The Role of Learning Algorithms in Collective Action

Omri Ben-Dov* [1 2] **Jake Fawkes*** [3] **Samira Samadi** [1 2] **Amartya Sanyal** [1]

Abstract

Collective action in machine learning is the study of the control that a coordinated group can have over machine learning algorithms. While previous research has concentrated on assessing the impact of collectives against Bayes (sub-)optimal classifiers, this perspective is limited in that it does not account for the choice of learning algorithm. Since classifiers seldom behave like Bayes classifiers and are influenced by the choice of learning algorithms along with their inherent biases, in this work we initiate the study of how the choice of the learning algorithm plays a role in the success of a collective in practical settings. Specifically, we focus on distributionally robust optimization (DRO), popular for improving a worst group error, and on the ubiquitous stochastic gradient descent (SGD), due to its inductive bias for "simpler" functions. Our empirical results, supported by a theoretical foundation, show that the effective size and success of the collective are highly dependent on properties of the learning algorithm. This highlights the necessity of taking the learning algorithm into account when studying the impact of collective action in machine learning.

1. Introduction

With the rapid increase in deployed machine learning models, a large number of firms rely on data contributed by users (Gerlitz & Helmond, 2013) to train their algorithms. In response to this, consumers and users have searched for ways to alter their data to influence the outputs of such models (Chen, 2018; Burrell et al., 2019; Rahman, 2021). *Algorithmic collective action* (Olson, 1965; Hardt et al., 2023) has emerged as a formal framework to study the effect a coordinated group of individuals can have on such models, by altering the data they provide to the firm. This naturally leads to various technical questions regarding how large the collective needs to be, how effective different data altering strategies are, and which details about firm's algorithm can be leveraged by the collective to achieve the best results.

Hardt et al. (2023) initiated the formal study into the success of different collective action strategies. Their work provided theoretical analysis based on the fractional size of the collective and on the properties of the signal with regards to the original data distribution. However, their results only hold for Bayes (sub-)optimal classifiers which do not immediately adapt to peculiarities of practical learning algorithms. In this work, we investigate two types of commonly used learning algorithms that exhibit distinct properties: (1) Distributional robustness and (2) Simplicity bias. We show how these properties lead to significantly different levels of collective success that are unexplained by prior work.

When the data is composed of multiple sub-populations, algorithms that optimise for average performance often perform poorly in minority sub-populations (Meinshausen & Bühlmann, 2015). A fairness-focused firm will want to ensure that their trained learning model performs well uniformly on all sub-populations, as opposed to being on-average good. Distributionally Robust Optimisation (DRO) is a family of algorithms designed to maximize this per-group accuracy (Hashimoto et al., 2018; Wang et al., 2020). We show that as a consequence, a small collective achieves higher success when the training algorithm performs DRO, compared to standard empirical risk minimization (ERM). Conversely, a large collective in the same settings achieves lower success, contradicting the expectation set by previous work regarding the effectiveness of large collectives.

Second, most machine learning algorithms used today are based on some form of gradient descent (GD). Such algorithms, including the popular Stochastic Gradient Descent (SGD), Adam (Kingma & Ba, 2015), and RMSProp, exhibit a preference for learning functions that are "simpler". This inductive bias of GD algorithms is popularly referred to as simplicity bias (Kalimeris et al., 2019; Shah et al., 2020). In practice, this preference results in the model "overlooking" certain complex features. We demonstrate that these overlooked features can be leveraged by a collective to design a strategy that will gain higher success

*Equal contribution [1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]Tübingen AI Center [3]Department of Statistics, University of Oxford. Correspondence to: Omri Ben-Dov <omri.ben-dov@tuebingen.mpg.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

compared to what is possible on a Bayes optimal classifier.

Our work initiates a study into an algorithm-dependent view on the success of collective action. We provide a theoretical foundation and empirical evidence to analyse the success of the collective for two important categories of algorithms: DRO and algorithms with simplicity bias (e.g. SGD).

2. Problem Formulation and Notation

In this section, we provide a formal discussion of both Collective Action and Distributionally Robust Optimisation in addition to defining the various notations that will be used throughout this manuscript.

2.1. Collective Action

While there are several goals in the domain of algorithmic collective action defined in Hardt et al. (2023), in this work we will focus on the goal of collectively *planting a signal* with a *features-label strategy*. We believe that other important goals, like *erasing signals*, can also benefit from the techniques and observations of this paper, but we leave the detailed study of such settings to future work.

Planting a signal Given a base distribution $\mathcal{P}_0$ on the domain of features and labels $\mathcal{X} \times \mathcal{Y}$, the classifier observes the mixture distribution

$$\mathcal{P}_\alpha = \alpha \mathcal{P}^* + (1-\alpha)\,\mathcal{P}_0, \tag{1}$$

where α is the collective's proportional size and $\mathcal{P}^*$ is the collective's distribution. The goal of the collective is to create an association in a classifier $f : \mathcal{X} \to \mathcal{Y}$, between a signal $g : \mathcal{X} \to \mathcal{X}$ and a label $y^* \in \mathcal{Y}$. Formally, the collective's goal is to maximize the success defined as

$$S(\alpha) = \mathcal{P}_0\left[f\left(g\left(x\right)\right) = y^*\right]. \tag{2}$$

The collective modifies their own data by planting a signal $(x, y) \to (g(x), y^*)$. $\mathcal{P}^*$ defines the distribution of $(g(x), y^*)$ where $x \sim \mathcal{P}_0$. This has been defined as the feature-label strategy in prior work. The signal g also defines the signal set $\mathcal{X}^* = \{g(x) \,|x \in \mathcal{X}\}$. For any distribution $\mathcal{P}$ over $X \times Y$, the Bayes optimal classifier on $\mathcal{P}$, which we denote as $f_{\mathcal{P}}$, is defined as

$$f_{\mathcal{P}}(x) = \arg\max_{y \in Y} \mathcal{P}\left(Y = y \mid X = x\right). \tag{3}$$

For Bayes (sub-)optimal classifiers, Hardt et al. (2023) identify four properties that affect the success of a collective action:

- The fractional size of the collective α (see Equation (1)) gives the collective greater statistical power. The larger the size, the higher the success.
- The uniqueness ξ of the signal. A signal is ξ-unique if $\mathcal{P}_0(\mathcal{X}^*) \leq \xi$. Informally, ξ is the measure of the codomain of the collective transformation g under the probability measure of the base distribution $\mathcal{P}_0$. The more unique the signal (smaller ξ), the easier it is to associate the signal to y^*, leading to higher success.
- The sub-optimality gap of the signal is defined as $\Delta = \max_{x \in \mathcal{X}^*} \max_{y \in \mathcal{Y}} \mathcal{P}_0(y|x) - \mathcal{P}_0(y^*|x)$. It measures the extent to which the collective competes with signals already present in $\mathcal{P}_0$. The smaller the sub-optimality gap, the higher the chances of success.
- The sub-optimality ϵ of a learned classifier relays how close a classifier is to the Bayes optimal. It is defined as the smallest total variation (TV) distance between $\mathcal{P}_\alpha$ and a distribution on which the learned classifier is actually Bayes optimal.

Using the above four properties, they derive the following lower bound on the success of the collective.

Theorem 1 (Theorem 1 in Hardt et al. (2023))**.** *Given the mixture distribution $\mathcal{P}_\alpha$ and the feature-label strategy for planting a signal, the success is lower bounded by*

$$S(\alpha) \geq 1 - \left(\frac{1-\alpha}{\alpha}\right)\Delta \cdot \xi - \frac{\epsilon}{1-\epsilon}, \tag{4}$$

where α, ξ, Δ, and ϵ are, respectively, the size, uniqueness, sub-optimality gap for y^* in the base distribution, and sub-optimality of the learned classifier on $\mathcal{P}_\alpha$.

While they are sufficient to characterise the success of the collective for a Bayes (sub-)optimal learner, various practically deployed algorithms show different behaviours, as is discussed in Section 3 and 4.

2.2. Distributionally Robust Optimisation

In Section 3, we inspect collective action on a set of learning algorithms that target *Distributionally-Robust* objectives (Delage & Ye, 2010; Sagawa et al., 2020; Duchi & Namkoong, 2021). Intuitively, these algorithms aim to learn classifiers that perform equally well on a set of distributions as opposed to any single one. Formally, they minimise the following objective

$$\mathcal{R}_{\text{dro}}(\theta) \coloneqq \sup_{q \in \mathcal{Q}_p} \mathbb{E}_q\left[\ell\left(g_\theta(x), y\right)\right], \tag{5}$$

where $\mathcal{Q}_p$ is an *uncertainty set* of distributions close to p over which we want to control the risk, ℓ is the loss function, and g_θ is a function with parameters θ. There are many possible definitions of the uncertainty set that the algorithm

is controlling for. One possible choice is an f-divergence[1] ball (Ali & Silvey, 1966; Csiszár, 1967).

$$\mathcal{Q}_p = \{q \ll p \mid \mathcal{D}_f(q \mid\mid p) \leq \delta\}, \tag{6}$$

where δ is the radius according to $\mathcal{D}_f$.[2] An often cited use-case of DRO algorithms is to protect the performance of small subgroups (Hashimoto et al., 2018).

Consider a set of subgroups or sub-populations denoted as $\mathcal{A}$ and let the observed distribution, p, arise as a mixture over the subgroups with distributions p_a as $p = \sum_{a\in\mathcal{A}} \alpha_a p_a$ so the uncertainty set then becomes

$$\mathcal{Q}_p = \left\{ \sum_{a\in\mathcal{A}} \beta_a p_a \mid \sum_{a\in\mathcal{A}} \beta_a = 1, \beta_a \geq 0 \right\}. \tag{7}$$

Under this setting, minimising $\mathcal{R}_{\text{dro}}(\theta)$ is equivalent to minimising the worst group loss (Sagawa et al., 2020)

$$\mathcal{R}_{\text{WGL}}(\theta) \coloneqq \max_{a\in\mathcal{A}} \mathbb{E}_q [\ell(g_\theta(x), y) \mid A = a]. \tag{8}$$

Thus, DRO algorithms are often employed when there is concern about performance on "similar" distributions or on subgroups of the data (Namkoong & Duchi, 2016; Duchi & Namkoong, 2019). In Section 3, we investigate how the success of the collective changes when minimising $\mathcal{R}_{\text{dro}}(\theta)$ as opposed to performing simple Empirical Risk Minimisation (ERM).

3. Effective Size and Validation Control

The most intuitive parameter to predict the success of collective action is the fractional size of the collective α; a larger collective will attain greater success. However, DRO algorithms assign different weights to different samples, rendering α inappropriate for predicting success. Instead, we introduce a correction to the collective size under a weighted distribution that we denote the effective size α_{eff}, and show that DRO algorithms can yield $\alpha_{\text{eff}} > \alpha$.

We experimentally validate this theory on a selection of two-stage re-weighting algorithms, specifically JTT (Liu et al., 2021) and LfF (Nam et al., 2020). Our findings, based on synthetic and image datasets, indicate that DRO algorithms can significantly increase collective success, surpassing standard Empirical Risk Minimization (ERM) for the same tasks.

