Improving Group Distributional Robustness by Learning to Rank

Anonymous Author(s)

ABSTRACT

We consider the challenging problem of training a classification model when the data are naturally structured into groups. Recent work has shown that standard training via empirical risk minimization (ERM) can produce models that achieve high accuracy on average but low accuracy on certain groups, particularly those groups that are underrepresented in the data. A predominant approach to tackle this group distributional robustness problem strives to minimize the worst group error (akin to a *minimax* strategy) on the training data hoping that it will generalize to the out-of-distribution (OOD) test data. However, this is often suboptimal, especially when the OOD test data contains previously unseen groups, leading to a significant drop in predictive performance. Inspired by ideas from the learning-to-rank literature, this paper first proposes to use Discounted Cumulative Gain (DCG) as a metric of model quality, which facilitates better hyperparameter tuning and model selection for this group robustness problem. Being a ranking-based metric, DCG weights multiple poorly-performing groups (instead of considering just the group with the worst performance). Our empirical results show that this smoothed *soft-minimax* approach leads to selecting models with better OOD performance on the test data. As a natural next step, we build on our results to propose a ranking-based training method called **Discounted Rank Upweighting (DRU)** which differentially up-weights a ranked list of poorly-performing groups in the training data to learn models that exhibit strong OOD performance on the test data. Results on several synthetic and real-world datasets highlight our group-ranking-based approach's superior generalization ability in selecting and learning models that are robust to group distributional shifts.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Learning to rank**; **Supervised learning by classification**; *Natural language processing*.

KEYWORDS

robust machine learning, distribution shifts, learning to rank

ACM Reference Format:

Anonymous Author(s). 2018. Improving Group Distributional Robustness by Learning to Rank. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Data are naturally split into groups in many machine learning contexts, e.g., a sentiment classification task with reviews from different users or an image-based user handwriting recognition system. In both these examples, a group corresponds to a user. In other contexts, such as online toxicity detection, the groups might be implicit, e.g., certain user demographics, and need annotation. More broadly, consider the scenario where we are given training examples stratified non-uniformly into groups. Our goal is to build a model for this scenario that generalizes to all groups by providing comparable classification accuracies—a key objective of deploying robust and fair machine learning practices [7, 8, 15].

Recent work on robust and equitable machine learning has shown that the traditional approach of minimizing the average training error, also known as empirical risk minimization (ERM), can be suboptimal for this grouped data setting. ERM produces models that achieve low test error on average but incur high errors on certain important groups in the data. Typically, the groups on which the ERM underperforms are the minority groups in the training data. In human-centric applications, such minority groups usually correspond to underrepresented or marginalized populations, which raises serious ethical and fairness concerns. One of the main reasons ERM conceals poor performance on minority groups behind a vastly superior average accuracy is its reliance on spurious relationships between labels and some features on majority groups to achieve high average accuracy [4, 6, 9, 11, 24]. Unfortunately, such correlations between labels and features are nonexistent or present with an opposite sign in the minority groups. This leads to ERM severely underperforming on minority groups (or new groups) while overfitting to the majority groups.

Prior research has tackled this intriguing grouped data classification problem by building models that have a low worst-group error on the training dataset. One such prominent model called Group Distributional Robust Optimization (Group DRO) seeks to minimize the worst group's training loss [22]. At every update step during the training procedure, Group DRO focuses only on the group with the highest regularized loss and strives to minimize it. While Group DRO has shown promising performance compared to ERM on some benchmark datasets, it is known to perform poorly when the different groups contain varying amounts of predictive signal [16]. Essentially, Group DRO (and related approaches) are myopic since they focus only on one group and ignore the rest.

This minimax-style approach (minimizing the worst-group performance) to dealing with grouped data simplistically assumes that the worst-performing group on the training dataset is distributionally similar to the worst-group in the test data. This assumption is especially problematic in *domain-generalization* scenarios where the test data contains previously unseen out-of-distribution (OOD) groups that do not overlap with the training or validation data.

Another problem with the minimax approach arises while tuning hyperparameters and performing model selection. Selecting hyperparameters based only on worst-group performance inevitably leads

to many ties as several choices of hyperparameters may lead to the same validation (worst-group) accuracy. Even if one can break ties using some heuristics, the model selected using worst-group performance on the validation dataset fails to provide consistently good worst-group performance on the test dataset. It turns out that the minimax approach ends up overfitting to the OOD validation dataset and does not account for the distributional similarity between OOD validation and test data since it only optimizes the performance of the worst-group during model training. This is concerning since it defeats the purpose of the minimax approach to learning models that are robust to OOD data in the first place.

In this paper, we draw on ideas from the learning-to-rank literature to provide a more effective solution to the group distributional robustness problem. We make two contributions in this paper that can be summarized as: **We develop methods that use a weighted ranking of groups based on their classification accuracies to 1) choose hyperparameters and perform model selection, and 2) train the model.**[This sentence may need to be polished? –Bohan Zhang]

At a high level, we propose to use a *soft-minimax* approach, which *smooths* the predictive signal from multiple poorly-performing groups by weighting them based on their accuracy-based ranked order. Specifically, we use **Discounted Cumulative Gain (DCG)** [13] metric from the learning-to-rank literature to rank and then weight several *poorly* performing groups to inform model selection. DCG allows us to consider the validation performance across more than one group while choosing hyperparameters, thus lowering the potential risk of overfitting. As we will see later in the results section, using DCG for model selection leads to superior OOD test set performance which hints at lower overfitting. Further, the DCG metric is also less prone to having ties in hyperparameter choices, leading to statistically identified models.

Our use of DCG for model selection is driven by the analogy between the learning-to-rank problem [5] and the task of building group distributionally robust machine learning models. In learning-to-rank, the objective is to maximize the scores earned by top items returned by an information retrieval system, e.g., a search engine, with the **higher**-ranked items having higher importance. Similarly, in group distributional robustness, we want to prioritize the groups with **lower** accuracy and give them higher importance in both model selection and model training.

Next, we turn to the task of developing a novel training method for group distributional robustness. Borrowing intuition from our use of DCG for model selection, we propose a new robust training method **Discounted Rank Upweighting (DRU)**. DRU iteratively upweights groups during each epoch of the training based on that group's classification accuracy ranking. We develop two variants of the DRU model, one which upweights all examples from a group and a second one which only upweights the misclassified examples. As we show later, DRU significantly outperforms multiple state-of-the-art methods for group distributional robustness on several synthetic and real-world benchmark datasets.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes the preliminaries including problem setup and baseline methods. In Section 4, we describe the methods for group distributional robustness as learning to rank compromising DCG for model selection and the novel model training method, the DRU. Then, in section 5 and 6 we present the experiment results for DCG model selection and DRU for model training respectively. We conclude in Section 7.

