MODELING THE SECOND PLAYER IN DISTRIBUTIONALLY ROBUST OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributionally robust optimization (DRO) provides a framework for training machine learning models that are able to perform well on a collection of related data distributions (the "uncertainty set"). This is done by solving a min-max game: the model is trained to minimize its maximum expected loss among all distributions in the uncertainty set. While careful design of the uncertainty set is critical to the success of the DRO procedure, previous work has been limited to relatively simple alternatives that keep the min-max optimization problem exactly tractable, such as f-divergence balls. In this paper, we argue instead for the use of neural generative models to characterize the worst-case distribution, allowing for more flexible and problem-specific selection of the uncertainty set. However, while simple conceptually, this approach poses a number of implementation and optimization challenges. To circumvent these issues, we propose a relaxation of the KL-constrained inner maximization objective that makes the DRO problem more amenable to gradient-based optimization of large scale generative models, and develop model selection heuristics to guide hyper-parameter search. On both toy settings and realistic NLP tasks, we find that the proposed approach yields models that are more robust than comparable baselines.

1 INTRODUCTION

Machine learning models trained with empirical risk minimization (ERM) are able to achieve high aggregate performance on data sampled from their training distribution. However, they often exhibit drops in accuracy when confronted with data from domains that are under-represented in their training data, such as those of different topic (Gururangan et al., 2020), sociolect (Blodgett et al., 2016), accent (Amodei et al., 2016) or writer age (Hovy & Søgaard, 2015) in language processing tasks, or skin color (Grother et al., 2019) or lighting (Georghiades et al., 2001) in image processing tasks. This is a particularly egregious issue in applications where higher error rates can have far reaching negative implications, such as the silencing of underrepresented minorities in toxicity detection systems (Dixon et al., 2018) or disparity amplifying feedback loops in credit rating models (Fuster et al., 2018).

This behaviour often arises from the objective function of ERM, where the parameters θ of the model are learned by minimizing the expectation of a loss function ℓ under a data distribution p (or, specifically in practice, an associated empirical data distribution $\hat{p}$)

$$\mathcal{L}_{\text{ERM}}(\theta) = \mathbb{E}_{(x,y)\sim\hat{p}}\ell(x, y, \theta). \tag{1}$$

When the model encounters data sampled from a different distribution $q_{\text{test}} \neq p$, performance can suffer significantly. Distributionally robust optimization (DRO) (Ben-Tal et al., 2013b) provides a natural solution to this issue by replacing the expected risk under a single distribution p with the *worst* expected risk over a pre-determined family of distributions $\mathcal{Q}$ (the "uncertainty set")

$$\mathcal{L}_{\text{DRO}}(\theta) = \sup_{q\in\mathcal{Q}} \mathbb{E}_{(x,y)\sim q}\, \ell(x, y, \theta). \tag{2}$$

If $\mathcal{Q}$ contains q_{test}, the DRO objective upper bounds the expected risk under q_{test}. However, *a priori* knowledge of possible test distributions is not always available or easy to acquire. For example, training a model to be robust to some demographic attributes ($\mathcal{Q} = \{q_{\text{demographic 1}}, q_{\text{demographic 2}}, \dots\}$) requires collecting and annotating data with the necessary information, an expensive and ethically

fraught endeavour. In the absence of such information, one has to resort to defining the uncertainty set analytically, drawing on one's intuition of what constitutes a possible test distribution given the observed training distribution, such as using moment constraints (Delage & Ye, 2010; Nguyen et al., 2020), f-divergence (Ben-Tal et al., 2013a; Hu & Hong, 2013; Faury et al., 2020), Wasserstein/IPM (Sinha et al., 2018; Husain, 2020) balls, or coarse-grained mixture models (Oren et al., 2019; Hu et al., 2018). However, the need for keeping the inner supremum in Eq. (2) tractable limits the possible choices.

In this paper, we propose that the uncertainty set be instead defined as a family of parametric generative models. The resulting DRO objective (§2) is a differentiable game with two players: the original model $\ell(x, y; \theta)$ and a model of its worst-case distribution $q_\psi(x, y)$, the titular "second player" which we hereafter refer to as *the adversary*. Using this formulation — which we call Parametric DRO (P-DRO) — allows for more flexibility in the choice of the adversary's architecture (and so the uncertainty set). Unfortunately, finding a solution of this game via direct application of simultaneous gradient descent (Singh et al., 2000) is difficult (Balduzzi et al., 2018). In particular, direct gradient descent on the uncertainty set suffers from instability due to the large variance of the gradients (Greensmith et al., 2004), and hyper-parameter selection is not straightforward.

To address these challenges, we make two main contributions (§3): first, we propose a new relaxation of the DRO game's inner maximization problem (with KL constraints). The resulting objective is more amenable to simultaneous gradient update than the original zero-sum game and significantly improves training stability, while still yielding useful adversaries. Second, we develop a principled approach for selecting hyper-parameters: we leverage the learned adversaries to decide which of any two given models trained with P-DRO is more robust than the other.

We do an in-depth set of experiments analyzing the effect of our proposed changes on both a toy task as well as a more realistic, yet still synthetic sentiment classification task (§4). Finally, we show that in the more realistic setting of toxicity detection, P-DRO yields models that are more robust to changes in demographic groups, even these groups are unknown at training time, opening up applications in combatting dataset bias (§5).

2 Parameterizing the Uncertainty Set

Consider a model parameterized by $\theta \in \mathbb{R}^{d_{\text{model}}}$. Minimizing the DRO objective described in Eq. (2) over the uncertainty set $\mathcal{Q}$ turns the optimization problem into the min-max (or zero-sum) game

$$\min_{\theta \in \mathbb{R}^d} \sup_{q \in \mathcal{Q}} \mathbb{E}_{(x,y)\sim q}\, \ell(x, y, \theta). \tag{3}$$

The first player controls the parameters θ, whilst the second player controls the worst-case distribution q. In the absence of explicit information on groups of interest (such as demographics, domain, etc.), an adequate choice of the uncertainty set $\mathcal{Q}$ is critical to the success of DRO. This is in fact very much an active area of research (Sinha et al. (2018); Duchi & Namkoong (2018); Oren et al. (2019), see Rahimian & Mehrotra (2019) for a survey). $\mathcal{Q}$ must be sufficiently large to contain test distributions of interest. But if it is too large, it may contain "adversarial" distributions on which no model can perform well. Moreover, the design of $\mathcal{Q}$ is also circumscribed by the necessity of keeping the min-max problem tractable, particularly in the context of stochastic optimization. In Hu & Hong (2013) and Duchi et al. (2016) for example, the choice of f-divergence balls allows the use of duality arguments to reformulate (3) as a more manageable min-min problem. Others, like Hu et al. (2018) or Oren et al. (2019), propose using mixture models, the simplicity of which enables them to solve the inner maximization problem efficiently.