Finally, we turn to iterative re-weighting algorithms, focusing on CVaR-DRO (Levy et al., 2020). Unlike two stage re-weighting algorithms, iterative re-weighting algorithms oscillate between fitting different parts of the data in training, relying on performance on a validation set as a stopping criterion. When the collective can influence both the training and validation set, we show that this stopping criteria makes them particularly sensitive to the collective's size in the validation set. To demonstrate this we first analyse an abstract theoretical version of CVaR-DRO, varying the degree of use of collective action in the validation set. Finally, we experimentally validate these claims, showing the collective success with CVaR-DRO is very sensitive to the collective proportion in the validation set.

3.1. Effective Collective Size

As mentioned, DRO algorithms allow for varying data points to have differing levels of impact on the algorithm by assigning different weights to the training data. Let $w : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a mapping from a sample to its weight. Those weights define the weighted distribution

$$\mathcal{P}^{(w)}(X = x, Y = y) = \frac{w(x,y)\,\mathcal{P}(X = x, Y = y)}{\mathbb{E}_{\mathcal{P}}[w(x,y)]}. \tag{9}$$

Within the context of collective action, this weighting can effectively boost or diminish the influence of the collective. To capture this change of impact, we introduce the following notion of effective collective size $\alpha_{\text{eff}}(w)$.

Definition 1. *For a distribution $\mathcal{P}^{(w)}$ where samples are up-weighted according to their covariates by $w(x)$ we define the effective collective size as*

$$\alpha_{\text{eff}}(w) = \frac{\mathbb{E}_{x,y\sim\mathcal{P}}[w(x,y)\,\mathbb{1}\{(x,y) \text{ is in the collective}\}]}{\mathbb{E}_{x,y\sim\mathcal{P}}[w(x,y)]}. \tag{10}$$

Note that if $w(x) = 1$ for all x, then $\alpha_{\text{eff}} = \alpha$. Now, since α_{eff} is the collective size under the weighted distribution, then bounding the success for this algorithm, akin to Theorem 1, requires adding a corrective term c for the non-weighted distribution, giving

$$S(\alpha) \geq 1 - \left(\frac{1-\alpha_{\text{eff}}}{\alpha_{\text{eff}}}\right)(\Delta\cdot\xi + c) - \frac{\epsilon}{1-\epsilon}, \tag{11}$$

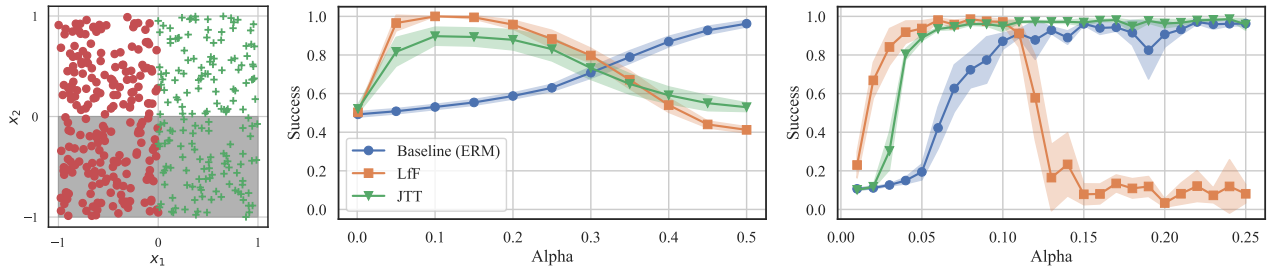
with the corrective term being

$$c = \mathbb{E}_{x\sim\mathcal{P}^*}\left[\frac{\Delta_x^w \mathcal{P}_0^{(w)}(x)}{(\mathcal{P}^*)^{(w)}(x)} - \frac{\Delta_x \mathcal{P}_0(x)}{\mathcal{P}(x)}\right], \tag{12}$$

where $\Delta_x^w = \max_{y\in Y}\left(\mathcal{P}^{(w)}(y \mid x) - \mathcal{P}^{(w)}(y^* \mid x)\right)$ and Δ_x is defined the same but for $\mathcal{P}_0$. Formal proof and definitions are in the appendix under Proposition B.3. Under certain circumstances we can have $c \leq 0$, implying a collective success greater than that guaranteed by Theorem 1 for $\alpha = \alpha_{\text{eff}}$. We provide such an example where $c \leq 0$ and $\alpha_{\text{eff}} \geq \alpha$ in Appendix B.1.

[1] Definition 2 in the appendix.

[2] The notation $q \ll p$ means q is absolutely continuous with respect to p.



(a) Synthetic 2D dataset (b) Success on synthetic 2D dataset (c) Success on CIFAR-10

Figure 1. Success with DRO algorithms. **(a)** An example of the 2D dataset. The color of each point represents its label, and the grey rectangle is the co-domain of the collective signal. **(b-c)** The success of a collective of different sizes α when trained with ERM (blue circles), JTT (orange squares) and LfF(green triangles) on a synthetic 2D and CIFAR-10.

We now provide theoretical results, showing that DRO algorithms can increase the effective collective size α_{eff}. Firstly, for any algorithm that targets a DRO defined by an f-divergence, we can say the following:

Proposition 3.1. *For a mixture distribution $\mathcal{P}_\alpha$, let $\mathcal{Q}_{\mathcal{P}_\alpha}$ be the set of distribution in a ball of radius δ around $\mathcal{P}_\alpha$ as defined in Equation (6). Then $\mathcal{Q}_{\mathcal{P}_\alpha}$ contains a distribution $\mathcal{P}_{\alpha_{eff}}$ with effective collective size*

$$\alpha_{eff} = \alpha + \frac{\delta}{\mathcal{D}_f\left(\mathcal{P}^* \| \mathcal{P}_\alpha\right)}. \tag{13}$$

The proof can be found in Appendix B. The δ parameter in this case is the radius of the ball that the algorithm is controlling performance over. It is either explicitly chosen, or implicitly defined by the hyper-parameters of the algorithm. This proposition tells us that if the algorithm is optimising against a wide range of distributions, this range will include a mixture distribution with a higher α_{eff}.

Now, we turn to analysing two-stage algorithms. In the first phase, these algorithms train a weak classifier, for example with early stopping or strong regularisation. Then, all samples in the error set of this classifier are up-weighted by a factor of λ for the second and final stage of training. This ensures an algorithm has good performance against the worst case subgroups in the original data. The following characterises the effective collective size for these algorithms.

Proposition 3.2. *[Effective Collective Size of JTT (Liu et al., 2021)] For JTT trained on $\mathcal{P}_\alpha$, let λ be the up-weighting parameter, f be the classifier learned in the first phase and define*

$$\begin{aligned} P_E &:= \mathcal{P}_\alpha\left[f(X) \neq Y\right] \text{ and} \\ P_{E|C} &:= \mathcal{P}_\alpha\left[f(X) \neq Y \mid (X,Y) \text{ in the collective}\right]. \end{aligned} \tag{14}$$

Then, the effective collective size is given by

$$\alpha_{eff} = \alpha \frac{\lambda P_{E|C} + \left(1 - P_{E|C}\right)}{\lambda P_E + (1 - P_E)}. \tag{15}$$

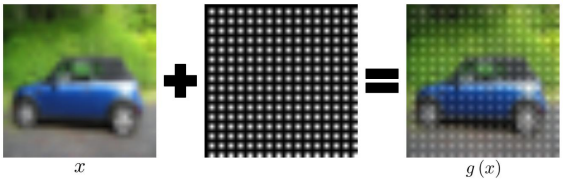


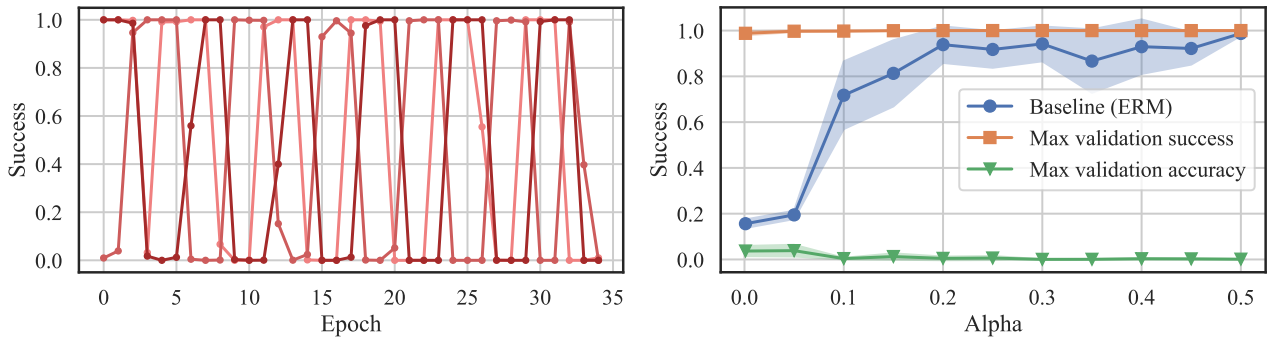
Figure 2. Image transformation used by the collective. The effect of the signal is exaggerated for visualisation purposes and in practice it is invisible to the human eye.

The proof can be found in Appendix B. This proposition demonstrates that if the first stage classifier f is more likely to make errors on the collective samples than on random samples ($P_{E|C} > P_E$), this leads to $\alpha_{\text{eff}} > \alpha$. This means that if the collective distribution represents a particularly challenging subgroup of the dataset, it will be up-weighted to have a much larger effect on the final output of these algorithms, which should lead to higher collective success.

3.2. Experiments Results for Two Stage Algorithms

We experimentally validate the above theory, showing that a collective can be more successful against JTT and LfF compared to ERM on a synthetic 2D dataset and CIFAR-10 (Krizhevsky, 2009). These algorithms are explained in Definition 3 and 4 in the appendix, and the synthetic 2D dataset comprises points sampled i.i.d. from a uniform distribution on $(-1,1)^2$, labeled by the sign of their first coordinate x_1 (Figure 1a). The collective wants the points with a negative x_2 to be labeled y^* and uses the strategy $\{(x_1, x_2), y\} \to \{(x_1, -|x_2|), y^*\}$. We also consider the multi-class classification problem of CIFAR-10. The collective transformation g, in the pixel space of integer values from 0 to 255, adds a perturbation of magnitude 2 to the value of every second pixel in every second row (Figure 2). This transform is virtually invisible to the human eye. Technical details can be found in Appendix C.

The results on the 2D dataset (Figure 1b) and on CIFAR-



(a) Success against time

(b) Success for different stopping conditions

Figure 3. Success sensitivity to stopping condition when using CVaR-DRO on the Waterbirds dataset with a collective of size $\alpha = 0.3$. **(a)** The success of the collective after every epoch of training. Each shade represents a differently initialised training. The rapid and sharp oscillations show how drastic it is to stop training at the right time. **(b)** The success achieved by different α for 2 different stopping conditions. The baseline (ERM) is shown in blue circles. Stopping at maximum accuracy in a validation set with no collective action is shown in the green triangles. Stopping at maximum success on the validation set is shown in orange triangles. When the firm is trying to maximize general accuracy, the collective has no success.

