Decomposing the Group DRO Advantage:

Balanced Sampling vs Adversarial Reweighting on Waterbirds

A Reproduction Study

Rayyan Huda
Systems Design Engineering
University of Waterloo

Abstract

Neural networks trained with standard empirical risk minimization (ERM) can achieve strong average accuracy while failing on subpopulations where a spurious correlation in the training data does not hold. Group Distributionally Robust Optimization (Group DRO) addresses this failure mode by combining two different mechanisms: a group-balanced training sampler and an adversarial, worst-group-weighted loss. Published evaluations generally report only their combined effect, so the individual contribution of each mechanism remains unclear. I reproduced Group DRO on the Waterbirds benchmark, training a ResNet-50 under the paper's reference configuration, and isolated each mechanism's contribution by additionally training a third configuration that uses the balanced sampler alone, with no adversarial reweighting. My ERM and Group DRO baselines obtain 97.25% average / 68.54% worst-group test accuracy and 95.40% average / 86.14% worst-group accuracy, respectively; a +17.60-point worst-group improvement at a cost of 1.85 points of average accuracy, which matches the shape, though not the exact magnitude of the original paper's result. The balanced-sampling-only ablation reaches 83.02% worst-group accuracy, recovering 82% of Group DRO's total worst-group gain over ERM; adding adversarial reweighting produces the remaining 18% of the improvement in worst-group accuracy, with the additional gain concentrated on the rarest training group. This result comes from a single training run per configuration, due to the lack of compute available for this project. It is not a claim of statistical certainty. On this benchmark, balanced sampling alone recovers most of Group DRO's practical benefit at a fraction of its implementation complexity, which describes a larger risk in evaluating combined interventions: a method's reported aggregate effect can obscure how much of that effect comes from its simplest component.

1 Introduction

Standard empirical risk minimization (ERM) optimizes average performance over a training distribution. When that distribution contains a spurious correlation between a nuisance attribute and the label, ERM

models will exploit it. On the Waterbirds benchmark, background scenery (land vs. water) is correlated with bird type (landbird vs. waterbird) for roughly 95% of the training set, so a model that partly relies on background texture rather than the bird itself will still score well on average and score badly on the minority groups where the correlation reverses. Sagawa et al. (2020) introduced Group Distributionally Robust Optimization (Group DRO) to address exactly this failure mode: rather than minimizing average training loss, Group DRO minimizes the loss of the worst-performing group, adaptively reweighting groups during training via an exponential-gradient update.

This paper has two goals. The first is a reproduction: I implement Group DRO from the paper and the reference implementation released alongside it (Koh and Sagawa, 2020), train it on Waterbirds via the WILDS benchmark package (Koh et al., 2021), and compare my results against the paper's. The second goal is an extension. Group DRO bundles two distinct mechanisms into one algorithm: (i) a group-balanced sampler, so that every group is seen with roughly equal frequency during training regardless of its true count in the dataset, and (ii) an adversarial loss that additionally up-weights whichever group currently has the highest loss. The published algorithm and its evaluation report the combined effect of both mechanisms together. I ask a narrower question: on Waterbirds specifically, how much of Group DRO's benefit is attributable to the sampler alone, before the adversarial loss is even introduced? Answering this matters because the two mechanisms have very different costs; a weighted sampler is a few lines of code with no change to the training loop or loss function, while the adversarial reweighting requires tracking per-group losses and an extra optimization step every iteration.

My reproduction reaches 86.14% worst-group test accuracy under Group DRO, against 68.54% for ERM, a +17.60-point improvement. Removing the adversarial loss and keeping only the balanced sampler still reaches 83.02%, a +14.48-point improvement, 82% of the full gain. Adding the adversarial loss produces a further +3.12-point improvement in worst-group accuracy, and this additional gain is not spread evenly across groups: it concentrates almost entirely on the rarest training group. I take this as more informative rather than accusing Group DRO's design: the adversarial component is important, but most of the worst-group-accuracy number that gets reported for "Group DRO" as a headline result is coming from a much simpler mechanism sitting inside it.