2 RELATED WORK

This paper focuses on group distributional robustness, i.e., training models whose performance does not deteriorate across pre-defined groups. There are other notions of robustness in machine learning, e.g., adversarial robustness or the study of long-tailed distributions, but that is beyond the scope of this paper.

2.1 Group Distributional Shifts

There are a couple of ways to split data into groups based on prior work. First, groups can occur in data organically based on the data collection procedure. For example, all reviews by a given user can be assembled into a group, or all the images taken from a particular camera can constitute a group. All the items in one group show the same characteristics and are assumed to follow the same data-generating process. These organic groups can be further divided into sub-populations based on meta-information about each group. For instance, the user groups can be divided into sub-groups based on the demographic constitution of those groups. Similarly, images taken from the same camera can be divided into sub-groups based on the photographer's identity. When the groups are pre-defined based on a human-related artifact, then social fairness also becomes salient due to a desire for parity in performance across the different demographics, e.g., CivilComments-WILDS [16]. The data can also be split into groups based on the interaction between the output label and a spurious feature, e.g., Waterbirds [25], CelebA[18],and MultiNLI [26] datasets.

Given the importance and prevalence of the grouped data setting, several algorithms have been developed for removing disparity in performance across the different groups. Some popular algorithms include Group Distributionally Robust Optimization (Group DRO), which directly minimizes the worst group's regularized error during model training [12, 22]. Invariant Risk Minimization (IRM) penalizes the distributions of learned representations with different optimal linear classifiers [1]. Both Group DRO and IRM require group annotation at training time. Recently, an approach called Just Train Twice (JTT) has been proposed that does not require group information at training time. JTT instead just upweights misclassified examples and retrains the model. It has been demonstrated to provide superior performance to Group DRO or IRM [17].

2.2 Minimax Framework

It is a popular decision rule that is widely used in many fields, including artificial intelligence, decision theory, and game theory [10]. It suggests that the optimal decision should have the best worst-case performance. In the group robustness literature, the minimax framework guides the decision to minimize the error of the worst-performing group, e.g., as done by Group DRO (See equation 1).

$$\min_{\theta \in \Theta} \max_{g \in G} \; Loss(x, y; \theta | g) \tag{1}$$

where $\theta \in \Theta$ is the set of model parameters, and $g \in G$ is the group. We extend the minimax framework into a *soft-minimax* setting

which smooths the worst-group performance by the weighted loss of multiple poorly-performing groups.

2.3 Learning to Rank

It is a subfield of the information retrieval literature which aims to build systems that can accurately retrieve top k documents from a document database. Essentially, it involves ranking the documents in a database based on their content. The common evaluation measures used in this literature include Mean Average Precision (MAP), Discounted Cumulative Gain (DCG), and (Normalized) Discounted Cumulative Gain ((N)DCG), and (N)DCG at k [13].

DCG at k simply adds up the scores earned at each position with inverse logarithm weights up to the k^{th} document, i.e.,

$$DCG@k = \sum_{i=1}^{k} \frac{Score_{(i)}}{\log_2(i+1)} \tag{2}$$

NDCG normalizes DCG by dividing the DCG for ideal order $iDCG@k$, i.e.,**[We can drop NDCG entirely since we don't use this concept at all. –Bohan Zhang]**

$$NDCG@k = \frac{DCG@k}{iDCG@k} \tag{3}$$

While our approach is inspired by learning-to-rank, the major difference is that in information retrieval literature, higher weights are assigned to the **higher**-ranked items (e.g., most relevant documents), while in our setting, higher importance weights are given to **lower**-ranked groups (i.e., worst performing groups). To the best of our knowledge, this is the first work that applies learning-to-rank to facilitating a *soft-minimax* strategy of training machine learning models with group distributional robustness.

3 PRELIMINARIES

3.1 Problem Setup

We consider the standard supervised learning setup of classifying an input $x \in \mathcal{X}$ as a label $y \in \mathcal{Y}$. We assume that the training data comprises of m_{train} groups from a set $\mathcal{G}$ where each group $g \in \mathcal{G}$ consists of n_g data points from a probability distribution $P_g(\mathcal{X}, \mathcal{Y})$. In addition to the feature x_j and label y_j, each training example j is also annotated with the subpopulation/group $g_j \in \mathcal{G}$ that it belongs to. To summarize, the training dataset contains n_{train} samples with group annotations in the format $\{(x_1, y_1, g_1), \ldots, (x_{n_{train}}, y_{n_{train}}, g_{n_{train}})\}$. Our goal is to learn a model $f_\theta : \mathcal{X} \times \mathcal{G} \rightarrow \mathcal{Y}$ parameterized by $\theta \in \Theta$. The group loss for group g is the average loss over all examples in group g, and we denote it as $l_g(\theta) = \mathbb{E}_{(x,y)\sim P_g(\mathcal{X},\mathcal{Y})}\, \mathcal{L}(x, y; f_\theta)$, for a loss function $\mathcal{L}$ and a machine learning model f_θ.

There are two common settings in the group distributional robustness literature, *domain generalization* [3] and *subpopulation shift* [14]. In the domain generalization setting, the OOD test data contains unseen groups. In contrast, in the subpopulation shift setting, the OOD test set only contains new proportions of groups or in-group distribution shifts but no previously unseen groups. This paper focuses on the more challenging domain generalization setting, which assumes no group overlap between OOD test and training/validation sets.

The performance evaluation metric for a robust model under group distribution shift is the OOD test set accuracy. More concretely, it is preferable to have a model with high worst-group accuracy on the OOD test data, but that does not sacrifice the average accuracy significantly.

3.2 Baseline Methods

We compare our approach against several competitive baselines as described below. All the methods (including our approach) use the same base learner f_θ—a finetuned DistilBERT model [23]. We describe the hyperparameter choices and other technical details of the various methods later in the paper.