10 (Figure 1c) show that for small α, a collective achieves higher success in JTT and LfF than with the same α in ERM. Proposition 3.2 predicts this behavior for JTT. When α is small, the data bias causes ERM to misclassify collective members. This inaccuracy increases the collective population in the error set of first phase. This leads to a higher $P_{E|C}$ relatively to P_E (Equation (14)), which consequently leads to $\alpha_{\text{eff}} > \alpha$. This is no longer the case when α is large enough. As α rises, the accuracy of ERM on the collective increases, and the collective membership in the error set decreases. As a result, $P_{E|C} < P_E$ and $\alpha_{\text{eff}} < \alpha$, lowering the success. This effect starts at $\alpha \approx 0.2$ on the 2D dataset and at $\alpha \approx 0.1$ on CIFAR-10.

Intuitively, the goal of JTT and LfF is to empower weak groups. Accordingly, these algorithms give a small collective more statistical power, which grants the collective a higher success than they would achieve in ERM. When the collective is large, these algorithms will take the power away, lowering the success. This teaches us that a large collective should alter its strategy in order to maximize success.

3.3. Iterative Re-weighting Algorithms and Collective Action on Validation Sets

We now turn to analyse algorithms that iteratively re-weight, focusing on CVaR-DRO. For these algorithms, the effective distribution at each step is chosen adversarially from the uncertainty set of distributions as discussed in Section 2.2. We explain the CVaR-DRO algorithm in definition 5 in the appendix. This causes the algorithm to cycle between fitting different parts of the distribution, terminating only when a high enough accuracy is reached on some validation set. As the validation set plays an important role in deciding when the algorithm terminates, we consider how varying α in the validation set can affect the collective success.

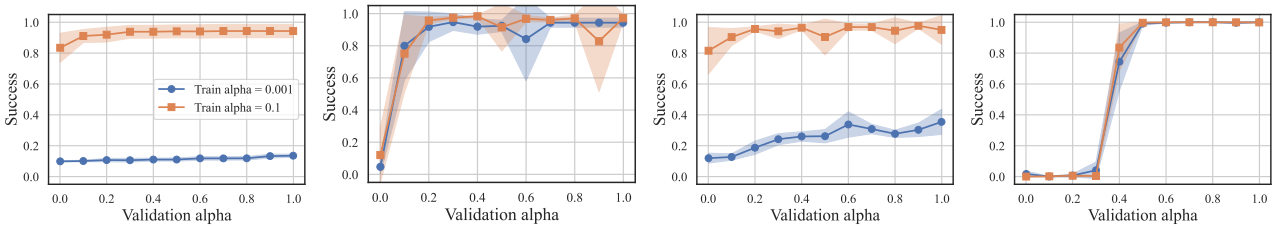
Theory In order to theoretically analyse the effect of the validation set on the final success of the collective, we look at an idealised version of a iterative re-weighting DRO algorithm. This ideal algorithm computes a sequence of classifiers $\mathcal{F}$, where each classifier $f^{(i)}$ is a Bayes optimal classifier for a distribution on which the previous classifier $f^{(i-1)}$ has the maximum error, starting with $\mathcal{P}_0$. The algorithm outputs the classifier $f \in \mathcal{F}$ that has the highest accuracy on the validation distribution affected by collective action, which is given by $\mathcal{P}_V = \beta P^* + (1-\beta)\mathcal{P}_0$, where β is the collective size in the validation set. A precise form of this algorithm is given in Algorithm 3 in the appendix.

Proposition 3.3. *Let f be the output of Algorithm 3, and $f_{\mathcal{P}_\alpha}$ be the Bayes optimal classifier on the mixture distribution $\mathcal{P}_\alpha$. Then we have that the success S_f and $S_{f_{\mathcal{P}_\alpha}}$ with f and $f_{\mathcal{P}_\alpha}$, respectively, relate as*

$$S_f - S_{f_{\mathcal{P}_\alpha}} \geq \frac{\mathcal{P}_V\left[f\left(X\right) = Y\right] - \mathcal{P}_V\left[f_{\mathcal{P}_\alpha}\left(X\right) = Y\right]}{\beta - \alpha}. \tag{16}$$

The proof is in Appendix B. If β is close to α, $f_{\mathcal{P}_\alpha}$ must still be Bayes optimal on $\mathcal{P}_V$ and so we have that S_f is not provably greater than $S_{f_{\mathcal{P}_\alpha}}$. However, as β increases, f will have better accuracy on the validation set than $f_{\mathcal{P}_\alpha}$ that was trained on the train set. In such a case, $\mathcal{P}_V\left[f\left(X\right)=Y\right] > \mathcal{P}_V\left[f_{\mathcal{P}_\alpha}\left(X\right)=Y\right]$, which leads to a strictly higher collective success from our idealised CVaR-DRO algorithm when compared to the Bayes optimal classifier on the training distribution.

Experimental Results Figure 3a demonstrates the oscillating success after every epoch when training CVaR-DRO



(a) ERM on CIFAR-10 (b) CVaR-DRO on CIFAR-10 (c) ERM on Waterbirds (d) CVaR-DRO on Waterbirds

Figure 4. Each graph shows the success of different levels on collective action in the validation set α_{val} when using ERM or CVaR-DRO on CIFAR-10 and Waterbirds. The blue circles are for a training set with $\alpha_{\text{train}} = 0.001$ and the orange squares are for $\alpha_{\text{train}} = 0.1$. ERM is almost not affected from α_{val}, but for CVaR-DRO α_{val} is crucial.

on the Waterbirds dataset (Sagawa et al., 2020) with α=0.3. These oscillations are due to the search over the distributions space $\mathcal{Q}_{\mathcal{P}_\alpha}$(Equation (6)). Proposition 3.1 predicts that this space contains a distribution where the collective is stronger than in the observed distribution. The peaks of the oscillations suggests that CVaR-DRO was able to find such a distribution. The frequency and amplitude of the oscillations show that the resulting success is sensitive to when training stops, varying from minimal to maximal success.

Realistically, training stops according to some condition on a validation set. For example, the firm can choose to stop training at the iteration that achieves the highest accuracy on the validation set. In Figure 3b we compare the success on ERM with the success on CVaR-DRO with different conditions on a collective-free validation set: maximum accuracy and maximum collective success. This comparison shows that when there is no collective action in the validation set, maximum validation accuracy cancels possible success. The reason that the success is 0 is because there is no collective signal in the validation set.

If the collective is able to affect the validation set, then increasing its proportion in the validation set will increase success, as suggested in Proposition 3.3. We show this experimentally by applying collective action in both training and validation set of the CIFAR-10 and Waterbirds datasets. Figure 4b and 4d show that when CVaR-DRO is used, the amount of collective in the validation set has a very large impact on the collective success compared to the collective size in the training set. In contrast, Figure 4a and 4c show that success in ERM is more sensitive to the collective size in the training set α rather than to the size in the validation set. With $\alpha = 0.1$ in ERM, the collective achieves almost full success, while when $\alpha = 0.001$, even with full control of the validations set, the success does not go over 0.4.

4. Leveraging Algorithmic Bias

In the previous section we described how different learning algorithms can modify the effective size of the collective. However, these results do not provide guidelines for designing a more effective signal g. In this section we address this gap by leveraging insights on the biases of the learning algorithm with regard to the properties of the base distribution $\mathcal{P}_0$. This bias deviates the resulting classifier from the Bayes optimal classifier discussed in Hardt et al. (2023), but in a way that allows the collective to take advantage of. Essentially, if the learning algorithm overlooks certain signals in the base distribution, it is akin to eliminating these signals from the distribution, rendering them unique.

4.1. Theoretical Results

We provide the following theoretical result to support this claim. First, for an arbitrary distribution $\mathcal{Q}_0$ we define the Bayes-optimal classifier on that distribution

$$f_{\mathcal{Q}_0}(x) \coloneqq \arg\max_{y\in\mathcal{Y}} \mathcal{Q}_0\left(Y = y | X = x\right), \tag{17}$$

and the $\mathcal{Q}_\alpha$-mixture distribution as

$$\mathcal{Q}_\alpha \coloneqq \alpha\mathcal{P}^* + (1-\alpha)\,\mathcal{Q}_0. \tag{18}$$

Theorem 2. *Consider a base distribution $\mathcal{P}_0$ and a learning algorithm $\mathcal{A}$ which outputs $h_{\mathcal{P}_\alpha}$ when learning on $\mathcal{P}_\alpha$. For any given distribution $\mathcal{Q}_0$, we denote the corresponding classifier TV distance as*

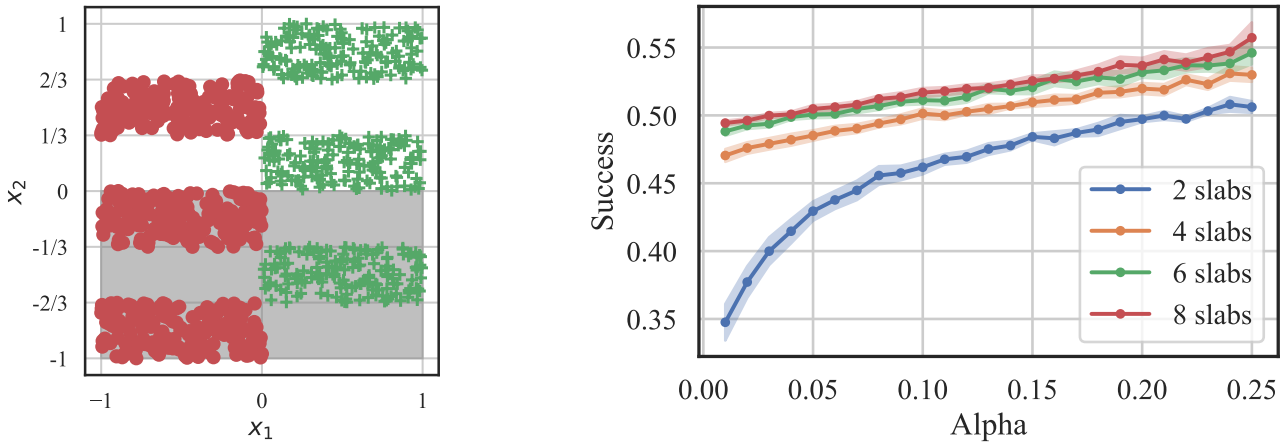
$$\omega_{\mathcal{Q}_0} = \text{TV}\left(\mathcal{P}^*(X) \times h_{\mathcal{P}_\alpha}(X) \,|\, \mathcal{P}^*(X) \times f_{\mathcal{Q}_\alpha}(X)\right). \tag{19}$$

Then the collective success of algorithm $\mathcal{A}$ on $\mathcal{P}_\alpha$, is bounded below as

$$S(\alpha) \geq \sup_{\mathcal{Q}_0} \left\{1 - \frac{\omega_{\mathcal{Q}_0}}{1-\omega_{\mathcal{Q}_0}} - \frac{1-\alpha}{\alpha}\mathcal{Q}_0(\mathcal{X}^*)\Delta_{\mathcal{Q}_0}\right\}, \tag{20}$$

where $\Delta_{\mathcal{Q}_0} = \max_{x\in\mathcal{X}^*}\max_{y\in\mathcal{Y}} \mathcal{Q}_0(y|x) - \mathcal{Q}_0(y^*|x)$.

The proof is given in Appendix B.3. Note that the bound in theorem 2 cannot be smaller than the bound in theorem 1, since $\mathcal{Q}_0$=$\mathcal{P}_0$ recovers the original bound in Equation (4).



(a) LMS-6 dataset

(b) Success on LMS-k

Figure 5. Results on LMS-k. **(a)** an example of LMS-6. The color of each point represents its label, the grey rectangle is the codomain of the collective signal. **(b)** Success over LMS-k. Larger complexity (k) increases the success.