Instead, we propose to explicitly model the second player in the DRO game as a parametric model q_ψ of the data. Of course, not all parameterizations $\psi \in \mathbb{R}^{d_{\text{adv}}}$ of a given generative model represent useful distributions, and we require that the adversary stay "close" to the underlying true data distribution p. As a measure of distance between q_ψ and p, we choose the KL (Kullback & Leibler, 1951) divergence due to its wide acceptance in the machine learning community, as well as its appealing properties in the context of DRO.[1] The KL upper bound, κ, is left as a parameter to be decided by

[1]For instance: $\text{KL}(q\|p) < +\infty$ implies that q stays within the support of p

the experimenter. We refer to the resulting DRO formulation as Parametric DRO

$$\min_{\theta} \max_{\substack{\psi \\ \mathrm{KL}(q_\psi \| p) \leq \kappa}} \underbrace{\mathbb{E}_{(x,y)\sim q_\psi} \ell(x, y, \theta)}_{\mathcal{L}_{\text{P-DRO}}(\theta,\psi)}. \tag{4}$$

3 Optimizing P-DRO

The min-max problem in Eq. (4) belongs to a class of games called "differentiable games" (another famous representative being generative adversarial networks (Goodfellow et al., 2014)). We can search for a solution of this game with simultaneous gradient descent (Singh et al., 2000), *i.e.* by simultaneously updating θ and ψ with $-\nabla_\theta \mathcal{L}_{\text{P-DRO}}$ and $\nabla_\psi \mathcal{L}_{\text{P-DRO}}$ respectively. Unfortunately, in general, there is no theoretical guarantee that simultaneous gradient descent will converge to a Nash equilibrium[2] (Balduzzi et al., 2018), nor that any such equilibrium even exists if the objective is non-convex in θ (or non-concave in ψ). The success of GANs and the follow-up literature (Wang et al., 2019) serves as an encouraging example that gradient based methods can yield useful solutions. In this section, we discuss difficulties that arise when optimizing θ and ψ jointly, and propose modifications of the objective to address them.

3.1 Training the Model θ

We could train the model θ to take negative gradient steps on $\mathbb{E}_{(x,y)\sim q_\psi} \ell(x, y; \theta)$. This gradient can be estimated by sampling examples from q_ψ and averaging the gradient of their losses. Unfortunately, this objective requires that q_ψ is well-behaved at all iterations, as it is the only source of supervision for θ. If q_ψ is initialized incorrectly or begins producing unrealistic (x, y), the quality of θ degrades as it begins to learn a predictor on invalid training examples from q_ψ. As an alternative, we opt to compute the gradients for θ with importance sampling, *i.e.* rewriting $\mathcal{L}_{\text{P-DRO}}$ as $\mathbb{E}_{(x,y)\sim p} \frac{q_\psi(x,y)}{p(x,y)} \ell(x, y; \theta)$, which ensures that all (x, y) samples will be derived from the training set itself. Unfortunately, the true density p is unknown to us. As an approximation, we replace $\frac{q_\psi(x,y)}{p(x,y)}$ with the likelihood ratio between q_ψ and the maximum likelihood estimate of p, $q_{\psi_0} \coloneqq \arg\max_{q_\psi} \mathbb{E}_{(x,y)\sim p} \log q_\psi(x, y)$. This changes the min-max problem to

$$\min_{\theta} \max_{\substack{\psi \\ \mathrm{KL}(q_\psi \| p) \leq \kappa}} \mathbb{E}_{(x,y)\sim p} \frac{q_\psi(x, y)}{q_{\psi_0}(x, y)} \ell(x, y, \theta). \tag{5}$$

This becomes a simple expected loss objective, which we can estimate by sampling from the empirical distribution $\hat{p}$. In experiments, we find that with this formulation we are able to train robust θ even when q_ψ is only a mediocre generative model (see Appendix C.2). To further stabilize training at the beginning of the optimization process, we initialize ψ with ψ_0, making the objective exactly the same as ERM for the first gradient step.

3.2 Training the Adversary ψ

According to Eq. (5) the adversary ψ must maximize $E_{(x,y)\sim q_\psi} \frac{p(x,y)}{q_{\psi_0}(x,y)} \ell(x, y, \theta)$ within a KL ball of fixed radius. This is challenging for several reasons: first, enforcing the bound is intractable for complex families of adversaries where *e.g.* projecting onto the KL ball is another difficult optimization problem of its own. Second, maximizing the expectation with respect to the parameters of the distribution q_ψ is prone to instability due to large gradient variance (Greensmith et al., 2004).

Lagrangian Relaxation To address the first difficulty, we loosen the strict KL constraint and instead maximize the Lagrangian relaxation $\mathbb{L}$

$$\mathbb{L}(\psi, \tau) = \mathbb{E}_{(x,y)\sim q_\psi} \frac{p(x, y)}{q_{\psi_0}(x, y)} \ell(x, y, \theta) - \tau \left(\mathrm{KL}(q_\psi \| p) - \kappa \right). \tag{6}$$

[2]Nash equilibria (Osborne & Rubinstein, 1994) can be thought of the game theoretic analog of global minima in optimization.

We fix the Lagrangian multiplier $\tau > 0$ as treat it as a "temperature" hyper-parameter. With some reorganization (which we develop in Appendix A.1), we can show that

$$\mathbb{L}(\psi, \tau) = -\tau \mathrm{KL}(q_\psi \| q^*_{\tau,\theta}) + C. \tag{7}$$

Where $q^*_{\tau,\theta} \propto p(x,y) e^{\frac{p(x,y)}{q_{\psi_0}(x,y)} \frac{\ell(x,y;\theta)}{\tau}}$ and C is a constant in ψ. In other words, maximizing $\mathbb{L}$ in ψ is equivalent to minimizing the KL divergence between q_ψ and $q^*_{\tau,\theta}$. One difficulty with this objective is that $q^*_{\tau,\theta}$ depends upon the unknown probability density $p(x,y)$. We avoid this problem by treating the density ratio $\frac{p(x,y)}{q_{\psi_0}(x,y)}$ as a constant, which is closely related to assumptions that have been used successfully in past formulations of DRO (Oren et al., 2019). Empirically, we find that incorporating q_{ψ_0} as a surrogate for p is a serviceable approximation, as demonstrated in Section 4.

