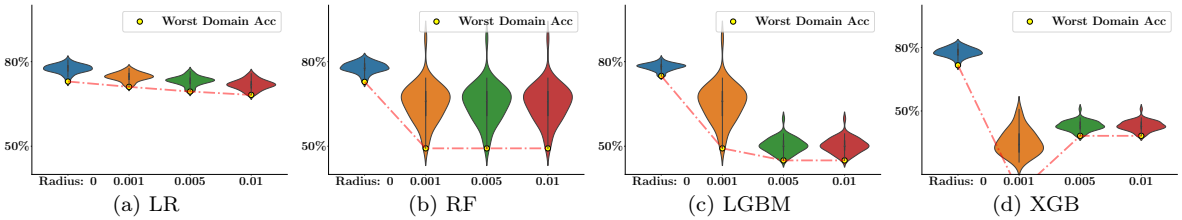
For the worst-case distribution analysis in Section 4.2, we provide results with more basic model classes in Figures 17, 18 and 19. Here in each figure, similar to Figure 6, the first bar represents the transfer accuracies from the training distribution $\widehat{P}$ to each of the 50 target domains ($\{\mathrm{TAcc}(\widehat{P}, \widehat{Q}_t)\}_{t\in[50]}$),

and the rest three bars represent the transfer accuracy from the worst-case distribution $\widehat{P}^\star$ to 50 target domains ($\{\mathrm{TAcc}(\widehat{P}^\star, \widehat{Q}_t)\}_{t\in[50]}$). The model classes $\mathcal{F}$ used here include LR, RF, LGBM, and XGB respectively.



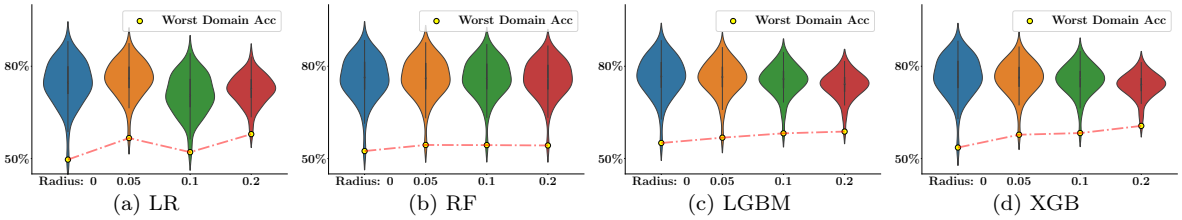
(a) LR (b) RF (c) LGBM (d) XGB

Figure 17. Transfer accuracy (4.4) on `ACS Income` (Setting 1) for different model classes $\mathcal{F}$ with respect to the worst-case distribution of KL-DRO.



(a) LR (b) RF (c) LGBM (d) XGB

Figure 18. Transfer accuracy (4.4) on `ACS Income` (Setting 1) for different model classes $\mathcal{F}$ with respect to the worst-case distribution of Wasserstein-DRO.



(a) LR (b) RF (c) LGBM (d) XGB

Figure 19. Transfer accuracy (4.4) on `ACS Pub.Cov` (Setting 4) for different model classes $\mathcal{F}$ with respect to the worst-case distribution of KL-DRO.

E Details in Section 5.1

E.1 Feature Attributions in Section 5.1

To quantify the shift in feature importance between the source and target domains, we compute Shapley value attributions for each feature using the `SHAP` framework. Specifically, we train two predictive models (each using a linear SVM) separately on the source data $(X_{\text{train}}, Y_{\text{train}})$ and the target data $(X_{\text{test}}, Y_{\text{test}})$, and use `shap.Explainer` to obtain SHAP values for each feature of each individual in both domains [71]. We then average the SHAP values over all instances within each domain to obtain per-feature mean attributions. Finally, we compute the differences between the

mean SHAP values in the source and target domains, resulting in a vector of feature-level attribution shifts. This vector highlights which features contribute differently to predictions across the two domains and thus helps identify critical components affected by distribution shifts.

E.2 Additional Results in other Settings besides Sections 5.1.1 and 5.1.2

In each setting, we select the top-K features, where $K \in \{1, 2, 3, 4, 5\}$ is chosen adaptively based on the number of features whose importance exceeds that of the remaining features by at least a threshold of 0.01.

Table 26: Index of Z selected in each setting

Setting	1	2	3	5	6
Shifted Z	{2}	{1,2}	{3,4}	{1,15,19,23,35}	{1,2,3,13,27}

Subsequently, we provide algorithmic intervention results that incorporate shift patterns in Table 26 for other settings, as shown in Tables 27 and 28. Across all five settings, the results remain valid. For settings where $Y|X$-shifts dominate, incorporating such $Y|X$-shift patterns leads to a more refined ambiguity set and (usually) improved model accuracy for the Marginal-DRO and Conditional-DRO methods, especially when there is substantial room for improvement in the base model class, i.e., higher accuracy and macro F1 scores for some tree-based ensemble method compared with SVM. For example, Marginal-DRO in Settings 1 and 5, and Conditional-DRO in Setting 1.