A toy example where the bound is strictly tighter can be found in Appendix B.4.

Intuitively, if a learning algorithm ignores "complicated" features, then one can conceptualise a surrogate distribution $\mathcal{Q}_0$ that is devoid of those "complicated" features. The corresponding mixture distribution is then defined as $\mathcal{Q}_\alpha = \alpha\mathcal{P}^* + (1-\alpha)\,\mathcal{Q}_0$. Within this surrogate distribution, the signal g is $\xi_{\mathcal{Q}_0}$-unique with a smaller $\xi_{\mathcal{Q}_0} < \xi$ uniqueness parameter. As $\mathcal{Q}_0$ and $\mathcal{P}_0$ share the same features that are relevant to the learned classifier, the Bayes optimal classifier on $\mathcal{Q}_\alpha$ is likely to closely resemble the learnt classifier on the original distribution $\mathcal{P}_\alpha$. This similarity effectively causes the signal to be $\xi_{\mathcal{Q}_0}$-unique on $\mathcal{P}_\alpha$ as well.

4.2. Experimental Results With Simplicity Bias

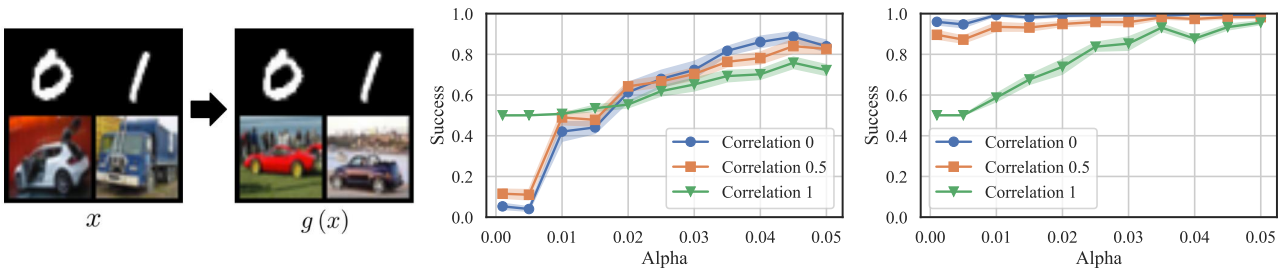
In the experiments, we focus on SGD, which is not only ubiquitous, but also has a bias towards "simple" features (Shah et al., 2020). We demonstrate our theory with three different experiments, each showing a different approach for constructing the collective signal g. In all these examples, we train the models using stochastic gradient descent (SGD) (more details in Appendix C), leveraging its known preference towards learning simpler features first.

Collective action on a complex feature The first approach we explore involves the collective embedding its signal within a complex feature. We illustrate this approach on a dataset similar to LMS-k from (Shah et al., 2020), shown in Figure 5a. In this dataset we consider a two-dimensional binary classification problem with variables x_1, x_2 representing the two dimensions. The dataset comprises two blocks along x_1, and k blocks along x_2. The classification label is primarily determined by a single threshold function on x_1, but can also be inferred using multiple thresholds on x_2. As noted by Shah et al. (2020), with an increasing number of blocks, models trained using SGD tend to increasingly disregard the x_2 feature, classifying by x_1 alone.

Here, the collective's goal is to classify samples with $x_2 < 0$ as $y^* = 1$. Figure 5b shows that the collective is more successful as x_2 becomes increasingly complex with higher values of $k \in \{2, 4, 6, 8\}$. Theorem 2 captures this intuition: As k grows, an SGD-based algorithm $h = \mathcal{A}(\mathcal{P}_\alpha)$ tends to learn the simpler x_1, becoming oblivious to variations in x_2. In turn, h becomes more similar to a Bayes optimal classifier on a distribution $\mathcal{Q}_0$ which is spurious on x_2, giving $\mathcal{Q}_0$ a smaller $\omega_{\mathcal{Q}_0}$, and therefore higher success for larger k.

When a simpler feature is less informative In the previous example, we observed that when both a simple and a complex feature are fully correlated with the label, the collective's success increases as the gap in the simplicity between these features widens. However, this is not always reflected in practice where the simpler feature may not fully correlate with the label.

One such example is the presence of spurious correlations in the dataset. To demonstrate this, we use another dataset, derived from the MNIST-CIFAR dataset in Shah et al. (2020). Each data point in this dataset comprises a pair of images: one from MNIST (zero or one) and one from CIFAR (truck or automobile), as illustrated in the left side of Figure 6a. The CIFAR image determines the label. We then adjust the correlation level between the MNIST image and the label. A correlation of one implies that automobiles always pair with MNIST zero, and trucks with MNIST one. A zero correlation indicates random pairing of MNIST digits, and a 0.5 correlation suggests that the MNIST image is randomly



(a) Collective signal (b) On Class Automobile (c) On Class Truck

Figure 6. Collective success on *correlated* MNIST-CIFAR **(a)** An automobile and a truck from MNIST-CIFAR and the transformation that changes the CIFAR part to automobile. **(b-c)** Success with $y^* =$ truck when the signal is (b) an automobile or (c) a truck . Blue circles are for no MNIST-to-label correlation, orange square are for 0.5 correlation and green triangles are for full correlation.

sampled half of the time and correlates with the label for the other half. In this experiment, the collective's strategy involves embedding a signal in the CIFAR component of the dataset, representing either an automobile or a truck, with the designated label y^*=truck.[3] Figure 6a shows a transformation using automobile pictures. The full dataset comprises 5000 images of trucks and automobiles, but the collective is restricted to only plant a randomly selected subset of a 100 of these images.

Our results in Figure 6b and 6c show that as the correlation decreases, the success increases. The underlying intuition is that with higher correlation the simpler MNIST part alone can yield high training performance. As a result, SGD tends to ignore the CIFAR part, where the signal acts. In contrast, a weaker correlation between the MNIST part and the label necessitates the algorithm's reliance on the CIFAR part, leading it to also learn the collective signal embedded therein. Note that for small α in Figure 6b, the success with non-zero correlation is low because the collective is labeling automobiles as y^*=truck, resulting in competing signals.

This effect can be explained by Theorem 2 as follows. First, note that unlike the analysis in the previous section, in this case it is not possible to design $\mathcal{Q}_0$ that is significantly different from $\mathcal{P}_0$ in the complex feature (CIFAR) while maintaining a small $\omega_{\mathcal{Q}_0}$. This difficulty arises because with weak correlation between the MNIST part and the label, the CIFAR part becomes the primary source of the label information. Consequently, $\mathcal{Q}_0$ must closely resemble $\mathcal{P}_0$ in the CIFAR part. Instead by choosing the surrogate distribution $\mathcal{Q}_0$ to be one where the MNIST part contains relatively small amount of information about the label, we can minimise $\omega_{\mathcal{Q}_0}$ while also keeping $\mathcal{Q}_0(\mathcal{X}^*)\Delta_{\mathcal{Q}_0}$ small. As the correlation increases, this will not be possible and we see a drop in the success of the collective.

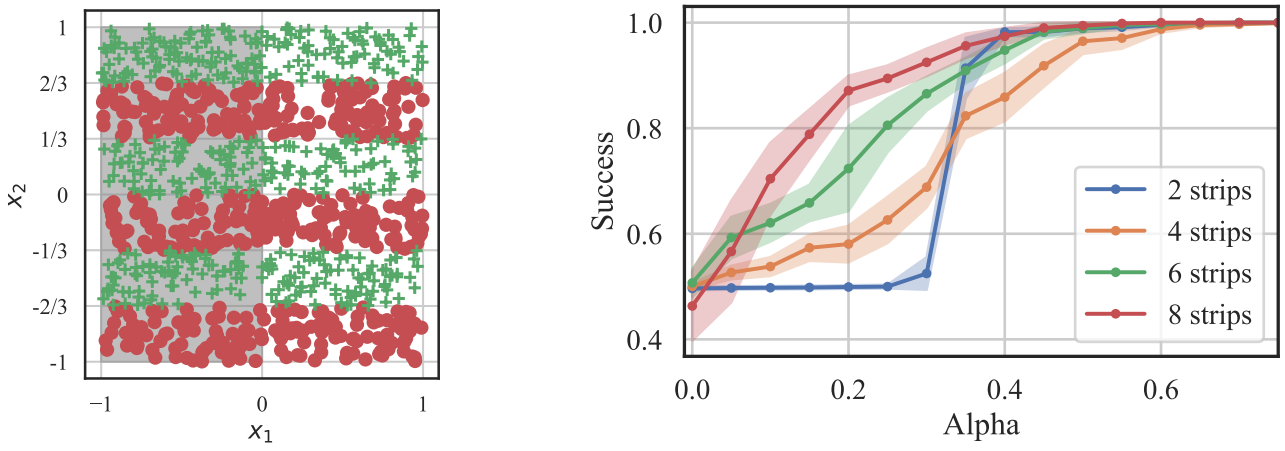
[3]Embedding signals in the complex feature is not necessarily the optimal approach in scenarios where the simpler feature is not predictive of the label.

Collective action on simpler feature Finally, we demonstrate that when the simpler feature is uncorrelated with the label, the more effective approach is to simply plant the signal in the simpler feature. This is especially true for algorithms that exhibit simplicity bias and use early stopping or strong regularisation, which are common practices in machine learning. Intuitively, if the learning algorithm on $\mathcal{P}_\alpha$ is stopped early, it is unlikely to have learned all the correlations present in the data, resulting in a sub-optimal model. However, it will have captured more of the simpler feature than the complex feature. Thus, if the collective's signal aligns with the simpler feature, this leads to greater success because the sub-optimal aspects of the classifier will be concentrated in areas outside the collective's signal set.

Remark 4.1. Note that this phenomenon is not captured by Hardt et al. (2023), where the sub-optimality ϵ is always considered to lie within collective's signal set.

To demonstrate how a collective can gain from planting a signal in the simpler feature, we use a new dataset, named k-strips, which is similar to LMS-k, but removes the correlation between x_1 and the label (Figure 7a). In this experiment, we create the synthetic dataset by sampling points from a uniform distribution on $(0,1)^2$, labeled by their x_2 values. The x_2-axis is divided into n horizontal strips, and a point has a positive label if it's in an even-numbered strip, and a negative label if it's in an odd-numbered strip.

The collective attempts to focus the attention on x_1 by giving positive labels if $x_1 < 0$. Figure 7b shows that with more horizontal strips, the collective attains higher success with a smaller strength α. This result is predicted by Theorem 2: As k grows, an SGD-based early-stopped algorithm on the collective $\mathcal{A}(\mathcal{P}_\alpha)$ will tend to learn the simpler x_1 feature while largely overlooking x_2. If $X' \subset \mathbb{R}^2$ is the part of the domain where $\mathcal{A}(\mathcal{P}_\alpha)$ accurately predicts on $\mathcal{P}_\alpha$, a distribution $\mathcal{Q}_0$ can place the majority of its probability mass on a thin horizontal strip and the remaining probability mass on X'. This will be sufficient to both get a small $\omega_{\mathcal{Q}_0}$ due to the small mass on X', and a small $\mathcal{Q}_0(\mathcal{X}^*)\Delta_{\mathcal{Q}_0}$



(a) 6-strips dataset

(b) Success of on k-strips

Figure 7. Results on k-strips. **(a)** Example of 6-strips. The color of each point represents its label, the grey rectangle is the codomain of the collective signal. **(b)** Success for different α and k.

due to only small part of x_2 having any probability mass under $\mathcal{Q}_0$. This leads to a higher lower bound in Theorem 2, thereby explaining the observations in Figure 7b.