(1) **Empirical Risk Minimization (ERM):** This is the standard training method which trains models to minimize the average training loss. The method doesn't take any group information into consideration while training the model.
(2) **Group Distributionally Robust Optimization (Group DRO):** Group DRO uses distributionally robust optimization to explicitly minimize the loss on the worst-case domain (or group) during training. Group DRO builds on the minimax approach developed in [20].
(3) **Just Train Twice (JTT):** As described earlier, JTT requires no group annotations and has been shown to improve worst group performance on several challenging benchmark applications over state-of-the-art Group DRO and IRM approaches. JTT involves a two-stage training approach which first trains a standard ERM model for several epochs and then trains a second model that upweights the training examples that the first model has misclassified [17].

4 METHODS: GROUP DISTRIBUTIONAL ROBUSTNESS AS LEARNING TO RANK

As just described, the goal of group distributional robustness is to learn models with superior worst-group accuracy on the OOD test dataset without sacrificing average accuracy. To achieve this goal, a common surrogate optimization objective function that past literature has employed is to learn models to maximize worst-group accuracy on the OOD validation dataset (See Equation 4). As discussed earlier, this minimax strategy of considering only the worst-group accuracy on the validation dataset is limiting and results in reduced generalizability.

$$\min_{\theta\in\Theta} \max_{g\in G_{val}} l_g(\theta) \tag{4}$$

Next, we discuss our proposed approach to tackle the limitations of the extant methods by using learning to rank inspired objective functions for group distributionally robust model selection and model training.

4.1 Using Learning-To-Rank for Model Selection

The minimax approach to robust model selection is suboptimal since it ignores the predictive signal from other groups. It also simplistically assumes that the worst-performing group on the validation dataset is distributionally similar to the worst group on the OOD test set (our evaluation metric). So, instead of this *hard*

minimax approach, we instead propose a *soft-minimax* approach that weights the errors from several poorly performing groups on the validation dataset to inform the hyperparameter choices for model selection. Intuitively, our approach can be seen as performing smoothing by borrowing statistical strength from several groups instead of just the worst group. We leverage the learning to rank literature to help us operationalize this soft group-weighting. Learning to rank literature contains several ranking-based metrics that provide discounted importance to various items, e.g., the Discounted Cumulative Gain (DCG) metric.

4.1.1 **Discounted Cumulative Gain (DCG) for Model selection:** First, use any base learner model, e.g., ERM to get the classification losses incurred by the different groups on the validation set. Next, sort the all the m_{val} groups according to their loss $g(1), g(2), \ldots, g(m_{val})$ from the largest (worst) to the smallest (best group). Then the composite DCG metric with a cutoff k becomes,

$$DCG@k(\theta) = \sum_{i=1}^{k} \frac{l_{g(i)}(\theta)}{\log_2(i+1)} \tag{5}$$

Equation 5 considers the k groups with the highest OOD validation errors and provides them increasing weights (higher weight for worse performing group).

Since the *inverse logarithm* function flattens fast as the number of groups increase, one can use DCG at the quantile-level as opposed to group-level. The quantile-level DCG takes in a list of quantiles $\boldsymbol{q} = [q_1, \ldots, q_k]$ that corresponds to groups $g^{(q_1)}, \ldots, g^{(q_k)}$ at these quantiles, for example $\boldsymbol{q} = [0, 1, \ldots, k]$ are the groups at quantile 0 (worst-group), quantile 1, up to quantile k. This leads to a slightly modified expression for DCG as shown below:

$$DCG_{\boldsymbol{q}}@k(\theta) = \sum_{i=1}^{k} \frac{l_{g^{(q_i)}}(\theta)}{\log_2(i+1)}. \tag{6}$$

An even more general form of DCG takes custom values $w_1, \ldots, w_{m_{val}}$ for group weights in the OOD validation set as:

$$DCG^{\boldsymbol{w}}(\theta) = \sum_{i=1}^{m_{val}} w_i \, l_{g(i)}(\theta) \tag{7}$$

It is easy to see that both the DCG at group-level with a cutoff k and the DCG at quantile-level with a quantile list $\boldsymbol{q}$ are special cases of this general form. Moreover when $w_1 = 1, w_2 = \ldots = w_{m_{val}} = 0$, Equation 7 reduces to the minimax approach of only weighting the worst-group performance and when $w_1 = \ldots = w_{m_{val}} = \frac{1}{m_{val}}$, it reduces to the average group accuracy.

4.1.2 Evaluation of the Model Selection metric. We just proposed a soft minimax-based model selection strategy, which generalizes the hard minimax used previously in the literature. Recall that the ultimate goal of effective model selection is to choose a model with superior performance on the OOD test set. So, how do we evaluate the effectiveness of our proposed metric over alternative model selection metrics?

An excellent way to think about this is how much concordance or agreement exists between the models selected by a given metric on the validation and the OOD test sets. A superior model selection metric should yield similar rankings of candidate models on either validation or test datasets. Thus, the best model on the validation set will also be the best model on the test data leading to effective model selection. For instance, consider our soft minimax metric; let's assume it ranks three candidate models as S2> S1> S3 based on validation set accuracy. Then, if the test set accuracies[1] of these three models are also S2> S1> S3, we consider the metric a good model selection strategy, and we can pick model S2 from this class of models. We use this intuition to guide our evaluation strategy for the model selection metric.

Let $r_{val}(M)$ denote the ranked accuracy list of models based on metric M, e.g., hard minimax, soft minimax, or average, on the validation set. Next, let r_{test}(worst-group-accuracy) represent a similar model accuracy list but based on worst group performance on the test set. Then, the metric M's *concordance* $C(M)$ can be defined as the similarity between $r_{val}(M)$ and r_{test}(worst-group-accuracy). Hence, a superior evaluation metric should have a high degree of similarity between the two rankings.

$$C(M) = \text{similarity}(r_{val}(M),\ r_{test}(\textit{worst-group-accuracy})) \tag{8}$$

The *similarity* in above Equation 8 can be operationalized by a function such as euclidean distance or cosine similarity.

4.2 Learning-To-Rank inspired novel method for Model Training

We just saw the use of DCG to *select* the best model from candidate models efficiently. Next, we propose a new method for *training* a model. Inspired by the recent success of the Just Train Twice (JTT) method [17] for group distributional robustness, our method also performs iterative upweighting of training examples. However, unlike JTT, it leverages group annotations at training time. Our novel method called **Discounted Rank Upweighting (DRU)**, which is inspired by learning to rank, performs an iterative upweighting on poorly performing groups. The key idea is to upweight training samples from the groups with the highest training errors but assign them differential importance commensurate with their ranking.

