Exploring Group Distributionally Robust Optimization in Machine Learning: A Stochastic Optimization Perspective

Jiayu Wang

1 Introduction

Machine learning models are typically trained to minimize the empirical average loss on a training set to maximize accuracy on an independent and identically distributed (*i.i.d.*) test set. However, the generalization capability of such models can be significantly undermined by the distribution shifts where the test distribution deviates from the training distribution. In this project, I am specifically interested in distribution shifts caused by class priors or group priors from training to test sets. Such shifts are usually caused by spurious correlations where Empirical Risk Minimization (ERM) may easily lead models to overfit to those spurious features. For example, the background can act as a spurious feature in classifying birds with different backgrounds, *e.g.,* waterbirds are predominantly observed in water background, similarly landbirds commonly appear in land background. To avoid overfitting to spurious correlations and therefore introduce high losses, Sagawa et al. [1] proposed instead to train models to minimize worst-case losses over groups in the training set which is an instance of Distributionally Robust Optimization (DRO). However, considering the potential over-conservatism of DRO, this project further explores and extends upon the group DRO framework by utilizing prior knowledge.

2 Preliminaries

In this work, we consider multi-class classification where the goal is to predict the class $y \in \mathcal{Y} := \{1, 2, \ldots, K\}$ of inputs $x \in \mathcal{X}$. Given a model family with parameters Θ, loss function $\ell : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}_+$, and a training set $\{(x_i, y_i)\}_{i=1}^N$ drawn *i.i.d.* from some underlying distribution P, the goal is to find optimal parameters that make the model generalize well, especially when the training data contains spurious features. As a long-standing challenge in machine learning, a variety of methods and formulations have been proposed in the literature. In this work, we primarily focus on DRO, and especially the Group DRO formulation.

Distributionally Robust Optimization (DRO) In DRO, the goal is instead to minimize the worst-case expected loss over an uncertainty set of distributions $\mathcal{G}$:

$$\min_{\theta \in \Theta} \left\{ \mathcal{R}(\theta) := \sup_{G \in \mathcal{G}} \mathbb{E}_{(x,y)\sim G}[\ell(y, f(x, \theta))] \right\}. \tag{1}$$

The uncertainty set $\mathcal{G}$ includes possible test distributions that one wants models to perform well on. Choosing a general family $\mathcal{G}$, such as a divergence ball around the training distribution, confers robustness to a wide set of distribution shifts, but can lead to overly conservative models if optimized over worst-case distribution [2].

Group DRO (GDRO) To construct a realistic set of distributions without being overly conservative, one can leverage the prior knowledge of spurious correlations: we assume each instance (x, y) has some attributes $a \in \mathcal{A}$ that correlate to its label. These instances can be categorized into $M = |A| \times |Y|$ groups which is the Cartesian product of $|A|$ attributes and $|Y|$ classes. The data distribution P is perceived as a mixture of M groups $\{P_m\}_{m=1}^M$. The uncertainty set $\mathcal{G} := \left\{ \sum_{m=1}^M g_m P_m : g \in \Delta_M \right\}$. This choice of $\mathcal{G}$ allows us to learn a model robust to prior shifts. Since linear programming achieves its optima at vertices, the worst-case risk $\mathcal{R}(\theta)$ (Eq. 1) is equivalent to taking the maximum over the risk of each group:

$$\mathcal{R}(\theta) = \max_{m \in \mathcal{M}} \mathbb{E}_{(x,y)\sim P_m}[\ell(y, f(x, \theta))] \tag{2}$$

Empirically, this is formulated as minimizing the empirical worst-group risk $\hat{\mathcal{R}}(\theta)$:

$$\hat{\theta}_{\text{GDRO}} := \arg\min_{\theta \in \Theta} \left\{ \hat{\mathcal{R}}(\theta) := \max_{m \in \mathcal{M}} \mathbb{E}_{(x,y)\sim \hat{P}_m}[\ell(y, f(x, \theta))] \right\}, \tag{3}$$

where $\hat{P}_m$ denotes the empirical distribution of training data $\{(x_i, y_i, m_i)\}_{i=1}^{N_m}$.
Remarks: The above objective minimizes the risk of the worst group in the training set. Note that this does not guarantee minimizing the risk of the worst group in the test set. The discrepancy (generalization gap) is denoted by $\Delta := \mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)$.