Reversing the KL Minimizing the KL divergence in this direction is difficult because it entails optimizing an expectation in q_ψ over ψ. Moreover, computing this KL necessitates access to the true theoretical density $p(x,y)$ in order to compute $q^*_{\tau,\theta}(x,y)$ in the argument of the expectation, which is unknown in practice.[3] To sidestep these issues, we elect to minimize the reverse direction $\mathrm{KL}(q^*_{\tau,\theta} \| q_\psi)$ instead. Due to the KL divergence being non-symmetric, this is a rather crude approximation[4], the implications of which are discussed in Norouzi et al. (2016). However, we find that dramatically stabilizes the gradient dynamics while still yielding good adversaries, as observed empirically in Section 4.4. Discarding the entropy term (constant in ψ), the resulting problem is equivalent to minimizing

$$\mathcal{L}_{\text{rev}}(\psi, \tau) = -\frac{1}{Z_{\tau,\theta}} \mathbb{E}_p \, e^{\frac{\ell(x,y;\theta)}{\tau}} \log q_\psi(x,y) \tag{8}$$

in ψ, where $Z_{\tau,\theta} = \mathbb{E}_p \, e^{\frac{\ell(x,y;\theta)}{\tau}}$ is the normalizer of q^*. In this case, we can estimate this expectation by substituting the empirical distribution $\hat{p}$ for p in the expectation.

Computing the Normalizer Approximating the inverse normalizer $\frac{1}{Z_{\tau,\theta}}$ in a minibatch yields a biased estimator. On the other hand, computing $Z_{\tau,\theta}$ over the entire training data at each step is prohibitive since it requires computing the loss of every single example. As a middle ground, we keep a running normalizer $\tilde{Z}_k$ computed from the average of the weights over a fixed number k of consecutive minibatches. In other words, if B_i and θ_i denote the minibatch and adversary parameters at step i respectively, the normalizer at step t will be

$$\tilde{Z}_k = \frac{1}{\sum_{i=t-k}^{t} |B_i|} \sum_{i=t-k}^{t} \sum_{x,y \in B_i} e^{\frac{\ell(x,y;\theta_i)}{\tau}}. \tag{9}$$

If k is too low, there is a risk of under-estimating the normalizer, especially if the distribution of weights contains infrequent high weight samples. On the other hand, if k is too high there is a risk of using "stale" weights in the normalizer. In experiments, we treat k as a hyper-parameter.

3.3 Optimal Stopping

When should one stop training a model with P-DRO? In ERM it is customary to stop training after the empirical risk — periodically evaluated on a held out validation dataset — stops decreasing. This is particularly important to prevent over-fitting to the training data. However, it is not an appropriate criterion for P-DRO, since the model is not trained to minimize empirical risk in the first place. A more pertinent choice is to compare the robust validation losses

$$\mathcal{L}_{\text{robust,valid}}(\theta) = \sup_{q_\psi \in \mathcal{Q}} \underbrace{\frac{1}{|D_{\text{valid}}|} \sum_{x,y \in D_{\text{valid}}} \frac{q_\psi(x,y)}{q_{\psi_0}(x,y)} \ell(x,y;\theta)}_{:= \mathcal{L}_{\text{valid}}(\theta,\psi)}. \tag{10}$$

[3]Note that substituting the empirical distribution $\hat{p}$ for p poses issues here, for q_ψ is not absolutely continuouis with respect to $\hat{p}$.

[4]For instance, the optimum of the reverse KL doesn't necessarily match that of the forward KL within the parametric confusion set $\mathcal{Q}$

However, finding the inner supremum for each of the T evaluation checkpoints $\theta_1 \dots \theta_T$ is expensive as it requires solving T independent optimization problems. Instead, we leverage the existence of adversaries ψ_t associated with each model θ_t, as well as the initial adversary ψ_0 and take the maximum over the $T+1$ adversaries $\{\psi_0, \dots, \psi_T\}$. Since our relaxation of the P-DRO objective loosens the KL constraint, we need weed out adversaries which might violate it. Specifically, we estimate the $\mathrm{KL}(q_\psi \| p) = \mathbb{E}_p \, q_\psi / p \log q_\psi / p$ on the validation set, using q_ψ / q_{ψ_0} as a stand-in for q_ψ / p, and reject all adversaries for which the result is greater than a threshold, which we set to $\log 10$ based on preliminary experiments detailed in Appendix C.1.[5] We refer to this stopping criterion as **Minmax**.

Computing the full min-max necessitates keeping track of T models and $T+1$ adversaries, which is ponderous when the model is large. As a solution, we propose an approximation, **Greedy-Minmax**, in which we only keep one best model θ^*. At each evaluation step T, we compare θ_T to θ^*, and update θ^* to whichever achieves lower robust validation loss over the $T+1$ adversaries $\psi_0, \dots, \psi_T$. By keeping track of only one additional model, and using the weights $\frac{q_{\psi_t}(x_i, y_i)}{q_{\psi_0}(x_i, y_i)}$ of individual examples in D_{valid} as sufficient statistics for computing the loss against each adversary, Greedy-Minmax can be achieved with space complexity $2d_{\text{model}} + T|D_{\text{valid}}|$, which is much more efficient than the $T(d_{\text{model}} + d_{\text{adv}})$ of Minmax.

3.4 Hyper-parameter Selection

Our proposed P-DRO method relies on 3 different hyper-parameters (in addition to the model's hyper-parameters): the adversary learning rate λ, the temperature τ and the size of the renormalizing window k. As a consequence, we need a reliable criterion for deciding which of two configurations is better. This model comparison bears many similarities with the stopping problem described above. Therefore, we resort to a similar solution: given two models θ_1, θ_2 trained with P-DRO, and their respective adversaries $\{\psi_0^1, \dots, \psi_T^1\}$, $\{\psi_0^2, \dots, \psi_T^2\}$ (for instance, the adversaries associated with θ_1 and θ_2 at periodic checkpoints during training), we select the best model following

$$\theta^* = \underset{\theta \in \{\theta_1, \theta_2\}}{\arg\min} \max_{\psi \in \{\psi_0^1, \dots, \psi_T^1, \psi_0^2, \dots, \psi_T^2\}} \mathcal{L}_{\text{valid}}(\theta, \psi). \tag{11}$$

4 Experimental Analysis of P-DRO

Before moving on to a real world scenario in Section 5, we first demonstrate that P-DRO is able to learn robust models in a synthetic Natural Language Processing (NLP) task, and perform ablation studies to examine the importance of the various modifications described in Section 3.