Table 27. Results (Accuracy) of 5 other pairs in Section 3.2, where we run each method with its top-10 configurations (according to the in-distribution accuracy) and report its mean accuracy and standard deviation. We boldface the best target performance within each class of methods in each setting.

Dataset / **Shift Pattern** / Source → Target Pair		Setting 1 $Y\|X$ dominates CA→PR	Setting 2 $Y\|X$ dominates MS→HI	Setting 3 $Y\|X$ dominates NYC→BOG	Setting 5 $Y\|X$ more CA→OR	Setting 6 X more 2010→2017
Basic Methods	SVM	72.8±0.0	71.2±0.1	**75.6**±0.1	**77.2**±0.1	68.7±0.1
	LR	73.5±0.5	70.7±0.5	74.9±0.2	77.0±0.2	68.5±0.2
	RF	**76.0**±0.8	**71.7**±0.4	73.2±0.2	67.9±0.3	**70.7**±0.4
	XGB	69.8±1.1	66.1±0.5	71.4±0.8	66.8±0.2	68.8±1.5
	LGBM	75.7±3.9	67.6±1.1	72.8±2.9	67.1±1.2	69.2±1.1
Linear-DRO Methods (base: SVM)	Wasserstein-DRO	74.2±2.4	**71.3**±0.1	75.3±0.1	**76.8**±0.3	67.8±0.0
	Wasserstein-DRO-Int	72.8±1.3	71.2±0.2	75.4±0.1	76.0±0.1	67.8±0.0
	Marginal-DRO	70.0±2.5	70.8±0.5	76.1±0.1	71.1±1.8	69.0±0.5
	Marginal-DRO-Int	75.1±0.4	69.9±0.6	**76.2**±0.6	**76.8**±1.7	**69.2**±0.4
	Conditional-DRO	73.2±0.2	70.8±0.2	76.0±0.1	76.0±0.2	69.0±0.0
	Conditional-DRO-Int	**75.7**±0.0	70.5±0.1	76.0±0.0	74.1±0.0	68.9±0.0

Table 28. Results (Macro F1 Score) of 5 other pairs in Section 3.2, where we run each method with its top-10 configurations (according to the in-distribution accuracy) and report its mean accuracy and standard deviation. We boldface the best target performance within each class of methods in each setting.

Dataset **Shift Pattern** Source → Target Pair		Setting 1 $Y\|X$ dominates CA→PR	Setting 2 $Y\|X$ dominates MS→HI	Setting 3 $Y\|X$ dominates NYC→BOG	Setting 5 $Y\|X$ more CA→OR	Setting 6 X more 2010→2017
Basic Methods	SVM	60.7±0.0	52.9±0.5	**75.6**±0.1	**60.9**±0.1	59.8±0.2
	LR	61.3±0.4	**53.8**±0.3	74.8±0.2	59.4±1.2	58.8±0.3
	RF	**62.0**±0.6	51.0±0.8	73.1±0.2	59.8±0.3	64.6±1.0
	XGB	57.9±1.0	52.5±0.5	71.3±0.8	60.4±0.4	64.7±0.9
	LGBM	62.4±2.7	53.2±0.4	72.6±3.0	59.7±1.2	**65.1**±0.7
Linear-DRO Methods (base: SVM)	Wasserstein-DRO	60.6±0.8	51.9±0.0	75.3±0.1	59.6±1.7	58.8±0.0
	Wasserstein-DRO-Int	60.7±0.9	51.9±0.1	75.3±0.1	61.7±0.1	58.8±0.0
	Marginal-DRO	60.9±0.5	51.7±0.4	71.2±1.9	62.6±0.5	58.6±1.7
	Marginal-DRO-Int	61.0±0.4	52.2±1.1	**76.1**±0.3	**62.8**±0.2	59.6±3.1
	Conditional-DRO	58.6±1.8	**53.2**±0.5	**76.1**±0.1	62.4±0.9	**62.2**±1.3
	Conditional-DRO-Int	**62.2**±0.3	52.7±1.1	73.0±0.6	61.1±0.3	61.6±1.6

E.3 Additional Results in Section 5.1.3

Our prompt to Gemini-2.5-Pro is as follows:

> Given the ACS Income dataset `https://fairlearn.org/main/user_guide/datasets/acs_income.html`, and the following set of one-hot encoded binary features: ... (mention feature names), each feature is a binary indicator, encoded via one-hot encoding. Suppose I train a predictive model using data only from California and aim to generalize its predictions for income to the other 50 states and Puerto Rico (PR). Which of the above features are most likely to exhibit significant distributional shifts between California and the other states/PR? Can you provide a list of all features as a dictionary, with keys being the feature and values being the weight?