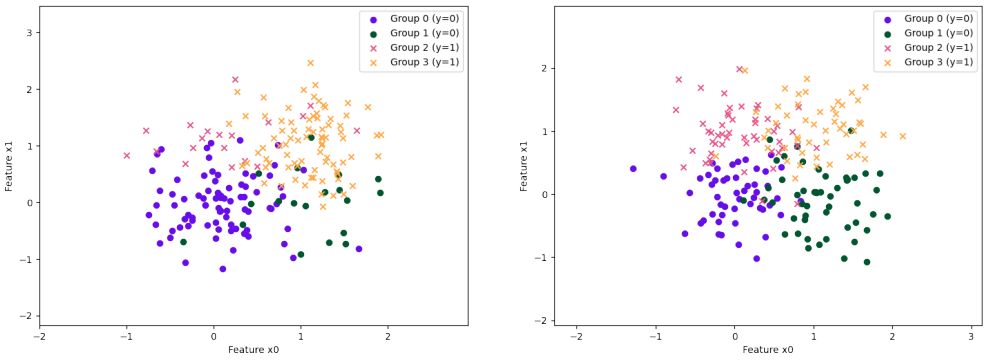
3 Problem Setup

In this section, we first introduce how we construct datasets with spurious correlations, and then describe the model and performance evaluation metrics.

Dataset To explore the spurious correlation between feature attributes and labels, we leverage Gaussian Mixture Models (GMM). For better visualization, we consider $\mathcal{X} \in \mathbb{R}^2$ and $\mathcal{Y} \in \{0, 1\}$. We use the first feature x_0 as the spurious attribute $a \in \mathcal{A} := \{0, 1\}$ which exhibits a spurious correlation with the label y: $\rho := P(a = 0|y = 0) = P(a = 1|y = 1)$ quantifies the correlation between the attribute and the label. For test sets, we set $\rho = 0.5$. A demonstration of the training and test set is shown in Fig. 1

We consider the class-balanced setting ($P(y = 0) = P(y = 1) = 0.5$). The dataset is constructed by sampling from four groups, each represented by a Gaussian distribution with a group-specific mean vector and a shared covariance matrix. The mean vector of

group m is $\mu_m \in \{(0,0)^\top, (1,0)^\top, (0,1)^\top, (1,1)^\top\}$. The covariance matrix is a scaled identity matrix $\sigma^2 \mathbf{I}_2$ for some σ. Based on the above specifications, the joint probability $P(a=0, y=0) = P(a=1, y=1) = 0.5\rho$. Therefore, the number of samples for group 0 ($a=0, y=0$) and group 3 ($a=1, y=1$) is $0.5\rho N$, while group 1 ($a=1, y=0$) and group 2 ($a=0, y=1$) each has $0.5(1-\rho)N$ samples.



(a) Training Set ($\rho = 0.8$) (b) Test Set ($\rho = 0.5$)

Fig. 1: Demonstration of generated GMM datasets. **Left**: Training set. 80% of points labeled 0 has attribute $a = 0$, 80% of points labeled 1 has attribute $a = 1$. **Right**: Test set. Unbiased samples without spurious correlation.

Model We consider logistic regression for binary classification. Our goal is to learn a classifier $h(x) := \text{argmax}_{j \in \mathcal{Y}} f_j(x)$ based on the model $f : \mathcal{X} \to \Delta_M$. In the case where f is a linear function, it takes the form $\mathrm{f}[x, \theta] = \theta^\top x$. Given training samples $\{(x_i, y_i)\}_{i=1}^N$, the maximum likelihood principle yields the following negative log-likelihood loss $l(\theta) := \sum_{i=1}^N -(1-y_i)\log[1-\sigma[\mathrm{f}[x_i,\theta]]] - y_i \log[\sigma[\mathrm{f}[x_i,\theta]]]$, where $\sigma[z] = \frac{1}{1+\exp[-z]}$.

Evaluation Metrics We consider both average accuracy and worst group accuracy. (1) The average accuracy represents the overall performance of our classification model on the test set. It is calculated by assessing whether each test point is correctly classified. For each test instance, a value of 1 is assigned if it is accurately classified, and 0 if it is not. The average accuracy is then obtained by summing these values and dividing by the total number of test instances. (2) The worst group accuracy focuses on the model's performance with respect to the least represented groups in the training set. Specifically, it calculates the accuracy only for the group that has the smallest number of instances in the training data. This metric is crucial for understanding how well the model performs on potentially underrepresented or more challenging subsets of the data, ensuring that the model's performance is not just evaluated on the most prevalent or easiest-to-classify cases.

4 Method

We consider four methods to solve the above problem: 1) solve ERM by stochastic gradient descent (SGD) with momentum; 2) solve GDRO with an online optimization algorithm proposed in [1]; 3) solve GDRO with an extensive form; 4) solve GDRO with Benders Decomposition. We implement 1), 2), and 4) for empirical comparison.