4.1 Experimental Setting

For analysis purposes, we design a simple NLP task amenable to DRO. We specifically choose NLP as a domain due to the striking success of language models as generative models of textual data (Sundermeyer et al., 2012; Radford et al., 2018), which can be used to model the uncertainty set. We base our task off of the binary version of the Stanford Sentiment Treebank dataset (SST-2; Socher et al. (2013)), which we modify to introduce spurious correlation. Specifically, we introduce a distractor token to some sentences. The distractor we use consists of prepending "so , " to the sentence ("i hated this movie" $\longrightarrow$ "so , I hated this movie"), which doesn't change the underlying sentiment. The resulting samples can be categorized in 4 "groups" depending on their label (positive or negative) and the presence or absence of the distractor. In particular, we add this distractor to 95% of the negative reviews and 5% of the positive reviews in the training and validation set, so that the presence of the distractor strongly correlates with negative sentiment (a similar construction is proposed in (Utama et al., 2020)). In the test data, we modify 50% of all sentences for each class equitably to ensure that there is enough data in each group. We call this modified task **BiasedSST**.

For the classifier, we train a simple one layer BiLSTM model with embedding/hidden dimension 300. For the adversary, we adopt an auto-regressive transformer model based on the successful GPT-2 language model architecture but with 6 layers, a dimension of 512 and 8 attention heads

[5] To simplify notation, this additional constraint is implicit in the rest of this section.

(we experiment with a smaller, LSTM based adversary in Appendix C.2). In order to model the input output pair (x, y), we pre-pend a special label-specific token to sentences before running them through the language model. We train the model with Adam (Kingma & Ba, 2014) and the adversary with vanilla stochastic gradient descent (which we found more stable in experiments). We refer to Appendix B for specific details of the experimental setting.

4.2 P-DRO CAN LEARN ROBUST MODELS

We train 7 models with P-DRO on BiasedSST using different hyper-parameters for the adversary. We start from configuration $\lambda = 10^{-4}$, $\tau = 0.01$, $k = 5$, and for each hyper-parameter we run a configuration with a smaller and a higher value, keeping all other hyper-parameters the same. We train for 50 epochs and select the best model using the strategies described in Section 3.

Table 1: Average and robust accuracies on BiasedSST. Underlining indicates statistically significant difference compared to ERM ($p < 0.05$)

	Robust	Average
ERM	2.15 ± 0.97	52.43 ± 1.82
Topic CVaR	5.18 ± 1.46	53.82 ± 0.77
NonParam	18.75 ± 5.55	60.89 ± 2.48
P-DRO	34.98 ± 9.39	67.73 ± 1.68
Oracle DRO	67.71 ± 3.03	74.04 ± 2.19

We also compare three other approaches. First, to appreciate how well the model could perform if the groups were known at training time, we train with Group-DRO on the oracle groups using an exponentiated-gradients based online algorithm (**Oracle DRO**; Sagawa et al. (2020)). Second, we implement **Topic CVaR** (Oren et al., 2019), a method for DRO on NLP where the uncertainty set is determined by mixtures of a topic model. Finally, we compare to non-parametric DRO with a Kullback-Leibler (KL) constrained uncertainty set (Hu & Hong, 2013; Hu et al., 2018), which we adapt to fit our online mini-batch training setting (**NonParam**). We refer to Appendix B.3 for details and hyper-parameters of the baselines.

We report the worst-case ("robust") accuracy over all groups on the test set, as well the average accuracy in Table 1 (we report the mean and standard deviation over 5 runs). We find that both Topic-CVaR and P-DRO are more robust than ERM, but the latter outperforms the former close to 30 points, achieving 52% of Oracle DRO's robust accuracy, while not leveraging any information on the oracle groups.

4.3 OPTIMAL STOPPING AND HYPER-PARAMETER SELECTION ABLATION

To understand the importance of the optimal stopping and hyper-parameter selection strategy described in Section 3.3, we perform an ablation on the BiasedSST dataset comparing 4 strategies:

- **Average**: models are selected based on their average zero-one loss (*i.e.* error rate) on the unmodified validation set. This is the baseline stopping criterion.
- **Minmax**: selection based on the adversaries (as described in Section 3.3), with and without the **KL constraint**, as well as its variant **Greedy-Minmax** for stopping.
- **Oracle**: in this setting the groups are known (in the validation set), and models are selected based on their error rate on the worst performing group. This is the optimal criterion for the group-DRO setting we are considering.

To compare stopping criterions experiments, we only consider one set of hyper-parameters: $\lambda = 10^{-4}$, $k = 5$ and $\tau = 0.01$. From the robust validation accuracies reported in Table 2a, we first observe that Average stopping results in a robust accuracy of 0, highlighting the necessity for a suitable stopping criterion. We find that Minmax, especially with a KL constraint, is a much better strategy, recovering $\approx 60\%$ of the performance achievable with Oracle stopping. Notably, the Greedy-Minmax variant which we use in practice reaches very close results (< 1 point difference) despite its requiring to keep track of only 2 out of the 50 model checkpoints at any time.

To understand the effectiveness of the Minmax strategy for selecting hyper-parameters. We take the models trained in Section 4.1, but select the best hyper-parameters using the different strategies described above. Results, shown in Table 2b, confirm that Minmax (with the KL constraint) is a better choice than Average for selecting hyper-parameters, even though the improvement is not as striking as for stopping.

Table 2: Effect of different optimal stopping and hyper-parameter selection strategies on robust validation accuracy.

(a) Optimal stopping

Criterion	Robust accuracy
Average	0.00 ± 0.00
Minmax	25.22 ± 13.01
+KL constraint	31.30 ± 10.07
+Greedy-Minmax	**32.17** ± ± 11.20
Oracle	50.95 ± 5.01

(b) Hyper-parameter selection

Criterion	Robust accuracy
Average	31.03 ± 12.16
Minmax	28.62 ± 12.37
+KL constraint	**35.65** ± 11.47
Oracle	38.26 ± 13.01

4.4 Importance of $\mathcal{L}_{\text{REV}}$

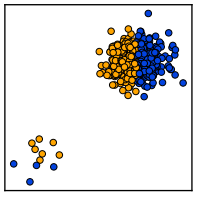

Figure 1: A toy classification task.