2 Background

2.1 Problem Setup

Consider training data drawn from a distribution over *(x, y, a)*, where x is an input, y is a label, and a is a nuisance attribute unavailable at test time in the sense that it should not influence the prediction. A *group* is a pair $g = (y, a)$; on Waterbirds, y ∈ {landbirds, waterbirds} and a ∈ {land background, water background}, giving four groups. ERM minimizes the loss averaged over the training distribution, which is mainly dominated by the majority groups (landbirds-on-land and waterbirds-on-water). Group DRO instead minimizes the worst-case loss over groups:

$$\min_{\theta} \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim P_g} \left[\ell(\theta; x, y) \right] \quad (1)$$

2.2 The Group DRO Algorithm

Sagawa et al. (2020) solve this minimax problem online, maintaining an adversarial weight q_g per group and updating it each step by exponentiated gradient ascent on the observed per-group loss, then taking a gradient descent step on the resulting weighted loss:

Two implementation details matter for my reproduction. First, the reference configuration additionally uses a `-reweight_groups` flag, which draws training minibatches from a sampler weighted so that every group is seen with equal expected frequency, independent of the adversarial weights q_g; this is the "balanced sampling" mechanism I isolate in Section 5. Second, a *generalization adjustment* term can be added per group to account for groups generalizing differently at a fixed training loss; the reference Waterbirds configuration sets this adjustment to zero, and I do the same.

Algorithm 1 Group DRO training setup (following Sagawa et al. (2020), Algorithm 1)

Initialize adversarial weights $q_g \leftarrow 1/|\mathcal{G}|$ for all $g \in \mathcal{G}$
for each minibatch B, drawn with group-balanced sampling do
    for each group g present in B:
        compute mean loss $\hat{\ell}_g \leftarrow \mathbb{E}_{(x,y)\in B\cap g}\left[\ell(\theta; x, y)\right]$
        $q_g \leftarrow q_g \exp(\eta_q \hat{\ell}_g)$ (exponentiated-gradient ascent, step size η_q)
    end for
    normalize q so that $\sum_g q_g = 1$
    take a gradient descent on θ using the loss $\sum_g q_g \hat{\ell}_g$
end for

2.3 Applications

Group shift robustness is relevant wherever a model can achieve strong average performance by relying on a correlation that does not hold, or reverses, on some known subpopulation: demographic subgroups in clinical risk prediction, dialect or register in NLP models trained on web text, or image source acting as a confounder for the true content of an image in downstream classification tasks. Waterbirds is a controlled synthetic testbed for this class of problem specifically because the spurious attribute (background) is known and labeled.

2.4 Related Work

Group DRO belongs to a broader line of work on distributional optimization, which replaces an average-case training objective with a worst-case objective over an uncertainty set of distributions (Duchi and Namkoong, 2018); Group DRO specializes this framework to uncertainty sets defined by a small number of known, labeled groups.

The specific question this paper asks - how much of a group performance method's benefit comes from simply changing which examples a model sees, versus changing how the loss weights them - connects to related works in other literature. Byrd and Lipton (2019) study the effect of importance weighting on deep

networks directly and find that its influence on the learned decision boundary can diminish over the course of training, particularly under weight decay, which is a relevant caution when interpreting any resampling-based result, including mine, as a fixed, purely additive contribution rather than one that can depend on training dynamics. Separately, work on class-imbalanced classification consistently finds that data-level rebalancing, oversampling minority classes, or reweighting their loss contribution captures a large share of the benefit of more elaborate and intricate loss-design interventions (Buda et al., 2018; Cui et al., 2019), a pattern that is consistent with the 82% figure I report for Waterbirds specifically. Most directly, Idrissi et al. (2022) report that simple data-balancing strategies are competitive with, and sometimes exceed, more sophisticated group-generalization methods across different standard benchmarks, including Waterbirds; my ablation offers a complementary mechanism-level view of the same broad conclusion, decomposing one specific algorithm's own gain rather than comparing separately tuned methods against one another.