4.1 ERM with Stochastic Gradient Descent

As one of the widely adopted and classic principles, empirical risk minimization (ERM) minimizes the expected loss of the empirical distribution:

$$\hat{\theta}_{\text{ERM}} := \arg\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\hat{P}}[\ell(y, f(x,\theta))], \tag{4}$$

where $\hat{P}$ is the empirical distribution over the training data. Despite the simplicity, Gulrajani and Lopez-Paz [3] demonstrated that with extensive hyperparameter tuning, ERM demonstrates good generalization performance across a wide range of real-world datasets compared to methods tailored to tackle spurious correlation. As ERM does not require any prior information on spurious correlation, we consider it as a baseline and solve by mini-batch stochastic gradient descent (SGD).

4.2 GDRO with Online Optimization

We also consider solving the problem with GDRO formulation (Eq. 3) using an online optimization algorithm first proposed in [1]:

Algorithm 1 Online optimization algorithm for group DRO

Require: Step sizes η_g, η_θ; P_m for each $m \in \mathcal{M}$

Initialize $\theta^{(0)}$ and $g^{(0)}$
for $t = 1, \ldots, T$ **do**
  $m \sim \text{Uniform}(1, \ldots, m)$ ▷ Choose a group m at random
  $x, y \sim P_m$ ▷ Sample x, y from group m
  $g' \leftarrow g^{(t-1)}; g'_m \leftarrow g'_m \exp\left(\eta_g \ell\left(\theta^{(t-1)}; (x,y)\right)\right)$ ▷ Update weights for group m
  $g^{(t)} \leftarrow g' / \sum_{m'} g'_{m'}$ ▷ Renormalize g
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta g_m^{(t)} \nabla \ell\left(\theta^{(t-1)}; (x,y)\right)$ ▷ Use g to update θ
end for

4.3 An Extensive Form for GDRO

In Section 2, we outlined a general framework for GDRO. Here, we delineate a more specialized context for the uncertainty set $\mathcal{G}$. For clarity, we remap the label space $\mathcal{Y} \in \{0, 1\}$ to $\{-1, 1\}$. Under this mapping, our logistic regression model is expressed as $P(y = \pm 1 \mid x, \theta) = \sigma\left(y\theta^\top x\right) = \frac{1}{1+\exp(-y\theta^\top x)}$. Consequently, the loss function is

reformulated as $l(\theta) = \frac{1}{N}\sum_{i=1}^{N} -\log(\sigma(y_i\theta^\top x_i))$, where $y_i \in \{-1, 1\}, x_i \in \mathbb{R}^d$, and $\theta \in \mathbb{R}^d$. Assume that we are given a reference prior distribution r (could be either from domain expert or the uniform empirical distribution) that we want our model to perform well on, the GDRO problem is then defined as:

$$\min_{\theta\in\Theta} \max_{g\in G(\delta)} \sum_{m\in[M]} g_m \mathbb{E}_{(x,y)\sim P_m}[\ell(y, f(x,\theta))] \tag{5}$$

where $G(\delta) = \left\{ g \in \mathbb{R}_+^M : \sum_{m=1}^M g_m = 1, D(g, r) \leq \delta \right\}$ is defined as the set of weights g constrained to lie within the δ-radius ball (defined by the divergence $D : \Delta_M \times \Delta_M \to \mathcal{R}$) around the target distribution r. Incorporating ϕ-divergence defined as $\phi(t) = |1-t|$ and adopting a uniform prior $r = [\frac{1}{M}, \cdots, \frac{1}{M}]$, the distance constraint $D(g, r) \leqslant \delta$ can be simplified as $\sum_{i=1}^M \phi(Mg_m) \leqslant \delta$. To solve the min-max problem in its extensive form, we initially address the inner maximization problem for a given θ:

$$\begin{aligned}
\max_g \sum_{m\in[M]} g_m \mathbb{E}_{(x,y)\sim P_g}[\ell(y, f(x,\theta))] = \max_{g,z} & -\sum_{m=1}^{M} \frac{g_m}{N_m} \sum_{i=1}^{N_m} \log(\sigma(y_i^m \theta^\top x_i^m)) \\
\text{s.t.} \quad & \sum_{m=1}^{m} g_m = 1 \\
& z_m \geqslant M g_m - 1, \quad \forall m \in [M] \\
& z_m \geqslant 1 - M g_m, \quad \forall m \in [M] \\
& \sum_{m=1}^{m} z_m \leq M\delta, \\
& z_m, g_m \geqslant 0 \quad \forall m \in [M]
\end{aligned} \tag{6}$$