Finally, we investigate the importance of modifying the adversary's objective as described in Section 3.2. For this experiment, we devise a simpler toy task on which directly training the constrained DRO objective is possible. Specifically, we consider the two-dimensional binary classification problem pictured in Figure 1. The training data consists of 10,000 points partitioned in two normally distributed "domains" with a 1:50 sampling ratio and different classification boundaries. We train a logistic regression model, which cannot perfectly fit the training data and must trade-off between accuracy on each domain. For the sake of simplicity, we only model the input variables x[6] as isotropic normal distributions with fixed variance: the adversaries' parameter $\psi \in \mathbb{R}^2$ represents the location of the Gaussian (we fix the variance to the empirical variance of the data).

We compare 3 different versions of P-DRO: first, naive simultaneous gradient descent on the zero-sum game, without any constraint on the adversary (**bare P-DRO**), then the same, but with an approximation of the explicit KL constraint between q_ψ and q_{ψ_0} (**+KL constraint**; see Appendix A.2 for more details). Finally we report results using our relaxation and the KL reversal described in Section 3.2 (**+$\mathcal{L}_{\text{rev}}$**). For each setting, we report the average and robust accuracy with mean and standard deviation over 10 runs. For the KL constraint and the relaxation, we report the best results among 4 values of the KL bound κ and the temperature τ respectively.

Table 3: Ablation of P-DRO to train the linear model on the toy task. We report accuracy on both domains, as well as robust accuracy.

	Average	Robust
ERM	**84.66** ± 0.10	49.75 ± 0.05
bare P-DRO	49.97 ± 0.10	49.85 ± 0.10
+kl constraint	76.63 ± 9.93	58.41 ± 9.25
+$\mathcal{L}_{\text{rev}}$	76.97 ± 0.43	**64.32** ± 1.31

In Table 3, we observe that bare P-DRO is too unstable and systematically diverges. The addition of a KL constraint mitigates this behaviour, but the zero-sum objective is still unstable, as evidenced by the high standard deviations. Finally, we find that the addition of $\mathcal{L}_{\text{rev}}$ stabilizes the training process greatly, leading to consistently high robust accuracy.

5 P-DRO in Practice: Case Study of Toxicity Detection

In this section, we demonstrate the effectiveness of P-DRO in the more realistic setting of toxicity detection, the task of recognizing various forms of toxic language (eg. hate speech or offensive language). Identifying online abuse on the internet is a crucial challenge, and has garnered much interest in the NLP community (Schmidt & Wiegand, 2017; Fortuna & Nunes, 2018). However, recent work (Sap et al., 2019) has shown that there is strong correlation between toxic labels and the presence of certain markers of dialects of English spoken by minority groups. This correlation is in turn amplified by hate speech classifiers trained on such data, leading to biased prediction.

[6] In other words, we set $q_\psi(x, y) = p(y \mid x) q_\psi(x)$, where $p(y \mid x)$, is the true conditional which will be canceled out in the ratio $\frac{q_\psi(x,y)}{q_{\psi_0}(x,y)}$.

Table 4: Robust test accuracy on the DWMW17 and FDCL18 toxicity detection tasks.

(a) No group information.

	DWMW17		FDCL18	
	Robust	Average	Robust	Average
ERM	53.19 ± 1.70	69.44 ± 0.53	19.57 ± 7.00	**81.56** ± 0.26
Topic CVaR	45.26 ± 3.47	61.68 ± 5.02	16.48 ± 5.46	80.49 ± 0.49
NonParam	54.13 ± 1.14	**70.54** ± 0.64	17.54 ± 6.41	81.20 ± 0.11
P-DRO	**69.06** ± 1.70	69.69 ± 2.50	**30.25** ± 10.13	79.91 ± 1.41
Oracle DRO	74.50 ± 1.74	65.79 ± 0.76	55.23 ± 3.97	72.43 ± 2.61

(b) With group information on the validation set.

	Robust	Average	Robust	Average
ERM	53.15 ± 0.87	69.64 ± 1.01	34.07 ± 3.20	78.78 ± 0.38
Topic CVaR	52.02 ± 1.26	**69.11** ± 0.49	34.82 ± 3.73	**79.59** ± 0.85
NonParam	49.41 ± 5.60	58.53 ± 5.71	43.13 ± 6.97	69.51 ± 3.07
P-DRO	**63.05** ± 4.25	63.07 ± 3.92	**47.61** ± 4.53	74.82 ± 1.90
Oracle DRO	74.50 ± 1.74	65.79 ± 0.76	55.23 ± 3.97	72.43 ± 2.61

Our results on BiasedSST suggest that P-DRO can provide one solution to preventing models from absorbing spurious correlations present in their training data, even in the absence of protected attributes (such as language variety here).

5.1 Experimental Setting

Following Sap et al. (2019) and Xia et al. (2020), we perform experiments on two datasets: **DWMW17** (Davidson et al., 2017), a corpus of 25K tweets classified in three categories: *hate speech* (6%), *offensive* (76%) and *neither* (18%), and **FDCL18** (Founta et al., 2018), a 100k sized dataset, also collected from Twitter and annotated with an additional *spam* label, with the following breakdown by categories: *hateful* (5%), *abusive* (27%), *normal* (54%) and *spam* (14%).

The released version of these datasets does not contain information on the dialect of each user. In order to be able to evaluate our models, and to train an Oracle DRO baseline, we follow Sap et al. (2019) and use annotations provided by the dialect classifier described in Blodgett et al. (2016) to label each example as one of four English varieties: White-aligned, African American, Hispanic, and Other. Note that, as these are automatically obtained labels, the groups may not exactly correspond to the actual racial sociolects, however Sap et al. (2019) does report that they correlate highly with self-reported race, and they serve as a useful proxy in the absence of manual annotation.

We formulate the group-DRO problem by separating each dataset into independent groups identified by both language variety and label, for a total of 12 and 16 groups for DWMW17 and FDCL18 respectively. Some of these groups are severely under-represented in the test set. In order to make our robust accuracy results reliable yet still representative of the under-represented groups, we combine groups that contain less than 100 samples into a single group to compute robust test accuracies.