Afterwards, we rescale the vector Δ_{LLM} by multiplying its components by a constant so that their sum equals the original feature dimensionality d.

F Details in Section 5.2

F.1 Further Discussions for Algorithm 1

In this part, we provide a more detailed introduction of our proposed Algorithm 1.

Choices of S_X. First, we construct a shared distribution S_X over X whose support is contained in that of both P_X and Q_X. We choose a specific *shared distribution* S_X over X whose support is

contained in that of P_X and Q_X (following [26]).

$$S_X \propto \frac{p_X(x)q_X(x)}{p_X(x)+q_X(x)} \tag{F.1}$$

Ideally, the chosen shared distribution would exhibit a higher density when both P_X and Q_X densities are high, and a lower density when either of the two possesses a low density. This strategy effectively allows regions of shared density to be more pronounced., which leads to the formulation of (F.1). Furthermore, we discuss different choices of S_X to provide more intuitions. As demonstrated in [26], we can use many forms of the shared distribution. For example, we could choose the following form:

$$s_X(x) \propto \min\{p_X(x), q_X(x)\}, \tag{F.2}$$

which guarantees that the support of S_X is contained in that of both P_X and Q_X. Another choice is:

$$s_X(x) \propto \begin{cases} p_X(x)+q_X(x) & \text{if } \min\left\{\frac{p_X(x)}{q_X(x)}, \frac{q_X(x)}{p_X(x)}\right\} \geq \epsilon, \\ 0 & \text{otherwise,} \end{cases}$$

for some $\epsilon \geq 0$. This form of S_X defines shared samples as those with high likelihood ratios. Notably, for all the three forms of S_X, if $P_X = Q_X$, then $S_X = P_X = Q_X$. And when $p_X(x) \gg q_X(x)$ or $p_X(x) \ll q_X(x)$, (F.1) and (F.2) become similar. In our Algorithm 1, we use the form of (F.1), and Cai et al. [26] observe that in practice the qualitative conclusions are not very sensitive to the specific choice of shared distribution.

Intuitions behind Algorithm 1. Since we do not have access to samples from the shared distribution S_X, we reweight samples from P_X and Q_X using the likelihood ratios:

$$\frac{s_X}{p_X}(x) \propto \frac{q_X(x)}{p_X(x)+q_X(x)} \text{ and } \frac{s_X}{q_X}(x) \propto \frac{p_X(x)}{p_X(x)+q_X(x)}. \tag{F.3}$$

Then we define $\hat{\alpha}$ as the proportion of the pooled data that comes from the distribution Q:

$$\hat{\alpha} = \frac{n_Q}{n_P+n_Q} \quad \text{and} \quad \hat{\pi}(x) = \mathbb{P}(\tilde{X} \text{ from } Q_X | \tilde{X} = x),$$

where $\hat{\pi}(x)$ denotes the probability of a sample to come from Q_X. Using Bayes' rule, we have:

$$\begin{aligned} \hat{\pi}(x) &= \frac{\mathbb{P}(\tilde{X}=x|\tilde{X} \text{ from } Q_X)\mathbb{P}(\tilde{X} \text{ from } Q_X)}{\mathbb{P}(x)} = \frac{\hat{\alpha}q(x)}{\hat{\alpha}q(x)+(1-\hat{\alpha})p(x)}, \\ &= \frac{\hat{\alpha}}{\hat{\alpha}+(1-\hat{\alpha})\frac{p(x)}{q(x)}}. \end{aligned}$$

Noting that the ratio $\hat{\pi}(x)$ can be modeled as the probability that an input x came from P_X vs Q_X, we train a binary "domain" classifier to estimate the ratios. (The "domain" classifier can be any black-box method, and we use XGBoost throughout).

Then the likelihood ratios that we care about could be reformulated as:

$$\frac{s_X}{p_X}(x) \propto \frac{1}{\frac{p_X(x)}{q_X(x)}+1} \text{ and } \frac{s_X}{q_X}(x) \propto \frac{\frac{p_X(x)}{q_X(x)}}{\frac{p_X(x)}{q_X(x)}+1},$$

which gives that:

$$\frac{s_X}{p_X}(x) \propto \frac{\hat{\pi}(x)}{(1-\hat{\alpha})\hat{\pi}(x)+\hat{\alpha}(1-\hat{\pi}(x))} \quad \text{and} \quad \frac{s_X}{q_X}(x) \propto \frac{1-\hat{\pi}(x)}{(1-\hat{\alpha})\hat{\pi}(x)+\hat{\alpha}(1-\hat{\pi}(x))}.$$