3 Experimental Setup

3.1 Dataset

Waterbirds (Sagawa et al., 2020) combines bird photographs from CUB-200-2011 (Wah et al., 2011) with background scenes from the Places dataset (Zhou et al., 2017), pasting each bird onto a land or water background. I accessed the dataset through the WILDS package (Koh et al., 2021), which provides the canonical train/val/test split and a grouper utility for computing per-group metrics. Table 1 reports the group counts from my own audit of the loaded dataset, which match the split sizes reported in the original paper exactly. Further access and reproducibility details are given in Appendix A.

Table 1: Waterbirds group counts by split. The training set is ~95% background-correlated; validation and test are constructed to be background-independent (identical land/water split within each label), which is what makes worst-group accuracy on val/test a strong measure for reliance on the spurious background correlation.

Split	Landbird/Land	Landbird/Water	Waterbird/Land	Waterbird/Water	Total
Train	3,498	56	184	1,057	4,795
Val	467	133	466	133	1,199
Test	2,255	642	2,255	642	5,794

The rarest training group, waterbird-on-land, has only 56 examples, which is a small enough count that single-seed variance on that group's accuracy specifically should be expected and is discussed as a limitation in Section 6.2

3.2 Model and Optimization

I train a ResNet-50 (He et al., 2016) from an ImageNet-pretrained initialization, matching the reference configuration: SGD with momentum 0.9, learning rate 10^{-3}, weight decay 10^{-4}, batch size 128, for 300 epochs, with no data augmentation. This configuration is held identical across every arm reported in this paper (ERM, Group DRO, and the reweight-only ablation in Section 5) so that the only variable between arms is the training distribution and loss, not any hyperparameter difference. All training was run on a

Kaggle-provided NVIDIA T4X2 instance; each 300-epoch run took approximately 7-7.5 hours. A full hyperparameter table is given in Appendix A.2.

3.3 Model Selection

Two natural checkpoint-selection criteria exist: the epoch with the best validation *average* accuracy, and the epoch with the best validation *worst-group* accuracy. I report both throughout because, as Section 4.1 shows, the choice is not a minor detail: it moves test worst-group accuracy by several points independent of the training algorithm, reproducing a methodological point made in the original paper's own appendix. Unless stated otherwise, comparisons between arms (ERM vs. Group DRO vs. the ablation) use the best-val-worst-group-accuracy checkpoint for each arm, since that is the selection criterion the original paper uses for its headline Table 1 numbers.

4 Reproduction Results

4.1 ERM Baseline

Table 2 reports my ERM reproduction against the paper's Table 1 ERM row. The first two rows differ only in the checkpoint-selection criterion; both are the same trained model at different epochs.

Table 2: ERM baseline, test accuracy

Model selected by	Avg. accuracy	Worst-group accuracy
Rayyan - best val-worst-group acc.	97.25%	68.54%
Rayyan - best val avg. acc	97.33%	60.12%
Sagawa et al. (2020), Table 1	97.3%	72.6%

Average accuracy matches the paper almost exactly (97.25-97.33% vs. 97.3%). Worst-group accuracy matches the original result closely and shows the expected large average/worst-group gap (~29 points) but sits about 4 points below the paper's reported figure; I attribute this to single-seed variance on a 56-example group and discuss it further in Section 6.2. Per-group test accuracy for the worst-group-selected checkpoint was: landbird/land 99.4%, landbird/water 72.0%, waterbird/land 68.5% (the minimum), waterbird/water 96.0%, which confirmed that essentially all of ERM's failure is concentrated on one small, background-conflicting group, exactly as the spurious correlation idea predicts.

My independent finding here is that switching the selection criterion from best-val-avg to best-val-worst-group moved test worst-group accuracy by 8.4 points (60.12% to 68.54%) while average accuracy barely moved at all (97.33% to 97.25%). A model with a better worst-group was sitting in the same training run the entire time, but only the checkpoint-selection rule determined whether it was the one reported.

4.2 Group DRO

Table 3 reports the Group DRO reproduction, trained with the balanced sampler and the exponential-gradient loss described in Section 2.2 (step size $\eta_q = 0.01$, generalization adjustment 0, which matches the reference Waterbirds configuration).