At each epoch t (excluding the first one) during the training process, a sample x with label y in the group $g \in \mathcal{G}_{train}$ can be upweighted by

$$w_g^t = \begin{cases} \frac{\log_2(C+2)}{\log_2(r_g^{t-1}+2)} & r_g^{t-1} <= C \\ 1 & r_g^{t-1} > C \end{cases} \tag{9}$$

where r_g^{t-1} is either the ranking index or the ranking quantile of the group g in the training set (ascending order of training accuracy) evaluated from the previous (t-1) epoch. C is a hyperparameter that controls the cutoff for upweighting (akin to k in $DCG@k$). If the group ranking is greater than the cutoff, then the weight is one, that is, no upweighting. Otherwise, if the group has lower accuracy, then, it will be weighted by the discounted log function shown above. Note that the constant '2' in this function is used to have discounted factors for training groups that are consistent with those in §4.1. If the upweighting is applied to all samples of each group regardless of their classification accuracy in the previous epoch, then the training objective of each epoch for a model with

[1] Recall that on test data we only care about worst group accuracy.

parameters θ is

$$J^t_{DRU}(\theta) = \sum_{g \in \mathcal{G}_{train}} \sum_{(x,y) \in g} w^t_g * \mathcal{L}(x, y; f_\theta) \tag{10}$$

for $t \neq 0$. As one can infer, the first epoch is always the standard ERM training.

The upweighting scheme shown in Equation 10 upweights all the samples from a given group. One can also choose to upweight only the misclassified samples from the previous epoch. Assuming the misclassified samples to constitute an error set E, the modified training objective function becomes:

$$J^t_{DRU}(\theta, E) = \sum_{g \in \mathcal{G}_{train}} \left[\sum_{(x,y) \in g \cap E} w^t_g * \mathcal{L}(x, y; f_\theta) + \sum_{(x,y) \in g \setminus E} \mathcal{L}(x, y; f_\theta) \right] \tag{11}$$

We compare both the objective functions in our empirical results. In the context of group robustness, researchers often try to model the underlying spurious features. However, unlike datasets with few groups and clear identification of spurious features by construction, e.g., WaterBirds [25], spurious features can be hard to locate. For instance, in tasks such as sentiment classification of user reviews, potential spurious features such as the writing style can be present in all the groups to varying degrees. We conjecture that differentially upweighting the various groups will help mitigate the impact of spurious features and help us identify robust predictive patterns in the data.

5 EXPERIMENTS: MODEL SELECTION

In this section, we evaluate the effectiveness of our soft minimax metric in selecting generalizable models. We first describe the real-world datasets that we used; then, we describe the details of our experimental setup. Finally, we present the model selection results.

5.1 Datasets Used

We use three real-world review sentiment classification datasets: AMAZON-WILDS [16], IMDB Movie Review Dataset [21], and a variation of Yelp Open Dataset[2]. The prediction task for all three datasets is to classify the review text to its corresponding 1-to-5 star rating. Each review is associated with a group, which corresponds to all reviews written by the same *user*. Each dataset consists of an in-distribution (ID) training set and out-of-distribution (OOD) validation and test sets. The OOD validation and test sets comprise reviews from disjoint sets of users (groups). The users in the training dataset are randomly split 50/50 to be in the ID validation and ID test datasets. Table 1 provides the summary statistics of each dataset. We use base and uncased DistilBERT [23] models finetuned on these three datasets as the base learner in all our experiments.

- **AMAZON-WILDS:** Collected as part of the WILDS dataset suite [16], the Amazon-WILDS dataset involves predicting star ratings from users' reviews of Amazon products. The training set has 245, 502 reviews from 1252 users (at least 75 reviews per user). The ID validation set consists of 46,950 reviews from 626 of the 1252 users in the training set. The ID test set is the same size as the ID validation set and contains reviews from the remaining 626 users from the training dataset. Finally, the OOD validation and the OOD test sets each have 100, 050 reviews (75 reviews per user) from 1, 334 new users.
- **IMDB Movie Reviews:** We downloaded the IMDB dataset from [21] and modified it to exhibit considerable OOD performance drops on the validation and test sets. To construct our dataset, we aggregate the data at the user level and split it into training, validation, and test sets using K-means clustering to ensure a significant distributional shift from ID to OOD sets. Specifically, we calculate the average of pre-trained DistilBERT embeddings of each user's reviews and then cluster their embeddings (k=2). One cluster is randomly selected as the ID set, and the other is the OOD set. Next, users in the OOD set are randomly split into OOD validation and OOD test sets, and the users in the ID set with at least 50 reviews are randomly divided into ID validation and ID test sets. The final training set has 41, 146 reviews from 666 users. The ID validation set consists of 20,070 reviews from 333 of the 666 users in the training set. Similarly, the ID test set contains 21,083 reviews from the other half of users in the training dataset. The OOD validation and test sets include 42, 703 and 43, 451 reviews from 561 and 560 unseen users, respectively, with each user containing at least 25 reviews.
- **Yelp Business Reviews:** WILDS dataset suite [16] contains a modified version of the Yelp Open Dataset; however, there's no accuracy drop from their ID set to OOD set. Thus, we modify it by clustering at the user level in a similar fashion as we did for the IMDB dataset. We set k=6 and select the two farthest clusters as the OOD and ID sets to have a significant out-of-distribution performance drop. The training set comprises 64, 931 reviews from 500 users. The ID validation and test sets consist of 20, 070 and 21, 083 reviews from 333 out of the 666 users in the training set, respectively. Finally, the OOD validation and test sets include 52, 200 and 52, 300 reviews from 522 and 523 unseen users, respectively.

[2]https://www.yelp.com/dataset/

Table 1: Dataset details. *Note:* **Number of groups is provided in the format (training, OOD validation, OOD test).**

Dataset	# Groups	Group size
AMAZON-WILDS	(1252,1334,1334)	≥ 75
IMDB Movie Review	(666, 561, 560)	≥ 25
Yelp Review	(500, 523, 522)	≥ 100

Table 2: Dataset Details - OOD accuracy drops. The performances are given in the format of (average, 10-th percentile, worst group). All three performance metrics are lower on OOD val/test sets than on their ID counterparts. *Note:* **10-th percentile group is one that has a lower accuracy than 90% of all groups.**

.