After obtaining the likelihood ratios $\frac{s_X}{p_X}(x)$ and $\frac{s_X}{q_X}(x)$, we could do an apples-to-apples comparison: we estimate $P_{Y|X}$ and $Q_{Y|X}$ over the shared distribution S_X (using XGBoost $\mathcal{F}$):

$$f_P := \arg\min_{f\in\mathcal{F}} \left\{ \mathbb{E}_{S_X}\left[\mathbb{E}_P[\ell_{tr}(f(X),Y)|X]\right] = \mathbb{E}_P\left[\ell_{tr}(f(X),Y)\frac{dS_X}{dP_X}(X)\right] \right\},$$
$$f_Q := \arg\min_{f\in\mathcal{F}} \left\{ \mathbb{E}_{S_X}\left[\mathbb{E}_Q[\ell_{tr}(f(X),Y)|X]\right] = \mathbb{E}_Q\left[\ell_{tr}(f(X),Y)\frac{dS_X}{dQ_X}(X)\right] \right\}.$$

Then, for any threshold $b \in [0,1]$, $\{x \in \mathcal{X} : |f_P(x) - f_Q(x)| \geq b\}$ suggests a region that may suffer model performance degradation due to $Y|X$-shifts.

F.2 Alternative Approach of Identifying Covariate Region

In Algorithm 1, we propose a simple way to identify the covariate region to explain the cause of $Y|X$-shifts. And the identified region could be used to guide the collecting process of target data, which could help to further reduce the effects of performance degradation. However, in practice, when the amount of target samples is quite small, it is difficult to train the f_Q only with target samples accurately.

Following this idea, we propose a sample-efficient alternative for identifying covariate regions in Algorithm 2, which does not need f_Q to fit $Q_{Y|X}$ on the target distribution. Since the training data is enough, it is feasible to fit f_P on the shared distribution S_X as:

$$f_P := \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_P\left[\frac{dS_X}{dP_X}\ell_{tr}(f(X),Y)\right]. \tag{F.4}$$

Then for n_Q samples from target distribution Q, we could use a prediction model $h(x)$ to approximate $|Y - f_P(X)|$ on the shared distribution (by reweighting density ratio $\frac{dS_X}{dQ_X}$). Note that this method does not need to train f_Q on target samples, but the quality of the density ratio dS_X/dP_X and the

Algorithm 2: Sample-Efficient Alternative for Identifying Regions

Input: Source samples $\{(x_i^P, y_i^P)\}_{i\in[n_P]} \overset{\text{i.i.d}}{\sim} P$ and target samples $\{(x_j^Q, y_j^Q)\}_{j\in[n_Q]} \overset{\text{i.i.d}}{\sim} Q$. Model discrepancy threshold b.

1 Estimate $\hat{\pi}(x) \approx \mathbb{P}(\tilde{X} \sim Q_x | \tilde{X} = x)$ by training a classifier on the source and target samples.

2 Calculate density ratios $w_\mu(\hat{\pi}(x), \hat{\alpha})$ according to Equation (5.6) and (5.7) for $\mu = P, Q$.

3 Fit the prediction model f_P according to Equation (F.4), where we set $\mathcal{F}$ as XGBoost; (*only fit f_P here*)

4 Fit a model $h(x)$ to predict $|f_P(x) - y|$ using *target* samples $\{(x_j^Q, y_j^Q)\}_{j\in[n_Q]}$ with the weight $\lambda_j^Q = \frac{w_Q(\hat{\pi}(x_j^Q),\hat{\alpha})}{\sum_{k\in[n_Q]} w_Q(\hat{\pi}(x_k^Q),\hat{\alpha})}, \forall j \in [n_Q]$.

Output: Region $\mathcal{R} = \{x \in \mathcal{X} : h(x) \geq b\}$.

prediction model $h(x)$ still depend on the target samples.

F.3 Other Results

F.3.1 Analysis of Decision Tree

In Algorithm 1, we use a shallow *decision tree* $h(x)$ to approximate $y = |f_P(x) - f_Q(x)|$ on the shared distribution S_X to find the covariate region with highest discrepancy. In our decision tree, we use the *squared error* as the splitting criterion. And below we demonstrate that this criterion is equivalent to maximizing the discrepancy between two children nodes.