Taking the dual of Eq. 6 with $\beta, \lambda_m^+, \lambda_m^-, \gamma$ as the dual variables for the 4 constraints above, we have:

$$\begin{aligned}
\min_{\beta,\lambda_m^+,\lambda_m^-,r} \quad & \beta + \sum_{m=1}^{M} \lambda_m^+ - \sum_{m=1}^{M} \lambda_m^- + M\delta\gamma \\
\text{s.t.} \quad & \beta + M\lambda_m^+ - M\lambda_m^- \geqslant -\frac{1}{N_m} \sum_{i=1}^{N_m} \log \sigma(y_i^m \theta^\top x_i^m), \quad \forall m \in [M] \\
& -\lambda_m^+ - \lambda_m^- + \gamma \geqslant 0, \quad \forall m \in [M] \\
& \lambda_m^+, \lambda_m^- \geqslant 0, \quad \forall m \in [M] \\
& \gamma \geqslant 0
\end{aligned} \tag{7}$$

We consider adding an ℓ_2 norm constraint on θ: $||\theta||_2 \leqslant C$. Since G is non-empty and bounded, by relatively complete recourse, the dual problem's feasible region is non-empty. By strong duality, the primal and dual problems have the same optimal

solution. The reformulated GDRO problem is thus expressed as:

$$\begin{aligned}
\min_{\theta,\beta,\lambda_m^+,\lambda_m^-,r} \quad & \beta + \sum_{m=1}^{M} \lambda_m^+ - \sum_{m=1}^{M} \lambda_m^- + M\delta\gamma \\
\text{s.t.} \quad & \beta + M\lambda_m^+ - M\lambda_m^- \geqslant -\frac{1}{N_m} \sum_{i=1}^{N_m} \log \sigma(y_i^m \theta^\top x_i^m), \quad \forall m \in [M] \\
& -\lambda_m^+ - \lambda_m^- + r \geqslant 0, \quad \forall m \in [M] \\
& ||\theta||_2 \leqslant C \\
& \lambda_m^+, \lambda_m^- \geqslant 0, \quad \forall m \in [M] \\
& \gamma \geqslant 0
\end{aligned} \tag{8}$$

Remarks: As a constrained optimization problem with nonlinearity (*e.g.,* log and $\sigma(\cdot)$), Gurobi cannot be used to solve it. Therefore, we do not implement this formulation and leave it as future work.

4.4 GDRO with Benders Decomposition

Alternatively, we can directly solve the primal problem instead of finding the dual by leveraging Benders Decomposition we learned in class.

Master Problem At iteration t, we first solve the master problem and obtain an optimal solution $(\hat{\alpha}^t, \hat{\theta}^t)$ which serves as a candidate lower bound for the objective value:

$$\begin{aligned}
\min_{\alpha,\theta} \quad & \alpha \\
\text{s.t.} \quad & \alpha \geqslant -\sum_{m=1}^{M} \frac{g_m}{N_m} \sum_{i=1}^{N_m} \log(\sigma(y_i^m \theta^\top x_i^m)), \quad \forall g \in \hat{V}^t \\
& ||\theta||_2 \leqslant C
\end{aligned} \tag{9}$$

where $\hat{V}^t \subseteq V(G)$, $V(G)$ is the finite set of vertices of feasible region G.

Subproblem After obtaining $(\hat{\alpha}^t, \hat{\theta}^t)$ at iteration t, we then solve the second-stage subproblem:

$$\begin{aligned}
Q(\hat{\theta}^t) = \max_{g,z} \quad & -\sum_{m=1}^{M} \frac{g_m}{N_m} \sum_{i=1}^{N_m} \log(\sigma(y_i^m \hat{\theta}^{t\top} x_i^m)) \\
\text{s.t.} \quad & \sum_{m=1}^{m} g_m = 1 \\
& z_m \geqslant M g_m - 1, \quad \forall m \in [M] \\
& z_m \geqslant 1 - M g_m, \quad \forall m \in [M] \\
& \sum_{m=1}^{M} z_m \leq M\delta, \\
& z_m, g_m \geqslant 0 \quad \forall m \in [M]
\end{aligned} \tag{10}$$

Solving the above problem yields a solution $(\hat{g}^t, \hat{z}^t)$ and a candidate upper bound for the objective value. A violated constraint is identified if $\hat{\alpha}^t \leqslant -\sum_{m=1}^{M} \frac{g_m^t}{N_m} \sum_{i=1}^{N_m} \log(\sigma(y_i^m \hat{\theta}^{t\top} x_i^m))$, which is then appended to the master problem. However, as Gurobi cannot handle non-linear and non-quadratic variable transformations, we consider a linear approximation for $q_i(\theta) = -\log(\sigma(y_i^m \theta^\top x_i^m))$ by $q_i(\theta) \leqslant q_i(\hat{\theta}) + \nabla q_i(\hat{\theta})^\top (\theta - \hat{\theta})$, where $\nabla q_i(\hat{\theta}) = -\left(1 - \sigma\left(y_i^m \hat{\theta}^{t\top} x_i^m\right)\right) \cdot y_i^m x_i^m$.