5. Conclusion and Future Work

In this work we conducted a theoretical and empirical study on how the choice of learning algorithms affects the success of a collective action to plant a signal. We presented various approaches a collective can take to design their signal when they have knowledge about the learning algorithm. In particular, these approaches include maintaining a small collective size against algorithms like JTT and LfF, influencing the validation set against CVaR-DRO, and changing the complexity of the signal against SGD. While we have focused on these three algorithms, these phenomena are relevant for several popular algorithms. Common algorithms in the topic of domain adaptation (Koh et al., 2021), distribution shift (Hendrycks & Dietterich, 2019), and improving fairness (Berk et al., 2017) use some form of DRO algorithms. It would be interesting to consider the impact of other kinds of algorithms used to improve worst group performance including un-supervsied and self-supervised representation learning (Shi et al., 2023) and fairness-inducing in-processing (Prost et al., 2019) and post-processing algorithms (Ţifrea et al., 2024). In a similar vein, recent research has seen a surge in algorithms that aim to improve *safety* of outputs of generative models including techniques like adversarial training (Ziegler et al., 2022), DPO (Rafailov et al., 2024), and RLHF (Christiano et al., 2017). It is important to also consider the impact of these algorithms on the success of a well-designed collective.

Algorithmic biases akin to simplicity bias are also exhibited by SGD training of various state-of-the art deep neural networks. This includes a texture bias for CNN (Hermann et al., 2020), an in-context bias for language models (Levine et al., 2022), and word-order biases in LSTMs and transformers (White & Cotterell, 2021). Our work discusses how these biases can affect the design of the collective signal and how successful they are expected to be. However, several algorithms also display a different kind of biases. For example, differentially private algorithms are unable to learn minority sub-populations (Bagdasaryan et al., 2019; Sanyal et al., 2022) and adversarially robust algorithms (Madry et al., 2018) are used to improve the smoothness of learned models. One direction for future work is to consider the impact of these algorithmic biases on the success of the collective.

Finally, a third important direction of future research is understanding what level of information and influence is required by the collective to design their signal. For example, our results suggest that improving success against certain DRO algorithms requires less information about the learning algorithm compared to success against SGD style algorithms. However, for iterative re-weighting algorithms like CVar-DRO (Sagawa et al., 2020), it is important to influence both the validation set and the training set. Future work should investigate whether a uniformly randomly selected collective can be as powerful as a strategically chosen collective (e.g. in adversarial vulnerability (Paleka & Sanyal, 2023)). We hope that our work will inspire further research into practical and algorithm-dependent view on collective action.

Acknowledgements

The authors thank Alexandru Tifrea for helpful discussions and suggestions. OB is supported by the International Max Planck Research School for Intelligent Systems (IMPRS-IS). JF is funded by the ESRPC.

Impact Statement

In this work, we show how a group of people can take advantage of inductive biases in order to obtain statistical significance. This statistical significance can indirectly give power to the collective over the machine learning model. While we imagine the collective working together to obtain results that will contribute to the greater good, we acknowledge that those results can also serve a malicious group for self-gain.

References

Ali, S. M. and Silvey, S. D. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1966.

Bagdasaryan, E., Poursaeed, O., and Shmatikov, V. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 2019.

Berk, R., Heidari, H., Jabbari, S., Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. A convex framework for fair regression. *arXiv:1706.02409*, 2017.

Burrell, J., Kahn, Z., Jonas, A., and Griffin, D. When users control the algorithms: Values expressed in practices on twitter. In *Proceedings of the ACM on Human-Computer Interaction*, 2019.

Chen, J. Y. Thrown under the bus and outrunning it! the logic of didi and taxi drivers' labour and activism in the on-demand economy. *New Media & Society*, 2018.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 2017.

Csiszár, I. On information-type measure of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.*, 1967.

Delage, E. and Ye, Y. Distributionally robust optimization under moment uncertainty with application to datadrivenproblems. *Operations Research*, 2010.

Duchi, J. and Namkoong, H. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 2019.

Duchi, J. C. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 2021.

Gerlitz, C. and Helmond, A. The like economy: Social buttons and the data-intensive web. *New media & society*, 2013.

Hardt, M., Mazumdar, E., Mendler-Dünner, C., and Zrnic, T. Algorithmic collective action in machine learning. In *International Conference on Machine Learning*, volume 2022, 2023.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2018.

Hendrycks, D. and Dietterich, T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, 2019.

Hermann, K., Chen, T., and Kornblith, S. The origins and prevalence of texture bias in convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2020.

Kalimeris, D., Kaplun, G., Nakkiran, P., Edelman, B., Yang, T., Barak, B., and Zhang, H. Sgd on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems*, 2019.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, 2021.

Krizhevsky, A. Learning multiple layers of features from tiny images, 2009.

Levine, Y., Wies, N., Jannai, D., Navon, D., Hoshen, Y., and Shashua, A. The inductive bias of in-context learning: Rethinking pretraining example design. In *International Conference on Learning Representations*, 2022.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. Large-scale methods for distributionally robust optimization. In *Advances in Neural Information Processing Systems*, 2020.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Meinshausen, N. and Bühlmann, P. Maximin effects in inhomogeneous large-scale data. *The Annals of Statistics*, 2015.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: De-biasing classifier from biased classifier. In *Advances in Neural Information Processing Systems*, 2020.

Namkoong, H. and Duchi, J. C. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems*, 2016.

Olson, M. The logic of collective action. *Contemporary Sociological Theory*, 1965.

Paleka, D. and Sanyal, A. A law of adversarial risk, interpolation, and label noise. In *International Conference on Learning Representations*, 2023.

Prost, F., Qian, H., Chen, Q., Chi, E. H., Chen, J., and Beutel, A. Toward a better trade-off between performance and fairness with kernel-based distribution matching. *arXiv:1910.11779*, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 2024.

Rahman, H. A. The invisible cage: Workers' reactivity to opaque algorithmic evaluations. *Administrative Science Quarterly*, 2021.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*, 2020.

Sanyal, A., Hu, Y., and Yang, F. How unfair is private learning? In *Uncertainty in Artificial Intelligence*, 2022.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, 2020.

Shi, Y., Daunhawer, I., Vogt, J. E., Torr, P., and Sanyal, A. How robust is unsupervised representation learning to distribution shift? In *The Eleventh International Conference on Learning Representations*, 2023.

Ţifrea, A., Lahoti, P., Packer, B., Halpern, Y., Beirami, A., and Prost, F. Frapp\'e: A post-processing framework for group fairness regularization. In *International Conference on Machine Learning*, 2024.

Wang, S., Guo, W., Narasimhan, H., Cotter, A., Gupta, M., and Jordan, M. Robust optimization for fairness with noisy protected groups. In *Advances in Neural Information Processing Systems*, 2020.

White, J. C. and Cotterell, R. Examining the inductive bias of neural language models with artificial languages. In *Annual Meeting of the Association for Computational Linguistics*, 2021.

Ziegler, D., Nix, S., Chan, L., Bauman, T., Schmidt-Nielsen, P., Lin, T., Scherlis, A., Nabeshima, N., Weinstein-Raun, B., de Haas, D., et al. Adversarial training for high-stakes reliability. *Advances in Neural Information Processing Systems*, 2022.

A. Definitions and Algorithms

Definition 2. *f-divergence is a function that measures the difference between two distribution P and Q. For a given convex function $f:[0,+\infty)\to(-\infty,+\infty]$ such that $f(x)$ is finite for all $x>0$, with $f(1)=0$ and $f(0)=\lim_{t\to 0^+} f(t)$, the f-divergence is defined as*

$$D_f(P\|Q)=\int_\Omega f\left(\frac{dP}{dQ}\right)dQ.$$

Definition 3. *JJT, as defined by Liu et al. (2021), has two stages. The first stage trains a classifier by ERM. The second stage trains a different classifier on a similar dataset, where each sample that was misclassified by the classifier from the first stage, is given a higher weight.*

Algorithm 1 JTT training

Input: Training set $\mathcal{D}$ and hyperparameters T and λ_{up}.
Stage one: identification
1. Train $\hat{f}_{\text{id}}$ on $\mathcal{D}$ for T steps.
2. Construct the error set E of training examples misclassified by $\hat{f}_{\text{id}}$.
Stage two: upweighting identified points
3. Construct upsampled dataset $\mathcal{D}_{\text{up}}$ containing examples in the error set λ_{up} times and all other examples once.
4. Train final model $\hat{f}_{\text{final}}$ on $\mathcal{D}_{\text{up}}$.

Definition 4. *LfF, as defined by Nam et al. (2020), simultaneously trains 2 models: a biased classifier f_B, and a de-biased classifier f_D. The biased model is encouraged to learn biases by using a generalized cross entropy (GCE) loss, and the de-biased model is trained by giving more weight to samples that the biased model fails on. Where CE stands for the cross*

Algorithm 2 Learning from Failure

Input: θ_B,θ_D, training set $\mathcal{D}$, learning rate η, number of iterations T
Initialize two networks $f_B(x;\theta_B)$ and $f_D(x;\theta_D)$.
for $t=1,\cdots,T$ **do**
  Draw a mini-batch $\mathcal{B}=\{(x^{(b)},y^{(b)})\}_{b=1}^{B}$ from $\mathcal{D}$
  Update $f_B(x;\theta_B)$ by $\theta_B\leftarrow\theta_B-\eta\nabla_{\theta_B}\sum_{(x,y)\in\mathcal{B}}\text{GCE}(f_B(x),y)$.
  Update $f_D(x;\theta_D)$ by $\theta_D\leftarrow\theta_D-\eta\nabla_{\theta_D}\sum_{(x,y)\in\mathcal{B}}\mathcal{W}(x)\cdot\text{CE}(f_D(x),y)$.
end for

entropy loss, and GCE with hyperparameter q is defined as

$$GCE\left(p\left(f\right),y\right)=\frac{1-\left(f\left(y\right)\right)^q}{q},$$

where $f(y)$ is the probability of label y after a softmax layer. The weight $\mathcal{W}$ per sample is defined as

$$\mathcal{W}(x)=\frac{CE\left(f_B\left(x\right),y\right)}{CE\left(f_B\left(x\right),y\right)+CE\left(f_D\left(x\right),y\right)}.$$

Definition 5. *CVaR-DRO, as defined by Levy et al. (2020), dynamically changes the weights of samples according to their loss at every iteration. After the loss is computed for every sample in the mini-batch, and the samples with the smallest are ignored (given a 0 weight) in the current update step.*

B. Theoretical Results

In this section, we provide theoretical proofs for results stated in the main text.