Suppose there are N samples with outcomes $\{y_i\}_{i\in[N]}$ belonging to tree node fa, and these samples are split into two children nodes s_1, s_2, where the node s_1, s_2 denote the set of sample indices in the two children nodes respectively. The squared error criterion to split fa into s_1 and s_2 is:

$$\min_{s_1,s_2} \left\{ \mathcal{L}(s_1, s_2) := \frac{1}{N} \left(\sum_{i\in s_1} (y_i - \mu_{Y,1})^2 + \sum_{i\in s_2} (y_i - \mu_{Y,2})^2 \right) \right\}, \tag{F.5}$$

where

$$\mu_{Y,1} := \frac{\sum_{i=1}^N y_i \mathbf{1}_{\{i\in s_1\}}}{\sum_{i=1}^N \mathbf{1}_{\{i\in s_1\}}}, \quad \mu_{Y,2} := \frac{\sum_{i=1}^N y_i \mathbf{1}_{\{i\in s_2\}}}{\sum_{i=1}^N \mathbf{1}_{\{i\in s_2\}}} \tag{F.6}$$

denote the mean values of the outcome Y with samples in children nodes s_1 and s_2. Denote the distribution of the outcome Y follows the empirical distribution over the N samples $\{y_i\}_{i\in[N]}$. Simplifying (F.5), we have:

$$\mathcal{L}(s_1, s_2) = P(Y \in s_1)\text{Var}_{s_1}(Y) + P(Y \in s_2)\text{Var}_{s_2}(Y) = \mathbb{E}_S[\text{Var}(Y|S)], \tag{F.7}$$

where $\text{Var}_s(Y)$ denotes the variance of the outcome variable Y in node s, $S = \{s_1, s_2\}$ is the variable

representing the children nodes. Therefore, given that $\text{Var}_{fa}(Y)(:= \text{Var}_S(\mathbb{E}[Y|S]) + \mathbb{E}_S[\text{Var}(Y|S)])$ is constant, the minimal $\mathbb{E}_S[\text{Var}(Y|S)]$ corresponds with the largest $\text{Var}_S(\mathbb{E}[Y|S])$, which maximizes the discrepancy of the outcome between two children nodes.

F.3.2 Details of Data-centric Interventions

In Section 5.2.2, we propose two potential data-centric interventions to mitigate the performance degradation. In this section, we introduce in detail the intervention of collecting specific data from the target.

Experiment setup. We focus on the income prediction task using the `ACS Income` dataset. Consider a practical scenario where the training set consists of 20,000 samples from California (CA) and the trained model was deployed in Puerto Rico (PR) in trial. After the trial deployment in PR, we got a *small amount* of samples from PR with labels and observed performance degradation. Under this setting, we investigate the effect of non-algorithmic interventions.

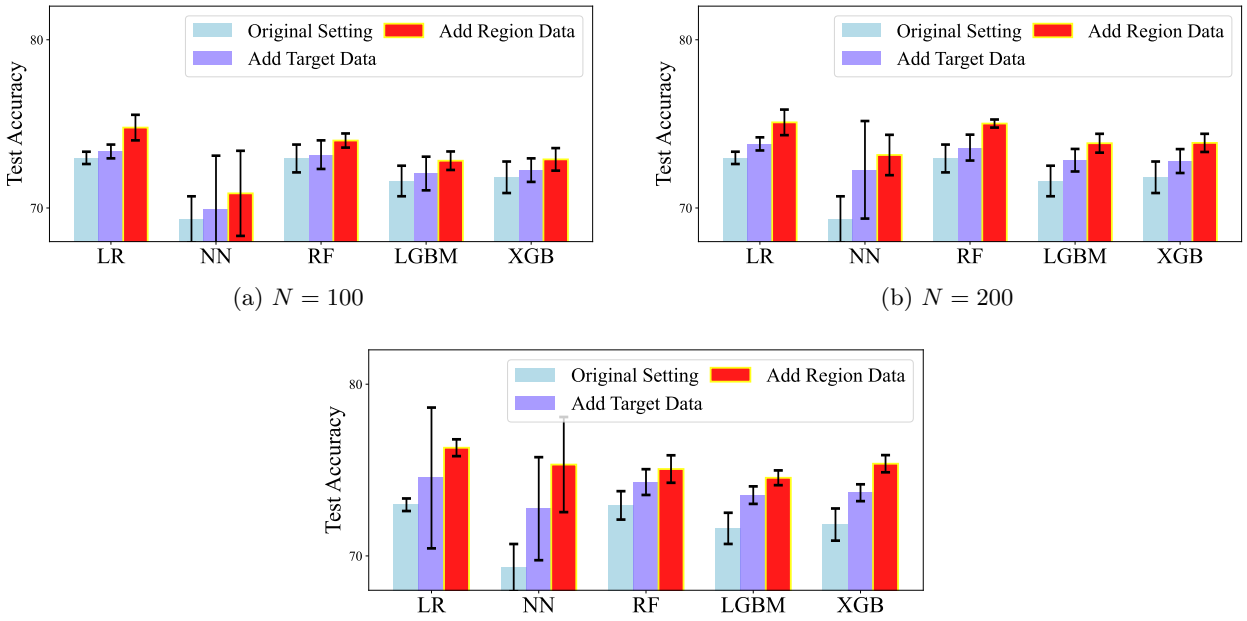
Collect specific data from the target. We first identify the regions with high discrepancy between source and target. Note that the sample size of the target state is small compared to the training samples. Then for basic ERM and tree-based ensemble methods like LR, NN, RF, LightGBM and XGB, we compare the performances of:

- original setting (only 20,000 samples from the source);
- original setting with N additional random samples drawn from the whole target state;
- original setting with N additional random samples drawn from the *covariate region* of the target domains.