Table 3: Group DRO, test accuracy

Model selected by	Avg. accuracy	Worst-group accuracy
Rayyan - best val-worst-group acc.	95.40%	86.14%
Rayyan - best val avg. acc	97.57%	70.72%
Sagawa et al. (2020), Table 1 (standard reg.)	97.4%	76.9%

Comparing to my own ERM baseline is essentially the reproduction claim, and it is unambiguous: Group DRO improves worst-group accuracy by +17.60 points (68.54% to 86.14%) at a cost of only 1.85 points of average accuracy (97.25% to 95.40%). This is a reproduction of the paper's central thesis: that mainly optimizing for the worst group will improve performance for that group at a small average-accuracy cost.

I don't compare against the paper's reported Group DRO numbers, and I do it plainly, because it is not seen as a win: my worst-group accuracy exceeds the paper's reported 76.9% by about 9 points, while my average accuracy comes in about 2 points below their 97.4%. I read this as landing at a different point on the same accuracy/performance trade-off curve rather than simply a better result. The paper's number is very likely averaged over multiple seeds, whereas mine is a single run; small implementation details, initialization, the exact batch composition the sampler produces, and minor numerical differences in the exponential-gradient update can shift where a single run lands on that curve. Per-group test accuracy for the worst-group-selected checkpoint was landbird/land 96.7%, landbird/water 87.9%, waterbird/land 86.1% (the minimum), and waterbird/water 92.8%, which was more balanced across groups than the ERM checkpoint above.

5 Extension: Decomposing the Group DRO Advantage

5.1 Motivation

Group DRO's headline number combines two mechanisms: a group-balanced training sampler and an adversarial loss reweighting. These are not the same kind of intervention. The sampler changes only which examples a fixed loss function sees, and is a direct drop-in replacement for `shuffle=True` in a standard data loader. The adversarial loss changes the loss function itself, requires tracking a per-group running weight, and adds an extra update step every iteration. If most of group DRO's practical worst-group-accuracy gain turns out to come from the sampler alone, that is a useful thing to know: it would mean a substantially simpler and cheaper intervention captures most of the benefit that gets reported under the more complex algorithm's name.

5.2 Method

I train a third arm, identical in every respect to the Group DRO run in Section 4.2: the same architecture, hyperparameters, 300-epoch run, group-balanced sampler, except that the loss function is plain cross-entropy rather than the exponential-gradient loss. This measures the effect of introducing balanced

sampling while holding the loss function equal to the ERM loss: any difference between this and full ERM is attributable to balanced sampling alone, since the loss function is unchanged from ERM, and the remaining gap between the balanced-sampling-only arm and full Group DRO measures the effect of adding adversarial reweighting on top of balanced sampling, since the sampler is unchanged from Group DRO. Because the study does not include an adversarial-loss-only condition, this comparison does not separately identify interaction effects between the two mechanisms. This ablation also has a practical advantage: because the training loop never needs to route group indices into the loss computation, it does not encounter the CPU/GPU device-placement issue that the full Group DRO implementation required a fix for (Appendix A.2), and the run was completed without any errors on the first try.

5.3 Results

Table 4 places all three arms side by side, using the best-val-worst-group checkpoint for each, which is consistent with how Sections 4.1 and 4.2 were read.

These are the main findings of my paper. Of Group DRO's total +17.60-point worst-group improvement over ERM, balanced sampling alone recovers +14.48 points (82% of the total) before the adversarial loss is introduced. The adversarial reweighting contributes the remaining +3.12 points, or 18% of the total gain. The average-accuracy cost splits more evenly between the two mechanisms: -0.97 points from the sampler, and a further -0.88 points from adding the exponential-gradient loss on top.

Table 4: Decomposing Group DRO's worst-group accuracy gain. "Balanced-sampling-only" uses the same balanced sampler as Group DRO but plain cross-entropy loss, with no adversarial reweighting.