B.1. Impact of α_{eff}

Proposition B.1. *Suppose we have a set of weights $w:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^+$, then we have that:*

$$\mathcal{P}_\alpha^{(w)}=\alpha_{\text{eff}}\left(\mathcal{P}^*\right)^{(w)}+(1-\alpha_{\text{eff}})\mathcal{P}_0^{(w)}$$

Proof. We have that:

$$\begin{aligned}
\Pr_{\mathcal{P}_\alpha^{(w)}}(X=x,Y=y) &= \frac{w(x,y)\left(\alpha\Pr_{(\mathcal{P}^*)^{(w)}}(X=x,Y=y)+(1-\alpha)\Pr_{\mathcal{P}_0^{(w)}}(X=x,Y=y)\right)}{\mathbb{E}_{\mathcal{P}_\alpha}[w(x,y)]} \\
&= \frac{\alpha\mathbb{E}_{\mathcal{P}^*}[w(x,y)]\Pr_{(\mathcal{P}^*)^{(w)}}(X=x,Y=y)+(1-\alpha)\mathbb{E}_{\mathcal{P}_0}[w(x,y)]\Pr_{\mathcal{P}_0^{(w)}}(X=x,Y=y)}{\mathbb{E}_{\mathcal{P}_\alpha}[w(x,y)]} \\
&= \frac{\alpha\mathbb{E}_{\mathcal{P}^*}[w(x,y)]}{\mathbb{E}_{\mathcal{P}_\alpha}[w(x,y)]}\Pr_{(\mathcal{P}^*)^{(w)}}(X=x,Y=y)+\frac{(1-\alpha)\mathbb{E}_{\mathcal{P}_0}[w(x,y)]}{\mathbb{E}_{\mathcal{P}_\alpha}[w(x,y)]}\Pr_{\mathcal{P}_0^{(w)}}(X=x,Y=y) \\
&= \alpha_{\text{eff}}\,(\mathcal{P}^*)^{(w)}+(1-\alpha_{\text{eff}})\mathcal{P}_0^{(w)}
\end{aligned}$$

Where the final line follows as we have $\frac{\alpha\mathbb{E}_{\mathcal{P}^*}[w(x,y)]}{\mathbb{E}_{\mathcal{P}_\alpha}[w(x,y)]} = \frac{\mathbb{E}_{\mathcal{P}_\alpha}[w(x,y)\mathbb{1}\,(\text{Sample from collective})]}{\mathbb{E}_{\mathcal{P}_\alpha}[w(x,y)]} = \alpha_{\text{eff}}$. □

Corollary B.2. *For any $x \in \mathcal{X}$ we have $f(x) = y^*$ if:*

$$\alpha_{\mathit{eff}} > (1-\alpha_{\mathit{eff}})\frac{\Delta_x^w\mathcal{P}_0^{(w)}(x)}{(\mathcal{P}^*)^{(w)}(x)}$$

Proof. This follows from the same argument as Hardt et al. (2023) where we now use the weighted distributions. □

Proposition B.3. *Let:*

$$c = \mathbb{E}_{\mathcal{P}^*}\left[\frac{\Delta_x^w\mathcal{P}_0^{(w)}(x)}{(\mathcal{P}^*)^{(w)}(x)}\right] - \mathbb{E}_{\mathcal{P}^*}\left[\frac{\Delta_x\mathcal{P}_0(x)}{\mathcal{P}^*(x)}\right]$$

Where $\Delta_x = \max_y\left(\mathcal{P}_0(y\mid x)-\mathcal{P}_0(y^*\mid x)\right)$ and $\Delta_x^w = \max_y\left(\mathcal{P}_0^{(w)}(y\mid x)-\mathcal{P}_0^{(w)}(y^*\mid x)\right)$.Then we have:

$$S(\alpha) \geq 1-\left(\frac{1-\alpha_{\text{eff}}}{\alpha_{\text{eff}}}\right)(\Delta\cdot\xi+c)-\frac{\epsilon}{1-\epsilon}, \tag{21}$$

Moreover, if we have that:

$$w(x,y') \leq \frac{\Delta_x E_{x,y\sim\mathcal{P}^0}[w(x,y)]}{\Delta_x^w E_{x,y\sim\mathcal{P}^\star}[w(x,y)]}w(x,y^\star)$$

Then $c \leq 0$ so that:

$$S(\alpha) \geq 1-\left(\frac{1-\alpha_{\text{eff}}}{\alpha_{\text{eff}}}\right)\Delta\cdot\xi-\frac{\epsilon}{1-\epsilon}, \tag{22}$$

Proof. First, for any $x \in \mathcal{X}$ we have $f(x) = y^*$ if:

$$\alpha_{\text{eff}} > (1-\alpha_{\text{eff}})\frac{\Delta_x^w\mathcal{P}_0^{(w)}(x)}{(\mathcal{P}^*)^{(w)}(x)}$$

Now, as in the original proof of (Hardt et al., 2023), we have that if the classifier is Bayes optimal on $\mathcal{P}_\alpha$:

$$\begin{aligned}
S(\alpha) &\geq 1-\frac{1-\alpha_{\text{eff}}}{\alpha_{\text{eff}}}\mathbb{E}_{x\sim\mathcal{P}^*}\left[\frac{\Delta_x^w\mathcal{P}_0^{(w)}(x)}{(\mathcal{P}^*)^{(w)}(x)}\right] &(23)\\
&= 1-\frac{1-\alpha_{\text{eff}}}{\alpha_{\text{eff}}}\mathbb{E}_{x\sim\mathcal{P}^*}\left[\frac{\Delta_x\mathcal{P}_0(x)}{\mathcal{P}^*(x)}\right]+\frac{(1-\alpha_{\text{eff}})\,c}{\alpha_{\text{eff}}} &(24)
\end{aligned}$$

Now, we have that c can be written as:

$$c = \mathbb{E}_{\mathcal{P}^*}\left[\frac{\Delta_x^w \mathcal{P}_0^{(w)}(x)}{(\mathcal{P}^*)^{(w)}(x)} - \frac{\Delta_x \mathcal{P}_0(x)}{\mathcal{P}^*(x)}\right] \tag{25}$$

$$= \mathbb{E}_{\mathcal{P}^*}\left[\frac{1}{(\mathcal{P}^*)^{(w)}(x)}\sum_{y'\in\mathcal{Y}} \Delta_x^w \frac{w(x,y')}{\mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}\mathcal{P}_0(x,y') - \frac{1}{(\mathcal{P}^*)^{(w)}(x)}\sum_{y'\in\mathcal{Y}} \Delta_x \frac{w(x,y^*)}{\mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]}\mathcal{P}_0(x,y')\right] \tag{26}$$

$$= \mathbb{E}_{\mathcal{P}^*}\left[\frac{1}{(\mathcal{P}^*)^{(w)}(x)}\left(\sum_{y'\in\mathcal{Y}} \Delta_x^w \frac{w(x,y')}{\mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}\mathcal{P}_0(x,y') - \Delta_x \frac{w(x,y^*)}{\mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]}\mathcal{P}_0(x,y')\right)\right] \tag{27}$$

$$= \mathbb{E}_{\mathcal{P}^*}\left[\frac{1}{(\mathcal{P}^*)^{(w)}(x)}\left(\sum_{y'\in\mathcal{Y}} \Delta_x^w \frac{w(x,y')}{\mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}\mathcal{P}_0(x,y') - \Delta_x^w \frac{\Delta_x^w w(x,y^*)}{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]}\mathcal{P}_0(x,y')\right)\right] \tag{28}$$

$$= \mathbb{E}_{\mathcal{P}^*}\left[\frac{1}{(\mathcal{P}^*)^{(w)}(x)}\left(\sum_{y'\in\mathcal{Y}} \Delta_x^w \mathcal{P}_0(x,y')\left(w(x,y') - \frac{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}{\Delta_x^w \mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]} w(x,y^*)\right)\right)\right] \tag{29}$$

$$\tag{30}$$

Where this term is negative if $w(x,y') - \frac{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}{\Delta_x^w \mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]} w(x,y^*) \leq 0$ which happens when $w(x,y') \leq \frac{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}{\Delta_x^w \mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]} w(x,y^*)$ □

.

Example *If we have $\frac{\Delta_x}{\Delta_x^w} \geq \lambda$ for all x then if for $w(x,y^*) = \lambda \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]$ we have that:*

$$\begin{aligned} w(x,y^*) - \frac{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}{\Delta_x^w \mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]} w(x,y^*) &= \lambda \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)] - \frac{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}{\Delta_x^w} \\ &\leq 0 \end{aligned}$$

For all other y', setting $w(x,y') \leq \frac{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}{\Delta_x^w}$ is sufficient for $w(x,y') - \frac{\Delta_x \mathbb{E}_{x,y\sim\mathcal{P}_0}[w(x,y)]}{\Delta_x^w \mathbb{E}_{x,y\sim\mathcal{P}^*}[w(x,y)]} w(x,y^*) \leq 0$. Finally we can see that $\alpha_{\text{eff}} = \lambda\alpha$ so setting $\lambda \geq 0$ we have a setting where $c \leq 0$ and $\alpha_{\text{eff}} \geq \alpha$.

B.2. Proofs for Results in Section 3

Proposition 3.1. *For a mixture distribution $\mathcal{P}_\alpha$, let $\mathcal{Q}_{\mathcal{P}_\alpha}$ be the set of distribution in a ball of radius δ around $\mathcal{P}_\alpha$ as defined in Equation (6). Then $\mathcal{Q}_{\mathcal{P}_\alpha}$ contains a distribution $\mathcal{P}_{\alpha_{\text{eff}}}$ with effective collective size*

$$\alpha_{\text{eff}} = \alpha + \frac{\delta}{\mathcal{D}_f\left(\mathcal{P}^* \| \mathcal{P}_\alpha\right)}. \tag{13}$$

Proof. Letting:

$$\mathcal{P} = \alpha\mathcal{P}^* + (1-\alpha)\,\mathcal{P}_0,$$

Then we want to consider for what λ does the mixture distribution $\lambda P^* + (1-\lambda)\mathcal{P}$ lies in $\mathcal{Q}_\delta$. Now we have that:

$$\begin{aligned} \mathcal{D}_f(\lambda\mathcal{P}^* + (1-\lambda)\mathcal{P} \,||\, \mathcal{P}) &\leq \lambda\mathcal{D}_f(\mathcal{P}^* \,||\, \mathcal{P}) + (1-\lambda)\mathcal{D}_f(\mathcal{P} \,||\, \mathcal{P}) \\ &= \lambda\mathcal{D}_f(\mathcal{P}^* \,||\, \mathcal{P}) \end{aligned}$$

Now, we have that:

$$\begin{aligned} \mathcal{D}_f(\mathcal{P}^* \,||\, \mathcal{P}) &= \mathcal{D}_f(\mathcal{P}^* \,||\, \alpha\mathcal{P}^* + (1-\alpha)\,\mathcal{P}_0) \\ &= \mathcal{D}_f(\mathcal{P}^* \,||\, \alpha\mathcal{P}^* + (1-\alpha)\,\mathcal{P}_0) \\ &\leq \alpha\mathcal{D}_f(\mathcal{P}^* \,||\, \mathcal{P}^*) + (1-\alpha)\mathcal{D}_f(\mathcal{P}^* \,||\, \mathcal{P}) \\ &= (1-\alpha)\mathcal{D}_f(\mathcal{P}^* \,||\, \mathcal{P}) \end{aligned}$$

Now plugging in the above we have that the following is sufficient to guarantee $\lambda P^* + (1-\lambda)\mathcal{P} \in \mathcal{Q}_\delta$:

$$\lambda \leq \frac{\delta}{(1-\alpha)\mathcal{D}_f(\mathcal{P}^* \,||\, \mathcal{P})}$$