In this experiment, we first select the best configuration of each method according to the $i.i.d$ validation set in the original setting (only samples from CA), and fix it for the other two interventions. We vary N as 100, 200, 300 and the results are shown in Figure 20. From Figure 20, incorporating data from the identified covariate region leads to a *stable improvement* on typical algorithms even for small target sample sizes. However, we observe that LightGBM and XGBoost would easily overfit f_Q on the target data, and we use random forest under this setup as an alternative. It is worth investigating approaches to find covariate regions effectively under small/imbalanced sample sizes in the future. The approach mentioned here is a simple way of non-algorithmic and explainable interventions and we hope it could inspire further research in this direction.

References

[1] Assistments. https://www.assistments.org/. Accessed: July 2025.



(a) $N = 100$

(b) $N = 200$

(c) $N = 300$

Figure 20: Test accuracies of different ways to incorporate data.

[2] College scorecard. https://collegescorecard.ed.gov/. Accessed: July 2025.

[3] Diabetes. https://www.cdc.gov/brfss/index.html. Accessed: July 2025.

[4] US taxi dataset. https://www.kaggle.com/competitions/nyc-taxi-trip-duration/data, . Accessed: June 2024.

[5] Other taxi dataset. https://www.kaggle.com/datasets/mnavas/taxi-routes-for-mexico-city-and-quito, . Accessed: June 2024.

[6] R. Adragna, E. Creager, D. Madras, and R. Zemel. Fairness and robustness in invariant learning: A case study in toxicity classification. *arXiv preprint arXiv:2011.06485*, 2020.

[7] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[8] J. D. Angrist and J.-S. Pischke. *Mostly harmless econometrics: An empiricist's companion*. Princeton university press, 2009.

[9] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831, 2010.

[10] H. Bastani. Predicting with proxies: Transfer learning in high dimension. *Management Science*, 67(5):2964–2984, 2021.

[11] G. Bayraksan and D. K. Love. Data-driven stochastic programming using phi-divergences. In *Tutorials in Operations Research*, pages 1–19. INFORMS, 2015.

[12] R. Belbasi, A. Selvi, and W. Wiesemann. It's all in the mix: Wasserstein classification and regression with mixed features. *Manufacturing & Service Operations Management*, 2026.

[13] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. T. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang. AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM J. Res. Dev.*, 63(4/5):4:1–4:15, 2019.

[14] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2): 341–357, 2013.

[15] A. Bennouna and B. Van Parys. Holistic robust data-driven decisions. *arXiv preprint arXiv:2207.09560*, 2022.

[16] A. Bennouna, R. Lucas, and B. Van Parys. Certified robust neural networks: Generalization and corruption resistance. In *International Conference on Machine Learning*, pages 2092–2112. PMLR, 2023.

[17] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, and K. Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep.*, 2020.

[18] B. Bischl, G. Casalicchio, M. Feurer, P. Gijsbers, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, and J. Vanschoren. Openml benchmarking suites. In *Neural Information Processing Systems Datasets and Benchmarks Track*, volume 35, 2021.

[19] J. Blanchet, Y. Kang, and K. Murthy. Robust Wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

[20] J. Blanchet, Y. Kang, K. Murthy, and F. Zhang. Data-driven optimal transport cost selection for distributionally robust optimization. In *Winter Simulation Conference*, pages 3740–3751. IEEE, 2019.

[21] J. Blanchet, K. Murthy, and V. A. Nguyen. Statistical analysis of wasserstein distributionally robust estimators. In *Tutorials in Operations Research: Emerging optimization methods and modeling techniques with applications*, pages 227–254. INFORMS, 2021.

[22] J. Blanchet, D. Kuhn, J. Li, and B. Taskesen. Unifying distributionally robust optimization via optimal transport theory. *arXiv preprint arXiv:2308.05414*, 2023.

[23] A. Blum and K. Stangl. Recovering from biased data: Can fairness constraints improve accuracy? In *Symposium on Foundations of Responsible Computing (FORC)*, volume 1, 2020.

[24] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[25] K. Budhathoki, D. Janzing, P. Bloebaum, and H. Ng. Why did the distribution change? In *International Conference on Artificial Intelligence and Statistics*, pages 1666–1674. PMLR, 2021.

[26] T. Cai, H. Namkoong, and S. Yadlowsky. Diagnosing model performance under distribution shift. *Operations Research*, 74(2):898–916, 2026.

[27] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye. Joint transfer and batch-mode active learning. In *International Conference on Machine Learning*, pages 253–261. PMLR, 2013.