Dataset	ID val	OOD val	ID test	OOD test
AMAZON	(75.7, 58.7, 24.0)	(72.3, 54.7, 6.3)	(74.7, 57.3, 24.0)	(71.9, 53.3, 12.0)
IMDB	(64.7, 46.7, 26.7)	(62.6, 43.1, 15.6)	(65.4, 48.0, 20.0)	(63.2, 42.9, 15.0)
Yelp	(65.2, 54.9, 41.0)	(64.5, 54.0, 34.0)	(64.0, 55.9, 26.9)	(63.0, 52.0, 18.0)

Intuitively, the performance of a ERM model should significantly downgrade on OOD validation and test sets than on ID validation and test sets. Table 2 confirms the significant accuracy drops from ID to OOD on all three datasets.

5.2 Experiment Setup

We consider the following metrics that one can use to select models/hyperparameters from a validation set:

- **worst-group:** accuracy of the worst group;
- **average:** average across all groups;
- **10th percentile:** the accuracy of the group at the 10th percentile (lower accuracy than 90% groups);
- **gDCG@10:** DCG at group level for the 10 percent worst-performing groups;
- **qDCG@10:** DCG at quantile level for percentiles [0,1,...,10];
- **qDCG@50:** DCG at quantile level for percentiles [0,1,...,50];
- **qDCG@100:** DCG at quantile level for percentiles [0,1,...,100];
- **gDCG@100:** DCG at group level for all groups.

To compare the effectiveness of these metrics for model selection, we trained DistilBERT base-learner models using the Just Train Twice (JTT) algorithm. Then, we varied the two most important hyperparameters (number of epochs and the upweighting weights) of JTT to generate 16 candidate models. In particular we considered $T \in \{1, 2, 3, 5\}$ (number of epochs) and $\lambda \in \{2, 3, 5, 10\}$ (upweighting weights). Next, we rank these 16 models using the various metrics on their OOD validation set accuracy and then rank all the 16 models on worst group accuracy on the test OOD dataset. This process provides us with a ranked list of 16 models for each model selection metric on the validation set and another list ranking all the 16 models on their worst group accuracy on the OOD test set. Finally, we can assess the model selection performance of all the metrics by computing the similarity between their rankings of models on validation set with the "ground-truth" ranking of models on the test set (cf. Equation 8). In particular, we calculate the similarity of two ranked lists using euclidean distance, cosine similarity, and the NDCG of the model ranking on validation using the ranking on test as gold standard scores (see Equation 2 and Equation 3). The metric(s) which ranks the models on validation into the most similar positions with the true performance of the models on test set is(are) the most effective for selection.

5.3 Experiment Results

Table 3 shows the model selection results for Amazon, IMDB, and Yelp datasets. We report the similarity between the ranked list of the 16 candidate models on OOD validation set corresponding to each model selection metric and the ranked list of the models by their true accuracy on the worst group in OOD test. It turns out that when using "worst-group" or "10th percentile" to rank models on validation, there are many ties, which makes it challenging to identify the best model.

As can be seen from the results, the "worst-group," that is, the *hard minimax* approach performs the poorest compared to every other metric. This is surprising as it has identical semantics to the ground-truth metric we used on the test data, which confirms that the *hard minimax* metric has poor generalizability when distribution shift is present. The quantile-level DCG metrics (qDCGs) perform the best on all three datasets. Specifically, qDCG@10 performs the best on Amazon and IMDB and qDCG@50 performs the best on Yelp dataset. The "10th percentile" metric works better than worst-group or average accuracy, although not as good as the DCG-based metrics. This is understandable as the "10th percentile" worst-group is a special case of rank-based metric and it also smooths the *hard minimax* to a certain extent.

Table 3: Concordance between the ranked lists of the models on OOD validation by different metrics and the ranked list by worst-group accuracies on OOD test. *Note:* **ED = Euclidean Distance (lower is better), CS = Cosine Similarity (higher is better), NDCG = Normalized Discounted Cumulative Gain using test-worst-group ranking list as the gold standard (higher is better).**

	Amazon			IMDB			Yelp		
Metric	**ED**	**CS**	**NDCG**	**ED**	**CS**	**NDCG**	**ED**	**CS**	**NDCG**
worst-group	27.037	0.715	0.765	14.142	0.933	0.937	15.297	0.919	0.906
average	27.037	0.743	0.775	10.677	0.962	0.964	12.923	0.943	0.971
10th percentile	23.452	0.797	**0.884**	8.660	0.975	0.963	13.601	0.935	0.963
gDCG@10	21.424	0.839	0.861	8.485	0.976	0.985	14.036	0.939	0.943
qDCG@10	**20.421**	**0.854**	0.855	**8.124**	**0.978**	**0.986**	13.229	0.941	0.938
qDCG@50	23.979	0.798	0.821	8.832	0.974	0.975	**12.207**	**0.950**	0.974
qDCG@100	25.865	0.765	0.798	10	0.967	0.967	12.610	0.946	**0.976**
gDCG@100	25.593	0.770	0.803	10.677	0.962	0.964	12.610	0.946	**0.976**

A hidden strength of the DCG-based metric that is not visible in the result tables is its ability to break ties between candidate models. In our experiments, 43.75% and 37.5% of models have identical 10th percentile and worst-group accuracies on all three datasets, respectively. The lack of model identification power makes it hard to figure out which model is the best, and one has to resort to suboptimal heuristics such as random tiebreaks to choose the best model. Ties are rare in the case of DCG-based metrics since they evaluate models using discounted (logarithm weighted) ranks of several poorly performing groups instead of just relying on a single accuracy number as done by the "worst-group" or "10th percentile" metrics. Thus, the smoothing produced by our soft minimax metrics leads us to select better models. However, our results are still inconclusive regarding how much smoothing is optimal (with regard to the cutoff threshold of DCG) as it appears dataset-specific based on our experiments.

6 EXPERIMENTS: DRU-BASED MODEL TRAINING

Similar to the DCG-based model selection metrics, the DRU-based upweighting can also be performed by ranking the quantiles of the groups, i.e., **qDRU**, or simply ranking the indices of the groups, i.e., **gDRU**. **qDRU** is preferable when the number of groups is large since the logarithm function flattens out quickly in such a case. These upweighting methods also have **one** hyperparameter C, which controls the cutoff or the amount of smoothing.

Since we want to contrast with the hard minimax approach throughout this paper, we train models by only upweighting samples of the worst-performing group from the previous training epoch by a constant weight λ. It is easy to see that this hard minimax approach is a special case of our soft minimax approach in

which the cutoff for DRU is the rank of the worst group (0). We denote this boundary case as **Worst** in our results.