Finally, note the collective proportion in $\lambda\mathcal{P}^* + (1-\lambda)\mathcal{P}$ is $\alpha + \lambda - \alpha\lambda$. Plugging in the largest λ and collecting terms gives the desired result. □

Proposition 3.2. *[Effective Collective Size of JTT (Liu et al., 2021)] For JTT trained on $\mathcal{P}_\alpha$, let λ be the up-weighting parameter, f be the classifier learned in the first phase and define*

$$\begin{aligned} P_E &\coloneqq \mathcal{P}_\alpha\left[f(X) \neq Y\right] \text{ and} \\ P_{E|C} &\coloneqq \mathcal{P}_\alpha\left[f(X) \neq Y \mid (X,Y) \text{ in the collective}\right]. \end{aligned} \tag{14}$$

Then, the effective collective size is given by

$$\alpha_{\textit{eff}} = \alpha\frac{\lambda P_{E|C} + \left(1 - P_{E|C}\right)}{\lambda P_E + (1 - P_E)}. \tag{15}$$

Proof. We have that the effective collective size is defined as:

$$\alpha_{\text{eff}} = \frac{\mathbb{E}[w(X)\mathbb{1}\{\text{X is in the collective}\}]}{\mathbb{E}[w(X)]}$$

In this case we have that $\mathbb{E}[w(X)] = \lambda P_E + (1 - P_E)$ and that:

$$\begin{aligned} \mathbb{E}[w(X)\mathbb{1}\{\text{X is in the collective}\}] &= P(\text{X is in the collective})\mathbb{E}[w(X) \mid \text{X is in the collective})] \\ &= \alpha\left(\lambda P_{E|C} + (1 - P_{E|C})\right) \end{aligned}$$

Inputting these expression into the effective collective size gives the result.

□

Validation Control For some validation distribution $\mathcal{P}_V = \beta\mathcal{P}^* + (1-\beta)\mathcal{P}_0$ and some minimal acceptable error ξ on the validation set, we first define an abstract version of CVaR-DRO in Algorithm 3. Then we restate and prove Proposition 3.3.

Algorithm 3 Ideal Continuous Reweighting

Input: Validation Distribution $\mathcal{P}_V = \beta\mathcal{P}^* + (1-\beta)\mathcal{P}_0$, Uncertainty ball $\mathcal{Q}_p$ centred at $\mathcal{P} = \alpha\mathcal{P}^* + (1-\alpha)\mathcal{P}_0$
$f_0(x) \leftarrow \arg\max_{y\in\mathcal{Y}} \mathcal{P}(Y = y \mid X = x)$
$t \leftarrow 1$
while $t \leq T$ **do**
  $\mathcal{P} \leftarrow \arg\max_{Q\in\mathcal{Q}_p} \mathbb{E}_Q\left[\ell\left(f_{t-1}(x), y\right)\right]$
  $f_t(x) \leftarrow \arg\max_{y\in\mathcal{Y}} \mathcal{P}(Y = y \mid X = x)$
  $t \leftarrow t + 1$
end while
$t_{\max} = \arg\max_{t\leq t} \mathbb{E}_{\mathcal{P}_V}\left[\ell\left(f_{t+1}(x), y\right)\right]$
Return: $f_{t_{\max}}$

Proposition 3.3. *Let f be the output of Algorithm 3, and $f_{\mathcal{P}_\alpha}$ be the Bayes optimal classifier on the mixture distribution $\mathcal{P}_\alpha$. Then we have that the success S_f and $S_{f_{\mathcal{P}_\alpha}}$ with f and $f_{\mathcal{P}_\alpha}$, respectively, relate as*

$$S_f - S_{f_{\mathcal{P}_\alpha}} \geq \frac{\mathcal{P}_V\left[f(X) = Y\right] - \mathcal{P}_V\left[f_{\mathcal{P}_\alpha}(X) = Y\right]}{\beta - \alpha}. \tag{16}$$

Proof. This follows as we have:

$$\Pr_{\mathcal{P}_V}(f(X)=Y)-\Pr_{\mathcal{P}_V}(f_{\mathcal{P}_\alpha}(X)=Y)=\frac{\beta-\alpha}{1-\alpha}\left(S_f-S_{f_{\mathcal{P}_\alpha}}\right)+\frac{1-\beta}{1-\alpha}\left(\Pr_{\mathcal{P}}(f(X)=Y)-\Pr_{\mathcal{P}}(f_{\mathcal{P}_\alpha}(X)=Y)\right)$$

But as $f_{\mathcal{P}_\alpha}$ is Bayes optimal on $\mathcal{P}$ we have that $\left(\Pr_{\mathcal{P}}(f(X)=Y)\leq\Pr_{\mathcal{P}}(f_{\mathcal{P}_\alpha}(X)=Y)\right)$ which implies that:

$$(\beta-\alpha)\left(S_f-S_h\right)\geq\Pr_{\mathcal{P}_V}(f(X)=Y)-\Pr_{\mathcal{P}_V}(f_{\mathcal{P}_\alpha}(X)=Y)$$

Re-arranging terms gives the result. □

B.3. Proofs for Results in Section 4

Theorem 2. *Consider a base distribution $\mathcal{P}_0$ and a learning algorithm $\mathcal{A}$ which outputs $h_{\mathcal{P}_\alpha}$ when learning on $\mathcal{P}_\alpha$. For any given distribution $\mathcal{Q}_0$, we denote the corresponding classifier TV distance as*

$$\omega_{\mathcal{Q}_0}=\mathrm{TV}\left(\mathcal{P}^*(X)\times h_{\mathcal{P}_\alpha}(X)\,|\mathcal{P}^*(X)\times f_{\mathcal{Q}_\alpha}(X)\right). \tag{19}$$

Then the collective success of algorithm $\mathcal{A}$ on $\mathcal{P}_\alpha$, is bounded below as

$$S(\alpha)\geq\sup_{\mathcal{Q}_0}\left\{1-\frac{\omega_{\mathcal{Q}_0}}{1-\omega_{\mathcal{Q}_0}}-\frac{1-\alpha}{\alpha}\mathcal{Q}_0(\mathcal{X}^*)\Delta_{\mathcal{Q}_0}\right\}, \tag{20}$$

where $\Delta_{\mathcal{Q}_0}=\max_{x\in\mathcal{X}^*}\max_{y\in\mathcal{Y}}\mathcal{Q}_0(y|x)-\mathcal{Q}_0(y^*|x)$.

Proof. Let Q be any distribution satisfying:

$$\mathrm{TV}(\mathcal{P}_\alpha(X)\times h_{\mathcal{P}_\alpha}(X),\mathcal{P}_\alpha(X)\times f_{Q_\alpha}(X))\leq\omega_{\mathcal{Q}_0}$$

Now following Hardt et al. (2023), we have that $f_{Q_\alpha}(x)=y$ if:

$$\alpha>(1-\alpha)\,\Delta_{x,Q}\frac{Q(x)}{P^*(x)}$$

Where $\Delta_{x,Q}=\max_{y\in\mathcal{Y}}Q(y\mid x)-Q(y^*\mid x)$. Therefore, following a similar argument to Hardt et al. (2023), we have that:

$$\begin{aligned}
S(\alpha)&=\Pr_{x\sim\mathcal{P}^*}\{f(x)=y^*\}\\
&\geq\Pr_{x\sim\mathcal{P}^*}\left\{\alpha>(1-\alpha)\,\Delta_{x,Q}\frac{Q(x)}{P^*(x)}\right\}\\
&=\mathbb{E}_{x\sim\mathcal{P}^*}\left[\mathbb{1}\left\{1-\frac{(1-\alpha)}{\alpha}\Delta_{x,Q}\frac{Q(x)}{P^*(x)}>0\right\}\right]\\
&\geq\mathbb{E}_{x\sim\mathcal{P}^*}\left[1-\frac{(1-\alpha)}{\alpha}\Delta_{x,Q}\frac{Q(x)}{P^*(x)}\right]\\
&=1-\frac{(1-\alpha)}{\alpha}\mathbb{E}_{x\sim\mathcal{P}^*}\left[\Delta_{x,Q}\frac{Q(x)}{P^*(x)}\right]\\
&\geq1-\frac{(1-\alpha)}{\alpha}\left(Q(\mathcal{X}^*)\Delta_Q\right)
\end{aligned}$$

Now the total variation constraint can be added to give that the success under $A(\mathcal{P}_\alpha)$ is

$$S(\alpha)\geq1-\frac{\omega}{1-\omega}-\frac{(1-\alpha)}{\alpha}\left(Q(\mathcal{X}^*)\Delta_Q\right).$$

□

Example An example of where this would hold be when $P_0(y^* \mid x) = 0$, $w(x, y^*) = a$ where a is constant, and $w(x, y') \leq \mathbb{E}_{x,y\sim\mathcal{P}^0}[w(x, y)]$ for all $x \in \mathcal{X}^*$. Intuitively, this corresponds to a scenario where the algorithm places lower weight the region on $\mathcal{X}^*$ than average.

B.4. Example for Theorem 2

Proposition B.4. *There exists a problem setting, defined by a data distribution $\mathcal{P}_0$, a biased learning algorithm $\mathcal{A}$, and collective signal g such that the success obtained with Theorem 2 is larger than that obtained by Theorem 1.*

We prove this below.

Base distribution Building on the intuition from the MNIST-CIFAR example, we assume a distribution $\mathcal{P}_0(x_1, x_2, y)$ over $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}$ where $\mathcal{X}_i$ is a set of size 10 and $\mathcal{Y} = \{y_+, y_-\}$. In the base distribution $\mathcal{P}_0$ we assume that both x_1, x_2 are perfectly correlated with y and so either is enough to determine the outcome. This can be seen as similar to MNIST-CIFAR where x_1 and x_2 correspond to the MNIST and CIFAR image respectively. We will also take the following:

1. Both labels have equal probability, so $\mathcal{P}_0(y) = \frac{1}{2}$.
2. Each set $\mathcal{X}_i$ can be partitioned as: $\mathcal{X}_i{=}\mathcal{X}_{i+} \cup \mathcal{X}_{i-}$ where $P(y{=}y_+ \mid x_i{\in}\mathcal{X}_{i+}){=}1$ and likewise for the other class. This is possible as each x_i is perfectly correlated with y. We also take that $|\mathcal{X}|_{i\star} = |\mathcal{X}|_{i\star}$ for $i \in \{1, 2\}$ and $\star \in \{+, y_-\}$.
3. The collective controls 10% of the data $\alpha{=}0.1$, its target label is $y^*{=}y_-$, and the signal is $g(x_1, x_2) = g(x_1, C)$ where $C \in \mathcal{X}_{2+}$ is constant.
4. The learning algorithm $\mathcal{A}$ works, similarly to JTT, in two stages:
   (a) In the first stage, the algorithm learns a Bayes optimal classifier $f_1 : \mathcal{X}_1 \to \mathcal{Y}$ that uses only x_1.
   (b) In the second stage, the algorithm then stores all the (x_1, x_2, y) triples that are misclassified by f_1 by saving the pair (x_1, x_2) in the error set E and their label y in a function $f_E : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{Y}$ that gets the pair (x_1, x_2) and outputs y.
   (c) The algorithm then outputs a classifier

$$f(x_1, x_2) = \begin{cases} f_E(x_1, x_2) & (x_1, x_2) \in E \\ f_1(x_1) & (x_1, x_2) \notin E \end{cases}$$

The second assumption defines the following 4 distributions.