[28] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *ACM SIGKDD International Conference on Knowledge Discovery*, pages 785–794. ACM, 2016.

[29] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[30] X. Dastile, T. Celik, and M. Potsane. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, 91:106263, 2020.

[31] F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 6478–6490, 2021.

[32] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[33] J. Duchi and H. Namkoong. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(1):2450–2504, 2019.

[34] J. Duchi, T. Hashimoto, and H. Namkoong. Distributionally robust losses for latent covariate mixtures. *Operations Research*, 71(2):649–664, 2023.

[35] J. C. Duchi and H. Namkoong. Learning models with uniform performance via distributionally robust optimization. *Annals of Statistics*, 49(3):1378–1406, 2021.

[36] J. C. Duchi, P. W. Glynn, and H. Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 46(3):946–969, 2021.

[37] P. M. Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[38] M. Fatima and M. Pasha. Survey of machine learning algorithms for disease diagnostic. *Journal of Intelligent Learning Systems and Applications*, 9(01):1–16, 2017.

[39] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

[40] R. Gao, X. Chen, and A. J. Kleywegt. Wasserstein distributionally robust optimization and variation regularization. *Operations Research*, 72(3):1177–1191, 2024.

[41] J. Gardner, Z. Popovic, and L. Schmidt. Subgroup robustness grows on trees: An empirical baseline investigation. In *Advances in Neural Information Processing Systems*, volume 35, pages 9939–9954, 2022.

[42] A. Ghorbani, J. Zou, and A. Esteva. Data shapley valuation for efficient batch active learning. In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pages 1456–1462. IEEE, 2022.

[43] J.-y. Gotoh, M. J. Kim, and A. E. Lim. Calibration of distributionally robust empirical optimization models. *Operations Research*, 69(5):1630–1650, 2021.

[44] L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, pages 507–520, 2022.

[45] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

[46] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[47] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 3315–3323, 2016.

[48] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

[49] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[50] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

[51] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

[52] Z. Hu and L. J. Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 2013.

[53] B. Y. Idrissi, M. Arjovsky, M. Pezeshki, and D. Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022.

[54] G. Iyengar, H. Lam, and T. Wang. Hedging complexity in generalization via a parametric distributionally robust optimization framework. *arXiv preprint arXiv:2212.01518*, 2022.

[55] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos. Towards efficient data valuation based on the shapley value. In *International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019.

[56] R. Jiang and Y. Guan. Risk-averse two-stage stochastic program with distributional ambiguity. *Operations Research*, 66(5):1390–1405, 2018.

[57] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[58] N. Justin, S. Aghaei, A. Gómez, and P. Vayanos. Learning optimal classification trees robust to distribution shifts. *arXiv preprint arXiv:2310.17772*, 2023.

[59] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pages 3149–3157, 2017.

[60] F. H. Knight. *Risk, uncertainty and profit*, volume 31. Houghton Mifflin, 1921.

[61] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. Earnshaw, I. Haque, S. M. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[62] D. Kuhn, S. Shafiee, and W. Wiesemann. Distributionally robust optimization. *Acta Numerica*, 34:579–804, 2025.

[63] S. Kulinski and D. I. Inouye. Towards explaining distribution shifts. In *International Conference on Machine Learning*, volume 202, pages 17931–17952. PMLR, 2023.

[64] J. Lee and M. Raginsky. Minimax statistical learning with Wasserstein distances. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[65] D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford. Large-scale methods for distributionally robust optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 8847–8860, 2020.

[66] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[67] J. Liu, T. Wang, P. Cui, and H. Namkoong. On the need for a language describing distribution shifts: Illustrations on tabular datasets. In *Advances in Neural Information Processing Systems*, pages 51371–51408, 2023.

[68] Z. Liu, H. Ding, H. Zhong, W. Li, J. Dai, and C. He. Influence selection for active learning. In *International Conference on Computer Vision*, pages 9254–9263. IEEE, 2021.

[69] D. Z. Long, M. Sim, and M. Zhou. Robust satisficing. *Operations Research*, 71(1):61–82, 2023.

[70] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

[71] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4768–4777, 2017.

[72] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230, 2013.

[73] J. Miller, K. Krauth, B. Recht, and L. Schmidt. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning*, pages 6905–6916. PMLR, 2020.

[74] J. P. Miller, R. Taori, A. Raghunathan, S. Sagawa, P. W. Koh, V. Shankar, P. Liang, Y. Carmon, and L. Schmidt. 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, 2021.

[75] S. Moosavi, M. H. Samavatian, S. Parthasarathy, and R. Ramnath. A countrywide traffic accident dataset. *arXiv preprint arXiv:1906.05409*, 2019.

[76] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath. Accident risk prediction based on heterogeneous sparse data: New dataset and insights. In *International Conference on Advances in Geographic Information Systems*, pages 33–42. ACM, 2019.