Arm	Avg. accuracy	Worst-group accuracy	Δ avg. vs ERM	Δ worst-grp. vs ERM
ERM (no reweighting, plain cross-entropy)	97.25%	68.54%	–	–
Balanced-sampling-only	96.28%	83.02%	-0.97 pts	+14.48 pts
Full Group DRO	95.40%	86.14%	-1.85 pts	+17.60 pts

5.4 Where the Adversarial Loss' Contribution Concentrates

The 18% share is small, but it is not spread evenly, and it's important to notice where it lands. Per-group test accuracy for the balancing-sampling-only checkpoint was landbird/land 98.0%, landbird/water 88.3%, waterbird/land 83.0% (the minimum), and waterbird/water 92.8%. Comparing this to the full Group DRO per-group numbers in Section 4.2: three of the four groups move by a point or less between reweight-only and full Group DRO (landbird/land 98.0 to 96.7, landbird/water 88.3 to 87.9, waterbird/water 92.8 to 92.8, unchanged), while the rarest training group, waterbird/land, improves by a stronger 3.1 points (83.0% to 86.1%). This is the behavior the algorithm was designed to produce: balanced sampling already ensures that every group is seen equally often per epoch, but the adversarial weight q_g adds onto that by biasing the loss toward whichever group is currently struggling the hardest, and on Waterbirds, that is consistently the smallest, most background-conflicted group. The adversarial mechanism is working on top of a training distribution that balanced sampling has already mostly fixed.

5.5 Framing the Results

This result does not diminish Group DRO: the full pipeline still wins over the balanced-sampling-only ablation (86.14% vs. 83.02% worst-group accuracy), and the additional adversarial mechanism is not redundant. What the decomposition shows is that, on Waterbirds specifically, the balanced-sampling-only configuration achieves most of the worst-group improvement in the full Group DRO configuration. This paper does not address whether this 82%/18% split generalizes to other group-shift benchmarks with different group-size distributions. Section 6.2 discusses it as a limitation.

6 Discussion

6.1 Practical Implications

For any machine learning engineer facing a group-shift problem similar to Waterbirds, with a small number of labeled groups and one or two underrepresented, these results suggest that balanced sampling may be a useful baseline intervention to evaluate before implementing the full Group DRO procedure. This conclusion is specific to the benchmark and configuration studied here and should not be interpreted as evidence that balanced sampling generally substitutes for adversarial reweighting.

6.2 Limitations

Every number in this paper comes from a single training run per configuration, not an average over multiple seeds. This is a consequence of compute constraints: each 300-epoch run took 7-7.5 hours on a Kaggle-provided GPU, and running the three configurations already reported here consumed roughly 21.6 GPU-hours over the course of this project. The paper's own Table 1 numbers are, based on standard practice for this kind of benchmark, likely averaged over multiple seeds, which is the most probable explanation for the gap between my ERM worst-group number (68.54%) and theirs (72.6%), the rarest training group only had 56 examples, and a single run's worst-group accuracy on a group that small is expected to carry seed-to-seed variance, however, this study does not quantify that variability.. The decomposition result in Section 5 is the paper's most notable contribution, and it is also the one most exposed to this limitation: a repeat with additional seeds would be the next step to confirm the 82%/18% split is a stable property of this benchmark and not just the result of one run's initialization.

A second limitation is scope: this paper examines one benchmark (Waterbirds) with one architecture (ResNet-50) at one set of hyperparameters (the paper's reference configuration). The original paper's Section 3 shows that Group DRO's benefit is sensitive to regularization strength in ways this reproduction does not explore; I did not run the strong-ℓ_2-penalty configuration from that section, and I make no claim about how the sampling/adversarial-loss decomposition would behave under it.

7 Conclusion

I reproduced the behavior of Group DRO on Waterbirds, obtaining an improvement in worst-group test accuracy from 68.54% under ERM to 86.14% under the full Group DRO configuration. The reproduction does not match the original result numerically, but it preserves the main trade-off: a higher worst-group accuracy followed by a smaller reduction in average accuracy.

Decomposing the training procedure into balanced sampling and adversarial reweighting shows that the balanced-sampling-only configuration achieves 83.02% worst-group accuracy, capturing 82% of the total improvement from ERM to full Group DRO. Adding adversarial reweighting produces a further 3.12-point improvement, concentrated on the rarest training group, the waterbird/land group.

These findings indicate that balanced sampling accounts for most of the improvement in this single-seed Waterbirds experiment, while the adversarial mechanism provides an additional benefit. Because the study uses one benchmark, one architecture, one hyperparameter configuration, and one training run per arm, the 82%/18% decomposition should be treated as an empirical observation rather than a general property of Group DRO.