In their basic form, our upweighting methods **qDRU**, **gDRU** upweight all the samples from certain groups. A related upweighting strategy can be to further zoom in to each group and only upweight the misclassified examples from that group. We experiment with this seemingly more precise weighting strategy and denote it using the suffix "+M" in our results. For clarity, the default strategy of upweighting all examples from a group is suffixed "+G". This leads to four different variants of our DRU models, **qDRU + M**, **qDRU + G**, **gDRU + M**, and **gDRU + G**.

We further compare against another variant of the upweighting strategy that upweights only the misclassified examples from the previous training epoch by a constant factor λ. We call this approach **Const** in our results. This variant will help us tease apart the impact of the ranking-based logarithmic weighting since it is plausible that the improved accuracy might not be sensitive to the differential upweighting. Note that the **Const** method can not be applied to all the samples from each group (**+G**) since upweighting all the examples by the same amount makes the weights useless.

In addition to these methods, we compare against baseline methods **ERM**, **Group DRO**, and **JTT** described in Section §3.2. All methods except ERM have one hyperparameter (C for DRU-based methods, λ for JTT, Const, and Worst, and stepsize for Group GRO). We also considered **IRM** [1] as a potential baseline but could not obtain comparable performance to other baselines (ERM, Group DRO, JTT). We hypothesize that IRM does not fit to our scenario where there are a large number of groups. We therefore do not include IRM in the following experiments.

6.1 Synthetic Data Experiment

Before diving into the empirical performance of our approach on real-world datasets, we first showcase its improved performance in a controlled environment where we generate the data using a fixed data generating process. Past work has also used synthetic data to validate new methods for distribution shift [1, 2].

6.1.1 Synthetic Data Generation. The procedure for synthetic data generation is summarized in Algorithm 1. Our data-generating process assumes that each observation is generated by combining two predictive signals. The first signal is a "shared signal" present across all the groups and easily captured by any model and the second signal is an "idiosyncratic signal" that varies significantly across groups. As part of our controlled simulation setup, we vary the percentage of groups U containing idiosyncratic signals in a dataset. Specifically, we model the shared signal M by a Gaussian distribution $\mathcal{N}(\mu_s, \sigma_s)$ and the idiosyncratic signal is operationalized by a set of W Gaussian distributions whose each element represents a unique idiosyncratic signal. For each sample i of a given group g, we sample a shared signal $shared_g^i$ from the distribution M. Next, if the group g contains idiosyncratic signal (depends on U), one idiosyncratic signal distribution w_g is sampled from W based on a prior distribution p. Then for each sample i of the group, an idiosyncratic signal $idiosyncratic_g^i$ is sampled from w_g. Then, another simulation parameter $a \in [0, 1]$ that controls the strength of the idiosyncratic signal is sampled from a truncated Gaussian distribution. Note that for a group without idiosyncratic signal, each sample of the group will only have the shared signal. Finally, we get the feature representation for a given sample i of the group g as $x_g^i = shared_g^i + b * a * idiosyncratic_g^i$ where b is a boolean value indicating whether the group g contains the idiosyncratic signal (determined by U). The label of the sample is obtained as $y = \mathbb{1}_{\mathbb{R}_+}(f(x_g^i) + \sigma)$, where $\mathbb{1}_{\mathbb{R}_+}$ is the indicator function, f is function that transform features into labels (e.g., sin function) and σ is random Gaussian noise.

Algorithm 1 Synthetic Data Generation

Input: Q: number of reviews per group; U: % of groups with idiosyncratic predictive signal, W: a set of predefined Gaussian idiosyncratic predictive signals; p: probability of each idiosyncratic signal; $M \sim N(\mu_s, \sigma_s^2)$, the shared predictive signal.

```
S ← {}
for g ∈ G do
    a_g ← TruncatedN(0.75, 0.25, 0, 1);
    w_g ~ N(μ_g, σ_g^2) ← random.choices(W, p);
    has_idiosyncratic ← U(0, 1) <= U
    for i = 1,2,...,Q do
        shared_g^i ← N(μ_s, σ_s^2)
        x_g^i ← shared_g^i
        if has_idiosyncratic then
            idiosyncratic_g^i ← N(μ_g, σ_g^2)
            x_g^i ← x_g^i + a_g * idiosyncratic_g^i
        end if
        y_g^i = 1_{R+}(f(x_g^i) + N(0, 0.25))
        S ← S ∪ {(x_g^i, y_g^i)}
    end for
end for
return S
```

Table 4: Four different synthetic dataset settings. p is the prior distribution of the four idiosyncratic signals (before normalizing). U is the portion of groups that have idiosyncratic signals in format of (training, validation, test)

	p_{train}	p_{val}	p_{test}	U
1	(1, 1, 1, 1)	(1, 1, 1, 1)	(1, 5, 1, 5)	(0.8, 0.8, 0.8)
2	(1, 1, 1, 1)	(1, 1, 1, 1)	(1, 5, 1, 5)	(0.2, 0.2, 0.8)
3	(0, 1, 1, 1)	(1, 1, 1, 1)	(1, 5, 1, 5)	(0.2, 0.2, 0.8)
4	(1, 1, 0, 0)	(1, 1, 0, 0)	(0, 0, 1, 1)	(0.2, 0.2, 0.8)

6.1.2 Experiment setup on Synthetic Data. As shown in Algorithm 1, in our experiments, the dimension of all signals is 2. The shared predictive signal M is $\mathcal{N}([0, 0], 4I)$ where I is the two-dimensional identity matrix. There are four idiosyncratic signals in W and their mean and variance values are chosen as [([0.25, 0.25], I), ([0.25, -0.25], I), ([-0.25, 0.25], I), and ([-0.25, -0.25], I)] respectively. f is a sine function which takes the sum of all feature dimensions as input. The strength factor a is sampled from a truncated Gaussian distribution $\mathcal{N}(0.75, 0.25, 0, 1)$ and the random noise σ is assumed to be distributed $\mathcal{N}(0, 0.25)$.

Using these parameters, we generate synthetic datasets under four different settings as shown in Table 4. Setting 1 is the one

Table 5: Results on Synthetic Dataset. G: upweighting all samples of a group. M: upweighting misclassified samples of a group. qDRU: upweighting according to ranking percentile. gDRU: upweighting according to ranking index. The results are in the format of (average accuracy, 10th percentile accuracy, worst group accuracy). BOLD: best OOD test performance under the same setting. Underline: second best performance.