[77] N. Mu and J. Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

[78] A. Natekin and A. Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[79] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[80] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[81] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. MIT Press, 2008.

[82] H. Rahimian and S. Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[83] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 workshop on active learning for natural language processing*, pages 27–32, 2010.

[84] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[85] R. T. Rockafellar, S. Uryasev, et al. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[86] 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. In *International Conference on Learning Representations*, 2020.

[87] R. Sahoo, L. Lei, and S. Wager. Learning from a biased sample. *arXiv preprint arXiv:2209.01754*, 2022.

[88] S. Santurkar, D. Tsipras, and A. Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations*, 2021.

[89] J. C. Schlimmer and R. H. Granger. Incremental learning from noisy data. *Machine learning*, 1:317–354, 1986.

[90] B. Settles. Active learning literature survey. 2009.

[91] S. Shafieezadeh-Abadeh, P. M. Esfahani, and D. Kuhn. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, volume 29, pages 1576–1584, 2015.

[92] S. Shafieezadeh-Abadeh, D. Kuhn, and P. M. Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.

[93] S. Shafieezadeh-Abadeh, L. Aolaritei, F. Dörfler, and D. Kuhn. New perspectives on regularization and computation in optimal transport-based distributionally robust optimization. *arXiv preprint arXiv:2303.03900*, 2023.

[94] A. Shapiro, E. Zhou, and Y. Lin. Bayesian distributionally robust optimization. *SIAM Journal on Optimization*, 33(2):1279–1304, 2023.

[95] R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[96] H. Singh, F. Xia, A. Subbaswamy, A. Gossmann, and J. Feng. A hierarchical decomposition for explaining ml performance discrepancies. In *Advances in Neural Information Processing Systems*, volume 38, pages 128516–128555, 2024.

[97] A. Sinha, H. Namkoong, and J. C. Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[98] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 739–748, 2020.

[99] B. Taskesen, V. A. Nguyen, D. Kuhn, and J. Blanchet. A distributionally robust approach to fair classification. *arXiv preprint arXiv:2007.09530*, 2020.

[100] D. Ulmer, L. Meijerink, and G. Cinà. Trust issues: Uncertainty estimation does not enable reliable ood detection on medical tabular data. In *Machine Learning for Health*, pages 341–354. PMLR, 2020.

[101] J. Wang, R. Gao, and Y. Xie. Sinkhorn distributionally robust optimization. *Operations Research*, 2025.

[102] B. Xie, L. Yuan, S. Li, C. H. Liu, and X. Cheng. Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 8058–8068. IEEE, 2022.

[103] S. Yan, K. Chaudhuri, and T. Javidi. Active learning with logged data. In *International Conference on Machine Learning*, pages 5521–5530. PMLR, 2018.

[104] Y. Yang, H. Zhang, D. Katabi, and M. Ghassemi. Change is hard: A closer look at subpopulation shift. In *International Conference on Machine Learning*, volume 202, pages 39584–39622. PMLR, 2023.

[105] H. Yao, C. Choi, B. Cao, Y. Lee, P. W. Koh, and C. Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *Advances in Neural Information Processing Systems*, volume 35, pages 10309–10324, 2022.

[106] H. Yu, P. Cui, Y. He, Z. Shen, Y. Lin, R. Xu, and X. Zhang. Stable learning via sparse variable independence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10998–11006, 2023.

[107] Y. Zeng, J. Liu, H. Lam, and H. Namkoong. LLM embeddings improve test-time adaptation to tabular $y|x$-shifts. *arXiv preprint arXiv:2410.07395*, 2024.

[108] R. Zhai, C. Dan, Z. Kolter, and P. Ravikumar. DORO: Distributional and outlier robust optimization. In *International Conference on Machine Learning*, pages 12345–12355. PMLR, 2021.

[109] H. Zhang, H. Singh, M. Ghassemi, and S. Joshi. "Why did the model fail?": Attributing model performance changes to distribution shifts. In *International Conference on Machine Learning*, volume 202, pages 41550–41578. PMLR, 2023.

[110] X. Zhang, Y. He, T. Wang, J. Qi, H. Yu, Z. Wang, J. Peng, R. Xu, Z. Shen, Y. Niu, et al. Nico challenge: Out-of-distribution generalization for image recognition challenges. In *Computer Vision–ECCV 2022 Workshops*, pages 433–450. Springer, 2023.

[111] E. Zhao, A. Liu, A. Anandkumar, and Y. Yue. Active learning under label shift. In *International Conference on Artificial Intelligence and Statistics*, pages 3412–3420. PMLR, 2021.

[112] J.-J. Zhu, W. Jitkrittum, M. Diehl, and B. Schölkopf. Kernel distributionally robust optimization: Generalized duality theorem and stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 280–288. PMLR, 2021.