If we find a violated cut in the subproblem with $(\hat{\theta}^t)$, we augment $\hat{V}$ with $\{\hat{g}^t\}$ and add a new constraint $\alpha \geqslant \sum_{m=1}^{M} \frac{\hat{g_m}^t}{N_m} \sum_{i=1}^{N_m} [q_i(\hat{\theta}^t) + \nabla q_i(\hat{\theta}^t)^\top (\theta - \hat{\theta}^t)]$ to the master problem. The iterative process is continued until the optimality gap (*i.e.,* the difference between the upper bound obtained from the subproblem and the lower bound from the master problem) converges within a pre-established tolerance.

5 Experiments and Discussions

Due to Gurobi's limitations in handling nonlinear programs, our analysis concentrates on comparing between ERM (Sec. 4.1), GDRO with Online Optimization (Sec. 4.2) and GDRO with Benders Decomposition (Sec. 4.4).

5.1 Experiment Setup

Hardware and software The Benders Decomposition is implemented using the Gurobi Python API, whereas ERM and the Online Optimization algorithm are implemented via PyTorch 3.11.5. All experiments are run on an 8-core Macbook with M1 processor.

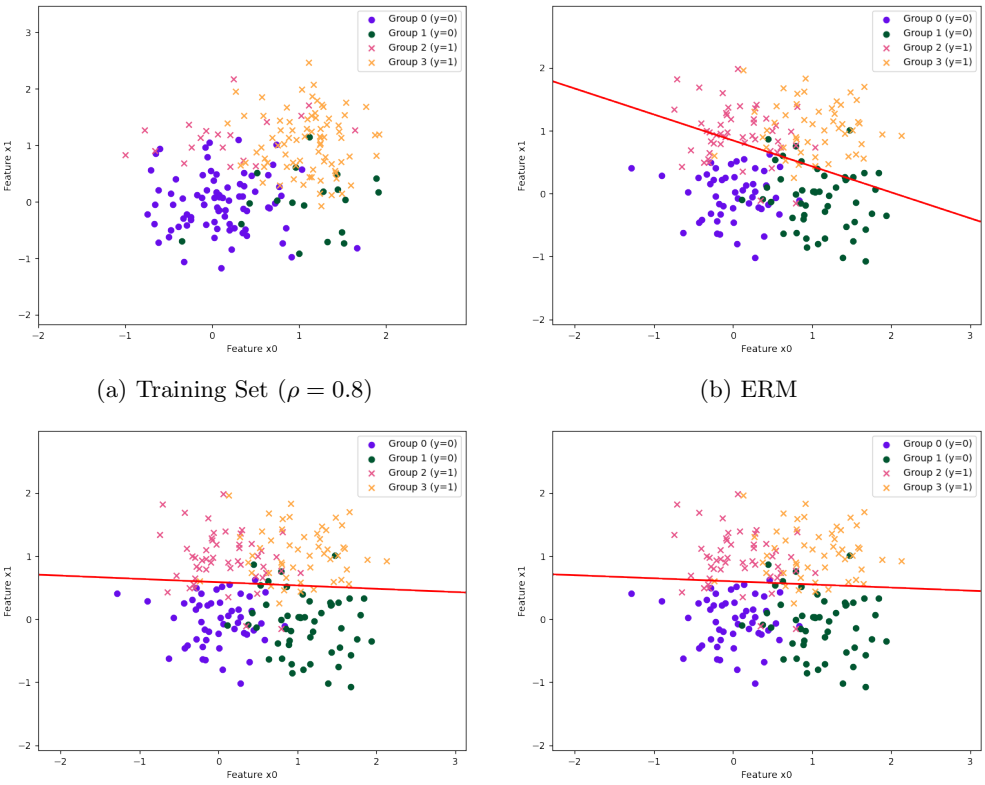
Datasets As detailed in Section 3, GMM datasets are generated and employed in our experiments. For ablation studies, by default, both training and testing sets comprise 200 samples ($N = 200$), each with an identity covariance matrix scale ($\sigma^2 = 0.2$). The training set is configured with a degree of spurious correlation ($\rho = 0.8$), whereas the test set remains unbiased. Consequently, the least represented groups by default are groups 1 and 2, for which we calculate the worst group accuracy.