Dataset	ERM	GroupDRO	JTT	Const+M	Worst+G	Worst+M	qDRU+G	qDRU+M	gDRU+G	gDRU+M
Setting 1	(70.2, 62.7, 54.7)	(**78.7**, **72.0**, 62.7)	(76.5, 69.3, 62.7)	(67.0, 60.0, 45.3)	(76.6, 69.3, 58.7)	(77.8, **72.0**, 62.7)	(77.2, 70.7, 61.3)	(77.6, 70.7, 64.0)	(75.9, 69.3, 60.0)	(78.5, **72.0**, **65.3**)
Setting 2	(68.1, 61.3, 52.0)	(76.0, 69.3, 57.3)	(73.1, 66.7, 58.7)	(67.6, 61.3, 52.0)	(78.1, 72.0, 62.7)	(76.9, 70.7, 58.7)	(77.7, 72.0, 61.3)	(79.0, **73.3**, **65.3**)	(77.3, 70.7, 58.7)	(**79.5**, **73.3**, **65.3**)
Setting 3	(68.9, 61.3, 50.7)	(75.7, 69.3, 60.0)	(76.3, 70.7, 61.3)	(70.3, 64.0, 56.0)	(77.4, 70.7, 64.0)	(76.3, 69.3, 61.3)	(77.1, 70.7, 62.7)	(**81.0**, **74.7**, **69.3**)	(76.8, 69.3, 60.0)	(76.4, 69.3, 62.7)
Setting 4	(66.9, 60.0, 50.7)	(75.3, 68.0, 60.0)	(75.2, 69.3, 60.0)	(66.9, 60.0, 49.3)	(76.5, 69.3, 61.3)	(76.7, 70.7, 62.7)	(79.4, 73.3, 65.3)	(**80.4**, **74.7**, **68.0**)	(77.3, 70.7, 60.0)	(76.9, 70.7, 62.7)

that exhibits the slightest distribution shift since 80% of groups in the training, validation, and test sets contain the same set of idiosyncratic signals. Setting 2 shows a realistic real-world scenario where the training data uniformly ($p_{train}(w_i) = 1$) contains each of the idiosyncratic signals, but only 20% of the training groups have an idiosyncratic signal. The test data, on the other hand, contains the idiosyncratic signal in 80% of the groups. The third and the fourth settings show substantial distribution shifts since they represent the case where some of the idiosyncratic signals are altogether hidden from the training dataset. This happens routinely in real-world scenarios when the training dataset is not large enough to include all the unique signals introduced by unseen groups in the OOD test data.

We generate 1000 training groups, 500 test OOD groups, and 500 validation OOD groups for each of these settings. Each group contains 75 samples. The base learner for training all these datasets is a three-layer feed-forward neural network with a hidden state size of 128. Each layer is connected by LeakyReLU [27], and a 0.5 dropout rate is applied. We performed a grid search $C \in [5, 10, 20, 50, 100]$ to select the best cutoff for **qDRU**. For all the upweighting methods with constant factors, i.e., **Worst**, **Const**, and **JTT**, λ is selected from the list [2, 3, 4, 5]. The step size for **Group DRO** is chosen as 0.01, and the first and second training steps of JTT were 5. Finally, we performed the model selection using **qDCG@10** metric, which was the best performing metric as we saw in §5.3.

6.1.3 Synthetic Data Results. The results are shown in Table Table 5, and as can be seen, the DRU-based methods significantly outperform the baseline methods in all simulation settings. DRU variants significantly boost the worst group accuracy (the last number in the accuracy lists (90,70,50) in Table 5) by up to 10% in some cases compared to the baselines. Overall, **qDRU+M** is consistently the best DRU-variant except for Setting 1, in which there is only a mild distribution shift. Group DRO and JTT also perform as well as DRU-based methods in Setting 1, but their relative performance drops in settings with significant distribution shifts. Interestingly, the constant upweighting method **Const+M** performs even worse than **ERM** which doesn't perform any weighting at all. When significant distribution shifts are present (especially in setting 3 and 4 where unseen idiosyncratic signals are present in test dataset), **qDRU+M** not only improves the OOD worst group accuracy but also the 10th percentile of worst groups, and it even introduces a considerable improvement (over 6%) in *average* test accuracy compared to ERM, GroupDRO, and JTT baselines. The performance of **qDRU+G** is notable in setting 4 where most distribution shift is present, and its performance is only inferior to **qDRU+M**.

6.2 Results on Real-World Data

6.2.1 Experiment Setup. We follow the lead of the authors of the WILDS Distribution Shift Benchmark Suite [16] and use a finetuned base uncased DistilBERT model as our base learner in all our experiments. We used the following hyperparameters for DistilBERT as also suggested by [16]: batch size 16; learning rate $1x10^{-5}$ for AdamW optimizer [19]; L2-regularization strength 0.01; 5 epochs with early stopping; and a maximum number of 512 tokens. Next, for both **qDRU** and **gDRU**, we performed a grid search to tune the cutoff hyperparameter $C \in [5, 10, 15, 20, 50, 100]$. λ is selected from the list [2, 3, 4, 5] for all methods that performed constant upweighting. For **JTT**, we set the first and second training steps as 5 as suggested by the authors. Finally, for **Group DRO** we fixed the step size as 0.01 following the best practice reported in [16].

6.2.2 Results and Discussion. The results on the validation and test OOD datasets for the various methods are shown in Table 6. The tables report average and 10th percentile group accuracy for completeness, although the worst-group accuracy on the test OOD dataset is the target. Broadly, we see a trend that upweighting methods that use group information outperform those oblivious to the presence of groups (ERM, JTT, Const+M). Among DRU-based methods, **qDRU+M** provides the highest worst group accuracy (target metric) on the OOD test dataset, with a comfortable margin of improvements over **JTT**, **Group DRO**, **Worst**, **Const**. The 10th percentile and average group performance of **qDRU+M** and other DRU-based methods are also competitive or even better than the baselines, suggesting that our soft-minimax based methods improve OOD worst-group performance without a perceptible sacrifice of better-performing groups or average accuracy. It is particularly interesting that the **qDRU+G** model consistently outperforms others on 10th percentile accuracy. We note that real world scenarios are often considerably more complicated than the controlled synthetic setting: individual groups and misclassified examples may be more affected by random noises or even adversarial signals (e.g., spam or fake reviews). In these scenarios, the 10th percentile group accuracy may be a more reasonable target for group robustness and upweighting all examples in a group may be more resilient to noise than only considering misclassified examples.