On DWMW17, we train the same BiLSTM model as described in Section 4.3. To illustrate the applicability of P-DRO to other model architectures, we pick BERT (Devlin et al., 2018), a large scale pre-trained model as a classifier on FDCL18. In both cases, we adopt the Transformer architecture described in Section 4.3 as the adversary. We train the adversary with a temperature of $\tau = 0.01$ and a normalizing window $k = 10$. To demonstrate the efficacy of automatic hyper-parameter selection in the P-DRO setting, we delegate the choice of the adversary's learning rate λ to grid-search, training 3 models with $\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}\}$ and selecting the best using the Minmax criterion described in Section 3.4. We also report numbers for Oracle DRO and Topic CVaR. Results are averaged over 5 runs, each with a different random seed.

5.2 CAN P-DRO PRODUCE MORE ROBUST MODELS?

Table 4a reports the robust test accuracies of all models on both tasks. Importantly, except for Oracle DRO, none of the methods compared here necessitate any knowledge of the groups, neither in the training nor validation data. We observe that in both settings P-DRO is able to achieve higher robust accuracy than ERM, Topic-CVaR and NonParam.

This suggests P-DRO as a useful option in case no group information whatsoever is available. However, in practice, it may be feasible to annotate at least a small amount of data with group information. To emulate this scenario, we perform the same experiment, but assume that group annotations are available on the validation data, which we use to determine optimal stopping and hyper-parameters. Results for this setting are reported in Table 4b. We find that, while the use of robust validation accuracy yields more robust models even for ERM (especially on FDCL18), P-DRO is still the best alternative that doesn't require group annotation on the training data.

6 IMPLICATIONS AND OUTLOOK

We have shown that there is promise in using parametric families of neural generative models for defining the uncertainty set in distributionally robust optimization. While we only perform experiments on NLP tasks, this approach can, in theory, be applied in any modality and in future work we hope to pursue this direction. In such cases where good quality generative models are unavailable, or such model cannot produce densities efficiently, an interesting direction would be to model the likelihood ratio q_ψ/p directly. This alternative formulation poses different implementation challenges, and we leave it as a promising avenue for future research.

REFERENCES

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 173–182, 2016. URL http://proceedings.mlr.press/v48/amodei16.html.

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 354–363, 2018. URL http://proceedings.mlr.press/v80/balduzzi18a/balduzzi18a.pdf.

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013a.

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013b.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1119–1130, 2016. URL https://www.aclweb.org/anthology/D16-1120.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2017.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 67–73, 2018.

John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018. URL `https://arxiv.org/pdf/1810.08750.pdf`.

John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Louis Faury, Ugo Tanielian, Elvis Dohmatob, Elena Smirnova, and Flavian Vasile. Distributionally robust counterfactual risk minimization. In *Proceedings of the 34th Meeting of the Association for Advancement of Artificial Intelligence (AAAI)*, volume 34, pp. 3850–3857, 2020.

Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. 2018.

Andreas Fuster, Paul Goldsmith-Pinkham, Tarun Ramadorai, and Ansgar Walther. Predictably unequal? the effects of machine learning on credit markets. *The Effects of Machine Learning on Credit Markets (November 6, 2018)*, 2018.

Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2672–2680, 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Patrick Grother, Mei Ngan, and Kayee Hanaoka. *Face Recognition Vendor Test (FVRT): Part 3, Demographic Effects*. National Institute of Standards and Technology, 2019.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL `https://www.aclweb.org/anthology/2020.acl-main.740`.

J. Hershey and P. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 4:IV–317–IV–320, 2007.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Dirk Hovy and Anders Søgaard. Tagging performance correlates with author age. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 483–488, 2015. URL https://www.aclweb.org/anthology/P15-2079.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 2029–2037, 2018. URL http://proceedings.mlr.press/v80/hu18a/hu18a.pdf.

Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 2013.

Hisham Husain. Distributional robustness with ipms and links to regularization and gans. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NIPS)*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Viet Anh Nguyen, Nian Si, and Jose Blanchet. Robust bayesian classification using an optimistic score ratio. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Mohammad Norouzi, Samy Bengio, zhifeng Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. Reward augmented maximum likelihood for neural structured prediction. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1723–1731. 2016.

Yonatan Oren, Shiori Sagawa, Tatsunori Hashimoto, and Percy Liang. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4227–4237, 2019. URL https://www.aclweb.org/anthology/D19-1432.

Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, 1994. ISBN 0262150417.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

Hamed Rahimian and Sanjay Mehrotra. Distributionally Robust Optimization: A Review. *arXiv preprint arXiv:1908.05659*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/pdf/1911.08731.pdf.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1668–1678, 2019. URL https://www.aclweb.org/anthology/P19-1163.

Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the 5th International Workshop on Natural Language Processing for Social Media (SocialNLP)*, pp. 1–10, 2017. URL https://www.aclweb.org/anthology/W17-1101.

Satinder P. Singh, Michael J. Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, UAI '00, pp. 541–548, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607099.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1631–1642, 2013.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association (InterSpeech)*, 2012.

Prasetya Ajie Utama, N. Moosavi, and Iryna Gurevych. Towards debiasing nlu models from unknown biases. *arXiv preprint arXiv:2009.12303*, 2020.

Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *arXiv preprint arXiv:1906.01529*, 2019.

Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. Demoting racial bias in hate speech detection. In *Proceedings of the 9th International Workshop on Natural Language Processing for Social Media (SocialNLP)*, pp. 7–14, 2020. URL https://www.aclweb.org/anthology/2020.socialnlp-1.2.

A Derivations

A.1 Reorganizing the Lagrangian $\mathbb{L}(\psi, \tau)$

Let us write the Lagrangian $\mathbb{L}$ explicitly:

$$\mathbb{L}(\psi,\tau) = \mathbb{E}_{(x,y)\sim q_\psi} \frac{p(x,y)}{q_{\psi_0}(x,y)} \ell(x,y,\theta) - \tau\left(\mathrm{KL}(q_\psi \| p) - \kappa\right) \tag{12}$$

$$= \mathbb{E}_{(x,y)\sim q_\psi} \frac{p(x,y)}{q_{\psi_0}(x,y)} \ell(x,y,\theta) - \tau\, \mathbb{E}_{(x,y)\sim q_\psi} \log \frac{q_\psi(x,y)}{p(x,y)} + \tau\kappa \tag{13}$$

$$= \tau\, \mathbb{E}_{(x,y)\sim q_\psi} \log \left(\frac{p(x,y) e^{\frac{p(x,y)}{q_{\psi_0}(x,y)} \frac{\ell(x,y,\theta)}{\tau}}}{q_\psi(x,y)} \right) + \tau\kappa \tag{14}$$

$$= \tau(\kappa - \mathrm{KL}(q_\psi \| q^*_{\tau,\theta})) + \log \left(\mathbb{E}_{(x,y)\sim p}\, e^{\frac{p(x,y)}{q_{\psi_0}(x,y)} \frac{\ell(x,y,\theta)}{\tau}} \right) \tag{15}$$

This last step requires that the log moment generating function of ℓ under p exist for τ. In most scenarios we consider, ℓ is typically the negative log likelihood of a neural network model, which is generally bounded. Therefore the moment generating function is defined everywhere.