Methods We provide additional details for the three methods described in Section 4:

- **ERM**: For the initial learning rate lr, we perform a grid search over $lr \in \{0.0001, 0.001, 0.01, 0.1, 0.4, 1, 2\}$ and set the default learning rate lr to 0.4, with a batch size of 64, a momentum of 0.9, and weight decay of $1e-4$. The model is trained for 30 epochs via mini-batch stochastic gradient descent (SGD) with momentum.
- **OnlineOpt**: The default learning rate is set to 0.2, with a step size of 0.01, a batch size of 16, a momentum of 0.9, a weight decay of $1e-4$, and training is conducted for 300 steps. Our implementation follows Algorithm 1.
- **Benders**: By default, we set $\delta = 1e-4$, $C = 50$, and the convergence tolerance to $1e-4$.

5.2 Main Results and Discussions

Figure 2 illustrates the decision boundaries (on the test set) of each method when trained on a dataset with $\rho = 0.8$. Recall that x_0 is the spurious feature and x_1 is the invariant feature. Intuitively, a horizontal decision boundary indicates a superior classifier as it mitigates the impact of spurious correlation. Therefore, we can see that both OnlineOpt and Benders are more effective than ERM.



(a) Training Set ($\rho = 0.8$)

(b) ERM

(c) OnlineOpt

(d) Benders

Fig. 2: Visualization of decision boundaries. We consider binary classification with four groups. Figure (a) illustrates the training set with a degree of spurious correlation $\rho = 0.8$ where three methods are trained on. Figure (b) to (d) display the same unbiased test set and the respective decision boundaries for ERM, OnlineOpt, and Benders. The red line represents the decision boundary where the model predicts $y = 1$ above the line and $y = 0$ otherwise. As x_0 is the spurious feature and x_1 is the invariant feature, a horizontal line indicates a desirable classifier while a vertical line indicates reliance on spurious correlations in the training set.

Method	$\rho = 0.5$ (unbiased)		$\rho = 0.6$		$\rho = 0.7$		$\rho = 0.8$		$\rho = 0.9$	
	avg acc	worst acc	avg acc	worst acc	avg acc	worst acc	avg acc	worst acc	avg acc	worst acc
ERM	0.895	0.890	0.905	0.880	0.890	0.790	0.865	0.760	0.860	0.750
OnlineOpt	0.905	0.850	0.905	0.880	0.905	0.860	0.905	0.830	0.885	0.900
Benders	0.910	0.870	0.910	0.880	0.910	0.870	0.905	0.880	0.890	0.900

Table 1: Comparison of three methods in binary classification by varying degrees of spurious correlation ρ from 0.5 (no correlation) to 0.9 (strong correlation). We report average and worst-group accuracies on the same test set. The highest accuracies are marked in yellow.

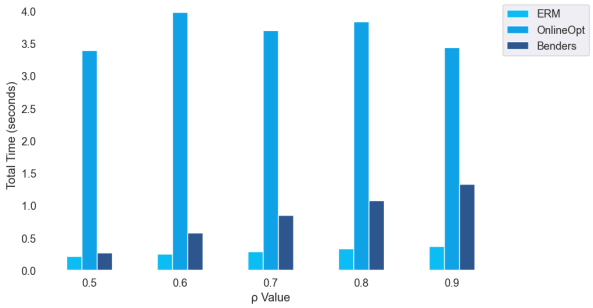


Fig. 3: Total time (seconds) of three methods under different degrees of spurious correlation ρ.

Table 1 shows the average accuracy and worst-group accuracy for three methods on binary classification under different degrees of spurious correlation ρ. We describe the main findings below:

OnlineOpt and Benders excel under stronger spurious correlations. Across scenarios with varying ρ, both OnlineOpt and Benders generally outperform or match ERM in terms of average and worst-group accuracy, except in cases with minimal prior shifts from the training to the test distribution. Notably, the performance gap between ERM and the other two methods widens as ρ increases, indicating a decrease in ERM's robustness under stronger spurious correlations.