7 CONCLUSION AND FUTURE WORK

In conclusion, this paper highlights the weakness of the canonical approach in group distributional robustness literature of focusing only on the worst group accuracy for model selection and model training. We introduced a suite of methods inspired by the learning-to-rank literature for group robust model selection and

Table 6: Results on OOD Test (top) and OOD Val (bottom). G: upweighting all samples of a group. M: upweighting misclassified samples of a group. qDRU: upweighting according to ranking percentile. gDRU: upweighting according to ranking index. The results are in the format of (average accuracy, 10th percentile accuracy, worst group accuracy). Bold: best OOD test performance under the same setting.

Dataset	ERM	Group DRO	JTT	Const + M	Worst + G	Worst + M	qDRU + G	qDRU + M	gDRU + G	gDRU + M
Amazon-WILDs	(**71.9**, 53.3, 12.0)	(70.0, 53.3, 8.0)	(71.6, 53.3, 9.3)	(71.0, 53.3, 14.7)	(71.7, 53.3, 14.7)	(70.6,53.3,17.3)	(70.2, **54.7**, 17.3)	(70.1, 53.3, **18.7**)	(70.2, 53.3, 17.3)	(71.5, **54.7**, 14.7)
Yelp	(**63.0**, 52.0, 18.0)	(59.1, 49.0, **27.0**)	(61.7, 51.0, 19.0)	(63.0, **53.0**, 21.0)	(62.8, 52.0, 25.0)	(62.5, **53.0**, 25.0)	(62.6, **53.0**, 23.0)	(62.5, 52.0, 21.0)	(62.8,**53.0**, 24.0)	(62.1, 52.0, **27.0**)
IMDB	(63.2, 42.9, 15.0)	(61.1, 40.4, 15.0)	(60.4, 42.2, 9.0)	(62.6, 44.1, 15.0)	(63.3, 44.1, 22.5)	(63.3, 43.8, 17.5)	(**64.1**, **45.8**, 20.0)	(62.0, 43.3, **25.0**)	(63.2, 43.9, 12.9)	(62.9, 44.4, 17.5)

Dataset	ERM	Group DRO	JTT	Const + M	Worst + G	Worst + M	qDRU + G	qDRU + M	gDRU + G	gDRU + M
Amazon-WILDs	(**72.3**, 54.7, 5.3)	(70.7, 54.7, 5.8)	(**72.5**, 53.3, 5.3)	(71.8, 54.7, 8.0)	(71.7, 54.7, 5.3)	(71.4, 54.7, 8.0)	(70.9, 54.7, 6.7)	(71.1, 54.7, 6.7)	(70.9, 54.7, 6.7)	(72.1, **56.0**, **8.0**)
Yelp	(**64.5**, 54.0, 34.0)	(60.2, 49.0, 31.0)	(63.2, 52.0, 32.0)	(64.4, 53.0, 37.0)	(63.8, 53.1, 38.0)	(63.6, 54.0, 39.0)	(64.0, 53.1, **40.0**)	(64.3, **54.3**, **40.0**)	(63.9, 54.0, 34.0)	(63.6, 54.0, 38.0)
IMDB	(62.6, 43.1, 15.6)	(60.6, 42.3, 15.6)	(59.9, 41.9, 17.9)	(62.5, 44.4, 17.6)	(63.2, 44.4, 15.6)	(63.0, 44.4, 20.0)	(**63.3**, 44.4, 19.3)	(61.9, **46.0**, 15.7)	(62.5, 45.9, **20.2**)	(62.6, 45.2, 15.6)

model training. Essentially, our ranking-based soft minimax approaches smooth the predictive signal learned at training time by performing a discounted weighting which leads to improved generalization performance on the OOD test dataset in the challenging *domain generalization* [16] setting. Our theoretical intuition regarding the fit of ranking-based methods for group robustness is backed by our methods' equally strong empirical performance on synthetic and several real-world benchmark datasets. It turns out that group identities carry a strong predictive signal (even if they do not overlap in training/test) since we observe that group-based approaches perform better than those that ignore the group structure. However, more research needs to be done to investigate this deeply. As part of future work, it will be interesting to study how the number of groups and their correlation structure impact the performance of our ranking-based methods. Learning to rank literature implicitly assumes orthogonality between the search results (or group features) in our case, so it remains to be seen how well our DRU-based approaches perform in the presence of strong group correlations.

REFERENCES

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893* (2019).

[2] Udit Arora, William Huang, and He He. 2021. Types of Out-of-Distribution Texts and How to Detect Them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 10687–10701. https://doi.org/10.18653/v1/2021.emnlp-main.835

[3] Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems* 24 (2011), 2178–2186.

[4] Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic Dialectal Variation in Social Media: A Case Study of African-American English. In *Proceedings of EMNLP*.

[5] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the learning to rank challenge*. PMLR, 1–24.

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

[7] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[8] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[9] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*. PMLR, 1929–1938.

[10] Michiel Hazewinkel. 2001. Minimax principle, Encyclopaedia of mathematics.

[11] Dirk Hovy and Anders Søgaard. 2015. Tagging performance correlates with author age. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)*. 483–488.

[12] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. 2018. Does distributionally robust supervised learning give robust classifiers?. In *International Conference on Machine Learning*. PMLR, 2029–2037.

[13] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[14] Mark G Kelly, David J Hand, and Niall M Adams. 1999. The impact of changing populations on classifier performance. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. 367–371.

[15] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* (2016).

[16] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*. PMLR, 5637–5664.

[17] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*. PMLR, 6781–6792.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*. 3730–3738.

[19] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[20] Nicolai Meinshausen and Peter Bühlmann. 2015. Maximin effects in inhomogeneous large-scale data. *The Annals of Statistics* 43, 4 (2015), 1801–1830.

[21] Aditya Pal, Abhilash Barigidad, and Abhijit Mustafi. 2020. Identifying movie genre compositions using neural networks and introducing GenRec-a recommender system based on audience genre perception. In *5th International Conference on Computing, Communication and Security, ICCCS 2020, Patna, India, October 14-16, 2020*. IEEE, 1–7. https://doi.org/10.1109/ICCCS49678.2020.9276893

[22] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731* (2019).

[23] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[24] Rachael Tatman. 2017. Gender and dialect bias in YouTube's automatic captions. In *Proceedings of the first ACL workshop on ethics in natural language processing*. 53–59.

[25] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. (2011).

[26] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[27] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).