Note that the KL term is the only one dependent on ψ, therefore maximizing $\mathbb{L}$ for ψ is equivalent to maximizing $-\mathrm{KL}(q_\psi \| q^*_{\tau,\theta})$, in other words minimizing $\mathrm{KL}(q_\psi \| q^*_{\tau,\theta})$

A.2 Enforcing the KL Constraint in the Toy Setting

Even in this simplest setting, the exact KL between q_ψ (a gaussian) and p (a mixture of gaussians) does not have an analytical expression (Hershey & Olsen, 2007). Instead, we fall back on enforcing the KL constraint between q_ψ and q_{ψ_0}, both isotropic gaussians with the same standard deviation. Let μ and $\mu_0 \in \mathbb{R}^2$ denote their respective mean, and $\sigma > 0$ their standard deviation. In this context, their KL divergence reduces to:

$$\mathrm{KL}(q_\psi \| q_{\psi_0}) = \mathrm{KL}(q_{\psi_0} \| q_\psi) = \frac{1}{2\sigma^2} \|\mu - \mu_0\|^2$$

In other words, the KL divergence is equivalent to the euclidean distance between the distributions' means. We use this fact to project ψ (in the KL sense) onto $B_\kappa = \{\hat{\psi} \mid \mathrm{KL}(q_{\hat{\psi}} \| q_{\psi_0}) < \kappa\}$:

$$\begin{aligned}\mathrm{proj}_{B_\kappa}(\psi) &:= \operatorname*{arg\,min}_{\hat{\psi} \in B_\kappa} \mathrm{KL}(q_\psi \| q_{\hat{\psi}}) \\ &= \psi_0 + \frac{\sqrt{2\kappa}\sigma}{\|\psi - \psi_0\|}(\psi - \psi_0)\end{aligned}$$

B Experimental Details

We describe in more details some of the experimental settings for our NLP experiments. To further facilitate reproducibility, we commit to releasing our code upon acceptance.

B.1 Model Settings

In all experiments, we split the text into sub-word tokens using the tokenizer described in (Devlin et al., 2018). During training, we sample minibatches that contain at most 64 sentences or 2500 tokens, whichever is greater, in order to prevent GPU memory overflow in case of long sentences. We train all models with Adam (Kingma & Ba, 2014) with an initial learning rate of 2×10^{-5}, which we decay linearly at each step until the end of training. We validate the models every epoch. For BERT, we start from the `bert-base-uncased` checkpoint.

B.2 Adversary Settings

In all experiments, we use a Transformer model based on the GPT-2 architecture (Radford et al., 2019) to serve as the adversary. In order to initialize the adversary (to obtain ψ_0), we first pre-train the model on a generic, relatively large language modeling dataset, WikiText-103 (Merity et al., 2017). We also use a batch size of 64 samples or 2500 tokens, and train with Adam for 10 epochs, with a fixed learning rate of 3×10^{-4}. Then, we fine-tune this model on each dataset, this time minimizing the negative log-likelihood of the (x, y) pair (by introducing the special "[label]" token as described in Section B), using the same hyper-parameters but a smaller learning rate (10^{-5}). We find that, due to the small to medium size of the datasets under consideration, this LM pretraining step helped achieve lower error on the generative modeling task.

B.3 Baseline Settings

B.3.1 Topic CVaR

To train the topic model for Topic CVaR, we first pre-process the text by removing all punctuation, urls and user mentions (for twitter data). Importantly, we remove stop-words for our toxicity experiments but *not* for our BiasedSST experiment. This is because the distractor token we use ("so") belongs to most English stop words lists, and removing it would completely prevent the topic model from picking up on the groups of interest. We then estimate the parameters of the model with Gensim[7] and use similar settings as Oren et al. (2019) ($\alpha = 0.1$, $\beta = 1.0$), setting the number of topics to 10.

[7] `https://radimrehurek.com/gensim/`

For both Oracle-DRO and Topic-CVaR, we use the algorithm proposed in Sagawa et al. (2020) to estimate the worst-case group (either oracle group or topic in Topic-CVaR) online during training. We perform grid-search over $\{1, 0.1, 0.01\}$ to find the best learning rate for the group weights update. For Oracle DRO, the best model is simply selected by robust validation accuracy. For Topic CVaR, unless specified otherwise, we select the model with the lowest worst-case error over all topics.

B.3.2 NonParam

In the KL-constrained non-parametric setting, the min-max problem reads

$$\min_{\theta} \max_{\substack{q \text{ s.t.} \\ \mathrm{KL}(q_\psi \| p) \leq \kappa}} \mathbb{E}_{(x,y)\sim q} \ell(x, y, \theta). \tag{16}$$

Here, κ is the desired radius of the KL ball, and is treated as a hyper-parameter. The solution of the inner maximum has an analytical solution of the form $q^*_\theta = \frac{a}{Z_{\theta,\tau^*}} p(x,y) e^{\frac{\ell(x,y;\theta)}{\tau^*}}$ (see Hu & Hong (2013); Hu et al. (2018) for details) with $Z_{\theta,\tau^*} = \mathbb{E}_p e^{\frac{\ell(x,y;\theta)}{\tau^*}}$ and τ^* such that

$$\mathrm{KL}(q^* \| p) = \mathbb{E}_p \frac{e^{\frac{\ell(x,y;\theta)}{\tau^*}}}{Z_{\theta,\tau^*}} \left(\frac{\ell(x,y;\theta)}{\tau^*} - \log Z_{\theta,\tau^*} \right) = \kappa.$$

Note that both computing Z_{θ,τ^*} and $\mathrm{KL}(q^* \| p)$ require taking expectations over p. In our setting, where $\ell(x, y; \theta)$ is the output of a large neural network, we cannot afford to take this expectation over the entire training data at each step. Instead, we fall back to taking the average over each mini-batch. We find τ^* with binary search in $\log_{10}$ space within the $[10^{-10}, 10^{10}]$ interval and clip to the lowest or highest value should the result lie outside the search interval.