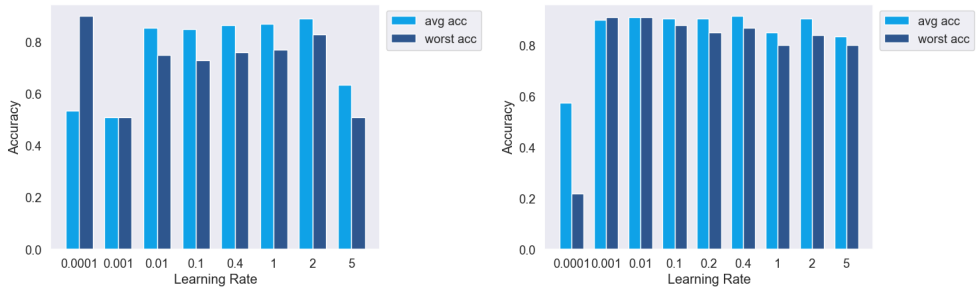
ERM is competitive under weak spurious correlations. In scenarios where this is no or slight spurious correlation (*e.g.*, $\rho = 0.5$ and 0.6), ERM maintains strong performance, achieving the highest worst-group accuracy among the three methods.

Benders marginally surpasses OnlineOpt across all settings. OnlineOpt and Benders demonstrate similar performance in terms of average and worst-group accuracies. However, Benders exhibits a slight advantage over OnlineOpt consistently in all the tasks evaluated.

OnlineOpt incurs the highest time complexity. We calculate the total training time for methods under various ρ levels. The results are shown in Fig. 3. We can see that ERM is the most computationally efficient one, consistently outperforming OnlineOpt and Benders across various ρ levels. OnlineOpt incurs the highest computational duration, which suggests its substantial complexity. Benders is more

efficient than OnlineOpt but still lags behind ERM, particularly when spurious correlation intensifies. This highlights the trade-off between computational efficiency and robustness against varying degrees of spurious correlations for these methods.

5.3 Ablation Studies



(a) Accuracies of ERM for different learning rates

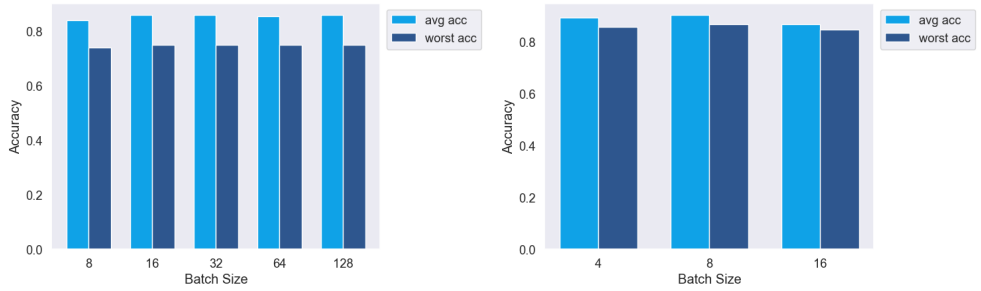
(b) Accuracies of OnlineOpt for different learning rates

Fig. 4: Accuracy comparison of ERM vs OnlineOpt for different learning rates. OnlineOpt outperforms ERM across different learning rates in general. ERM is more sensitive to learning rate than Online Optimization.

Sensitivity of ERM and OnlineOpt to learning rate. Fig. 4 illustrates the influence of learning rate adjustments on the performance of ERM and OnlineOpt. We can see that 1) ERM is sensitive to changes in the learning rate, exhibiting notable variations in both average and worst group accuracies across different rates. 2) In contrast, OnlineOpt demonstrates considerably less sensitivity to learning rate adjustments, with the exception of a very small learning rate (0.0001). This suggests an advantage of OnlineOpt, as it does not require as meticulous a calibration of the learning rate compared to ERM.

ERM and OnlineOpt are stable across various batch sizes. As shown in Figure 5, ERM and OnlineOpt demonstrate consistent performance across diverse batch sizes. Specifically, ERM maintains steady average and worst-group accuracies across a wide range of batch sizes from 8 to 128. In contrast, OnlineOpt shows optimal performance with smaller batch sizes (4 and 8), but its effectiveness slightly diminishes with a larger batch size (16). Notably, the maximum batch size for OnlineOpt is inherently limited by the number of samples in the smallest group, which can constrain its applicability in datasets with significant group size disparities. This highlights ERM's broader adaptability to different batch sizes compared to OnlineOpt and underscores the importance of careful batch size selection, especially for methods sensitive to intra-dataset group size variations like OnlineOpt.