Reproducibility Statement

All training configurations, dataset audits, and per-group results reported in this paper were produced by the author on Kaggle-provided NVIDIA T4X2 instances, using the WILDS package for data loading and group evaluation and the implementation of the Group DRO loss following Sagawa et al. (2020) and the reference implementation (Koh and Sagawa, 2020). All three training configurations (ERM, Group DRO, balanced-sampling-only) share identical architecture, optimizer, learning rate, weight decay, batch size, and epochs, differing only in the sampler and loss function as described in Section 3-5. Every reported number reflects a single training run; no number in this paper is estimated or drawn from any source other than the authors' own measured runs. Full dataset and training configuration details are given in Appendix A.

References

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. In International Conference on Learning Representations (ICLR), 2020. arXiv:1911.08731.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In International Conference on Machine Learning (ICML), 2021.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple Data Balancing

Achieves Competitive Worst-Group-Accuracy. In Conference on Causal Learning and Reasoning (CLeaR), 2022.

Jonathon Byrd and Zachary Lipton. What is the Effect of Importance Weighting in Deep Learning? International Conference on Machine Learning (ICML), pp. 872–881, 2019.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks. Neural Networks, 106:249–259, 2018.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9268–9277, 2019.

John Duchi and Hongseok Namkoong. Learning Models with Uniform Performance via Distributionally Robust Optimization. arXiv preprint arXiv:1810.08750, 2018.

Pang Wei Koh and Shiori Sagawa. group_DRO: reference implementation accompanying Sagawa et al. (2020). https://github.com/kohpangwei/group_DRO

A Experimental Details

This appendix consolidates dataset and training configuration details for reproducibility. Unlike Sagawa et al. (2020), this paper includes no separate proofs appendix, since it derives no new theoretical results, and no supplementary-experiments appendix, since every experiment run for this project is already reported in the main text above.

A.1 Dataset

The Waterbirds dataset was accessed through the `wilds` PyPI package, which provides data loading, the train/val/test split, and a `CombinatorialGrouper` utility for computing group membership and per-group metrics; it does not itself provide an ERM or Group DRO training loop. Group indices follow the formula $g = a + 2y$, verified from the `CombinatorialGrouper` source: $g = 0$ is landbird/land, $g = 1$ is landbird/water, $g = 2$ is waterbird/land, and $g = 3$ is waterbird/water. Split-level group counts are reported in Table 1. To avoid repeated downloads across training runs in numerous notebooks, the dataset was cached once as a Kaggle Dataset and reused as an input to every subsequent notebook.

A.2 Model and Training

Table 5 lists the full training configuration, held identical across all three arms reported in this paper (ERM, Group DRO, and the balanced-sampling-only ablation) except for the sampler and loss function

Table 5: Full training configurations, all arms

Setting	Value
Architecture	ResNet-50, ImageNet-pretrained initialization

Optimizer	SGD, momentum 0.9
Learning rate	10^{-3}
Weight decay	10^{-4}
Batch size	128
Epochs	300
Data augmentation	None
Robust loss step size η_q (Group DRO only)	0.01
Generalization adjustment (Group DRO only)	0
Hardware	Kaggle NVIDIA T4X2
Time per 300-epoch run	~ 7 - 7.5 hours
Data loading/group evaluation	`wilds` package (pip)

Two implementation issues encountered during this reproduction are recorded here since they were not obvious from the paper or the WILDS documentation, and could prevent a reproduction of the reported training and evaluation procedure. First, WILDS' group-wise evaluation path imports `torch_scatter`, an undeclared dependency not installed by a plain `pip install wilds`; rather than following my footsteps and chasing a version-matched wheel, I reimplemented the group-mean reduction with `torch.scatter_add_`, which ships with PyTorch. Second, the WILDS grouper's internal group-membership tensor is constructed on CPU and was never moved, so calling it on GPU inside the training loop raises a device-mismatch error; the fix is to compute group membership while the metadata is still on CPU and then move the resulting group-index tensor to the training device. Neither issue affects the reported numbers.