1. $P_{1+}(x_1)$ is a uniform discrete probability over the set X_{1+} which contains 5 values that correspond to the label y_+, i.e. $P_{1+}(x_1 \in X_{1+}) = \frac{1}{5}, P_1(x_1 \notin X_{1+}) = 0$.
2. $P_{1-}(x_1)$ is a uniform discrete probability over the set X_{1-} that correspond to the label y_-.
3. $P_{2+}(x_2)$ is a uniform discrete probability over the set X_{2+} that correspond to the label y_+.
4. $P_{2-}(x_2)$ is a uniform discrete probability over the set X_{2-} that correspond to the label y_-.

Then we can get the full probabilities

$$\begin{aligned} \mathcal{P}_0(x_1, x_2 | y = y_+) &= P_{1+}(x_1) P_{2+}(x_2) \\ \mathcal{P}_0(x_1, x_2 | y = y_-) &= P_{1-}(x_1) P_{2-}(x_2) \\ \mathcal{P}_0(x_1, x_2) &= \frac{1}{2}\left(P_{1+}(x_1) P_{2+}(x_2) + P_{1-}(x_1) P_{2-}(x_2)\right). \end{aligned}$$

Let the collective signal be $g(x_1, x_2) = g(x_1, C)$ where C is a constant and the target label is $y^* = y_-$. Then the uniqueness of the signal is

$$\begin{aligned}
\xi = \mathcal{P}_0(x_1, C) &= \sum_{x_1 \in X_1} \left[P_{1+}(x_1) P_{2+}(C) P(y = y_+) + P_{1-}(x_1) P_{2-}(C) P(y = y_-) \right] \\
&= \sum_{x_1 \in X_1} \left[\frac{1}{2} \left(P_{1+}(x_1) P_{2+}(C) + P_{1-}(x_1) P_{2-}(C) \right) \right] \\
&= \frac{1}{2} P_{2+}(C) \left(\sum_{x_1 \in X_1} P_{1+}(x_1) \right) + \frac{1}{2} P_{2-}(C) \left(\sum_{x_1 \in X_1} P_{1-}(x_1) \right) \\
&= \frac{1}{2} \left(P_{2+}(C) + P_{2-}(C) \right) \\
&= \frac{1}{2} \left(\frac{1}{5} + 0 \right) \\
&= \frac{1}{10},
\end{aligned}$$

and the sub-optimality gap is

$$\begin{aligned}
\Delta &= \max_{x \in X^*} \max_y \left(\mathcal{P}_0(y|x) - \mathcal{P}_0(y^*|x) \right) \\
&\geq \mathcal{P}_0(y_+ | x_1 \in X_{1+}, x_2 = C) - \mathcal{P}_0(y_- | x_1 \in X_{1-}, x_2 = C) \\
&= 1.
\end{aligned}$$

Now, according to theorem 1, the lower bound of success when using Bayes optimal classifier with $\epsilon = 0$ is

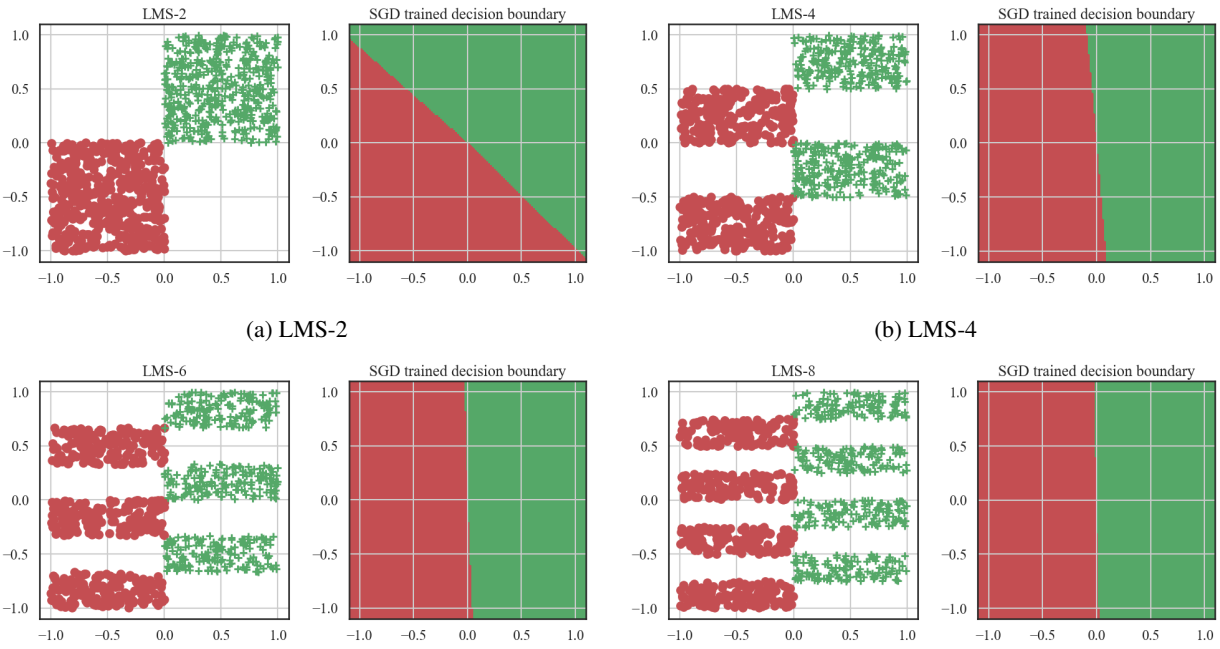
$$S_{\text{Bayes}}(\alpha) \geq 1 - \frac{1-\alpha}{\alpha} \Delta\xi - \frac{\epsilon}{1-\epsilon} = 1 - \frac{9}{10} - 0 = 0.1.$$

Algorithmic bias The observed mixture distribution $\mathcal{P}_\alpha$ now contains conflicting labels when $x_1 \in \mathcal{X}_{1+}$ and $x_2 = C$, as it can be sampled from either the base distribution $\mathcal{P}_0$ with $y{=}y_+$ or from the collective distribution $\mathcal{P}^*$ with $y{=}y_-$. As defined, the first-stage classifier f_1 of the learning algorithm $\mathcal{A}$ is Bayes optimal w.r.t x_1. The Bayes optimal f_1, with the collective being small, will predict the $f_1(x_1 \in \mathcal{X}_{1+}) = y_+$ label as it is more probable to come from $\mathcal{P}_0$. As a result, only the collective samples will be misclassified and will dominate the second stage classifier f_E such that $f_E(x_1, C) = y_-$. Finally the algorithm will return the classifier $f = \mathcal{A}(\mathcal{P}_\alpha)$ where

$$f(x_1, x_2) = \begin{cases} y^* = y_- & x_2 = C \\ \arg\max_y \mathcal{P}_0(y|x_1) & \text{else} \end{cases}.$$

Compare this with a Bayes optimal classifier f that uses both x_1 and x_2 equally when given a sample $(x_1 \in X_{1+}, x_2 = C)$:

$$\begin{aligned}
f(x_1 \in X_{1+}, x_2 = C) &= \arg\max_y \mathcal{P}_\alpha(y | x_1 \in X_{1+}, x_2 = C) \\
&= \arg\max_y \begin{cases} \mathcal{P}_\alpha(y = y_+ | x_1 \in X_{1+}, x_2 = C) & y = y_+ \\ \mathcal{P}_\alpha(y = y_- | x_1 \in X_{1+}, x_2 = C) & y = y_- \end{cases} \\
&= \arg\max_y \begin{cases} \alpha \mathcal{P}^*(y = y_+ | x_1 \in X_{1+}, x_2 = C) + (1-\alpha) \mathcal{P}_0(y = y_+ | x_1 \in X_{1+}, x_2 = C) \\ \alpha \mathcal{P}^*(y = y_- | x_1 \in X_{1+}, x_2 = C) + (1-\alpha) \mathcal{P}_0(y = y_- | x_1 \in X_{1+}, x_2 = C) \end{cases} \\
&= \arg\max_y \begin{cases} \alpha \cdot 0 + (1-\alpha) \cdot 1 & y = y_+ \\ \alpha \cdot 1 + (1-\alpha) \cdot 0 & y = y_- \end{cases} \\
&= \arg\max_y \begin{cases} 0.9 & y = y_+ \\ 0.1 & y = y_- \end{cases} = y_+
\end{aligned}$$



(a) LMS-2

(b) LMS-4

(c) LMS-6

(d) LMS-8

Figure 8. Trained classifier on LMS-k with no collective action. Left figure of each pair is a visualisation of the dataset, and the right figure of each pair is the decision boundary of the trained classifier.

Surrogate distribution and success bounds Let $\mathcal{Q}_0$ be the a distribution similar to $\mathcal{P}_0$, such that $\mathcal{Q}_0(y|x_1) = \mathcal{P}_0(y|x_1)$ but x_2 is discrete uniformly distributed over N values, meaning $\mathcal{Q}_0(\mathcal{X}^*) = \mathcal{Q}_0(x_2 = C) = \mathcal{Q}_0(x_2) = \frac{1}{N}$. Note than N can be as large as we want, virtually resulting in $\mathcal{Q}_0(\mathcal{X}^*) \approx 0$. In other words, large N makes the signal almost 0-unique.

Since x_2 in $\mathcal{Q}_0$ is i.i.d. and is not correlated with y, the Bayes optimal classifier $f_{\mathcal{Q}_\alpha}$ on the mixture distribution $\mathcal{Q}_\alpha$ will only will only x_1, or x_2 if it is equal to C:

$$f_{\mathcal{Q}_\alpha}(x_1, x_2) = \begin{cases} y^* & x_2 = C \\ \arg\max_y \mathcal{Q}_0(y|x_1) & \text{else} \end{cases}.$$

Then, from the definition of $\mathcal{Q}_0$ it stems that $f_{\mathcal{Q}_\alpha} = h_{P_\alpha}$. Plugging that in the definition for w (Equation (19)) we get $w_{\mathcal{Q}_0} = 0$. Now, the bound for success according to theorem 2:

$$S_{\text{bias}}(\alpha) \geq 1 - \frac{w}{1-w} - \frac{1-\alpha}{\alpha} Q_0(\mathcal{X}^*) = 1 - \frac{9}{N} \approx 1 > S_{\text{Bayes}}$$

Making the bound from theorem 2 tighter than the bound of theorem 1.

C. Experiments

Experimental details For the 2D datasets, we used an MLP with layers sizes of [64, 32, 16, 2] with ReLU activations. For all image datasets we used the ResNet50 model. In all experiments we used the PyTorch ADAM optimizer with the default parameters, a learning rate of 5×10^{-4} and a batch size of 128. Each experiment was run multiple times with different random seeds, and in all figures the lines represent the means over the seeds, and the region around the lines is the 95% confidence interval according to Student's t-distribution.

Example of simplicity bias in action To show the effect of simplicity bias, here we repeat training a classifier on the LMS-k dataset with no collective action for different ks. Figure 8a to 8d show how the decision boundary depends more on x_1 as k grows.