In all experiments, we try 4 different values for κ: 0.01, 0.1, 1 and 10. Unless indicated otherwise, we perform early stopping and hyper-parameter selection using our Minmax criterion using the non-parametric weights as adversaries on the validation data.

C Additional Experiments

C.1 Minmax Validation KL Threshold

The Monte-Carlo estimate of $\mathrm{KL}(q_\psi \| p)$ on the validation set is $\frac{1}{|D_{\text{valid}}|} \sum_{x,y \in D_{\text{valid}}} \frac{q_\psi(x,y)}{p(x,y)} \log \frac{q_\psi(x,y)}{p(x,y)}$. Similarly to Section 3, we approximate the (unknown) likelihood ratio $\frac{q_\psi(x,y)}{p(x,y)}$ with $\frac{q_\psi(x,y)}{q_{\psi_0}(x,y)}$.

We want to reject all adversaries where this approximated KL is greater than some threshold, κ_{valid}, but how do we choose a good value for κ_{valid}? Consider an adversary which selects a fraction of the validation data of size $\alpha |D_{\text{valid}}|$ for some $\alpha \in (0, 1]$. In such a case, the likelihood ratio is $1/\alpha$ on this subset and 0 everywhere else, and the resulting KL estimate will be $\log \alpha$. In other words, choosing a threshold of κ_{valid} means allowing the adversary to potentially select any subset of size at least $1/e^{\kappa_{\text{valid}}}$ of the original data. Our heuristic choice, $\log 10$, corresponds to allowing subsets of size at least 10% of $|D_{\text{valid}}|$.

Of course, this is only a heuristic because the adversary can reweight the validation set non-uniformly. To assess the effect of κ_{valid} on Greedy-Minmax, we compute the average robust validation error of the selected model across 5 runs for 3 different values of the adversary's learning rate. Results on BiasedSST, depicted in Figure 2, show that adversaries with higher learning rate are more sensitive to the choice of threshold, but all values of κ_{valid} between $\log 5$ and $\log 20$ seem to work for these settings.

C.2 P-DRO Experiments with an LSTM Adversary

We replicate the experiments BiasedSST experiments in Section 4, but this time using a smaller generative model, which is unlikely to generate good samples. Specifically, we use a one layer

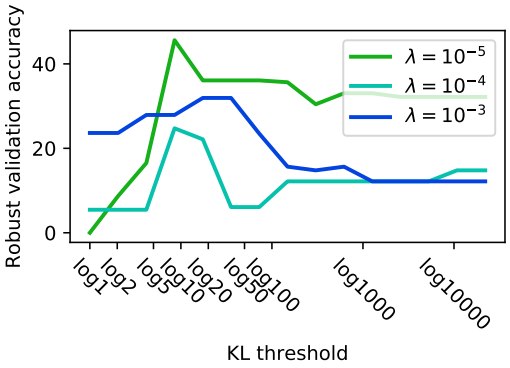

Figure 2: Evolution of the robust validation accuracy of the model selected by Greedy-Minmax as a function of the KL threshold κ_{valid}

Table 5: Average and robust accuracies on BiasedSST when P-DRO is trained with an LSTM adversary. Underlining indicates statistically significant difference compared to ERM ($p < 0.05$)

	Robust	Average
ERM	2.15 $\pm$ 0.97	52.43 $\pm$ 1.82
Topic CVaR	5.18 $\pm$ 1.46	53.82 $\pm$ 0.77
NonParam	18.75 $\pm$ 5.55	60.89 $\pm$ 2.48
P-DRO	43.68 $\pm$ 4.93	68.72 $\pm$ 0.87
Oracle DRO	67.71 $\pm$ 3.03	74.04 $\pm$ 2.19

LSTM model (Hochreiter & Schmidhuber, 1997) with embedding and hidden dimension 256. We only perform grid-search over $\lambda \in [10^{-5}, 10^{-4}, 10^{-3}]$ and select the best with Minmax.

Once pre-trained on the BiasedSST dataset, this model achieves a perplexity of 227.0, more than 4 times worse than the transformer model we use in other experiments (49.8). However, as evidenced by its robust accuracy displayed in Table 5, P-DRO is still able to learn a robust model. We take this as evidence that the re-weighting introduced in Section 3 helps stabilize training even when q_ψ is not a perfect model of the data.

C.3 Influence of hyper-parameters on P-DRO

We study the influence of the 3 hyper-parameters τ (temperature), k (size of the renormalization window) and λ (learning rate of the adversary) on the performance of P-DRO. All experiments are run on the BiasedSST dataset, and the analysis proceeds as follows: starting from configuration $\tau = 0.01$, $k = 5$ and $\lambda = 10^{-4}$ and vary each of the hyper-parameters independently. We report two numbers for each configuration: robust accuracy of the best model using Greedy-Minmax stopping and using Oracle stopping. The latter is useful to disentangle the effect of the stopping criterion.

As seen in the results shown in Table 6, we find that τ has the least effect on robust accuracies. While the renormalization window parameter k has some effect on optimal stopping, the best robust accuracy achieved by the model (with oracle stopping) varies little. We observe the adversary's learning rate λ to be the most sensitive hyper-parameter, which is why we restrict our grid-search to λ in Section 5.

Table 6: Effect of hyper-parameters on robust validation accuracy on BiasedSST

	Robust accuracy	
	Minmax stopping	Oracle stopping
$\lambda = 10^{-5}$	28.62 ± 12.37	45.10 ± 4.50
$\lambda = 10^{-4}$	44.74 ± 3.24	50.43 ± 5.05
$\lambda = 10^{-3}$	25.57 ± 10.33	38.70 ± 2.97
$\tau = 0.1$	39.72 ± 5.55	50.00 ± 4.98
$\tau = 0.01$	44.74 ± 3.24	50.43 ± 5.05
$\tau = 0.001$	44.74 ± 3.24	50.87 ± 5.09
$k = 0$	41.98 ± 4.48	49.60 ± 5.39
$k = 5$	44.74 ± 3.24	50.43 ± 5.05
$k = 10$	32.17 ± 11.20	50.95 ± 5.01