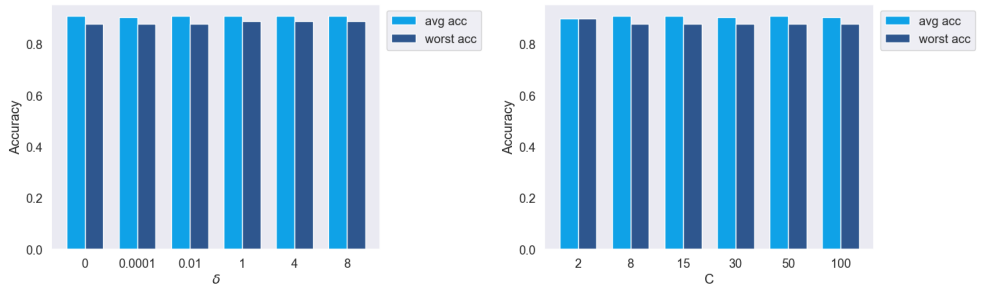
Benders' robustness to parameters δ and C. As evidenced in Fig. 6, we can see that Benders decomposition is relatively robust to variations in key parameters



(a) Accuracies of ERM for different batch sizes

(b) Accuracies of OnlineOpt for different batch sizes

Fig. 5: Comparative analysis of ERM and OnlineOpt across various batch sizes. Both methods exhibit robustness to batch size variations. ERM demonstrates a more versatile range in batch size adaptability, in contrast to OnlineOpt constrained by the size of the smallest group.



(a) Accuracies of Benders for different δ

(b) Accuracies of Benders for different C

Fig. 6: Benders' performance with respect to variations in δ and C, demonstrating its stability and adaptability.

such as δ and C. Such robustness highlights the advantage of Benders for addressing spurious correlation. It is important to note, however, that Benders' applicability is contingent upon convexity for the first stage variable (θ). In scenarios lacking convexity, OnlineOpt may provide a viable alternative, especially adept at handling non-convex models.

Ablation on the degree of spurious correlation ρ. Table 2 illustrates the performance of Benders across varying degrees of spurious correlation (ρ). We observe that the total computation time increases with ρ, indicating a higher complexity at higher correlation levels. The number of iterations required for convergence and the number of cuts added also show variability with different ρ values.

Ablation on the training set size N. Table 3 demonstrates the impact of different training set sizes (N) on the model's performance. As N increases, there is a

ρ	Total Time (s)	# Cuts Added	# Iters	Opt ObjVal
0.5	0.272	36	37	0.304
0.6	0.583	41	42	0.240
0.7	0.852	37	38	0.246
0.8	1.080	32	33	0.292
0.9	1.336	36	37	0.238

Table 2: Computational details of Benders with varying degrees of spurious correlation ρ.

N	Avg Acc	Worst Acc	Total Time (s)	# Cuts Added	# Iters	Opt ObjVal
100	0.890	0.810	0.175	42	43	0.086
200	0.905	0.880	0.388	32	33	0.292
500	0.900	0.870	0.987	37	38	0.279
1k	0.905	0.880	2.231	39	40	0.282
2k	0.910	0.900	4.053	29	30	0.337
5k	0.910	0.890	8.806	30	31	0.305
10k	0.900	0.870	17.595	28	29	0.310

Table 3: Performance and computational details of Benders with varying training set size N.

notable increase in total computation time, reflecting the added complexity of larger datasets. The number of iterations and cuts added appears to stabilize with larger datasets. Notably, both average and worst-case accuracies improve with larger training sets, reaching a peak at $N = 2000$. This suggests the computation-performance trade-off of Benders.

6 Conclusion

In this work, we explore binary classification problem under spurious correlations (prior shifts from the training to the test distribution). We discuss the formulation of four potential solutions and empirically examine the effectiveness of three methods: ERM with SGD, GDRO with Online Optimization, and GDRO with Benders Decomposition. Our findings reveal that both Online Optimization and Benders Decomposition excel under strong spurious correlations, with Benders Decomposition slightly outperforming Online Optimization. However, ERM still demonstrates effective performance under weak spurious correlation. In terms of computational costs, Online Optimization is significantly more demanding compared to other methods. We also observe that both ERM and Online Optimization are sensitive to learning rate adjustments. Benders Decomposition, on the other hand, is insensitive w.r.t. key parameters (δ and C). Consequently, when convexity conditions are satisfied, Benders Decomposition offers a more stable and time-efficient solution than Online Optimization and ERM under strong spurious correlation.

References

[1] Sagawa, S., Koh, P.W., Hashimoto, T.B., Liang, P.: Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv preprint arXiv:1911.08731 (2019)

[2] Duchi, J.C., Hashimoto, T., Namkoong, H.: Distributionally robust losses against mixture covariate shifts. Under review **2**(1) (2019)

[3] Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. In: International Conference on Learning Representations (2021)