Robust Representation Learning for Group Shifts and Adversarial Examples

Ming-Chang Chiu
University of Southern California
mingchac@usc.edu

Xuezhe Ma
Information Science Institute
xuezhema@isi.edu

Abstract

Despite the high performance achieved by deep neural networks on various tasks, extensive research has demonstrated that small tweaks in the inputs can lead to failure in the model's predictions. This issue affecting deep neural networks has led to a number of methods to improve model robustness, including adversarial training and distributionally robust optimization. Although both of these two methods are geared towards learning robust models, they have essentially different motivations: adversarial training attempts to train deep neural networks against perturbations, while distributional robust optimization aims to improve model performance on the most difficult "uncertain distributions". In this work, we propose an algorithm that combines adversarial training and group distribution robust optimization to improve robust representation learning. Experiments on three image benchmark datasets illustrate that the proposed method achieves superior results on robust metrics without sacrificing much of the standard measures.

1 Introduction

Deep neural networks (DNNs) have been demonstrated to significantly improve the benchmark performance of a wide range of application domains, including computer vision [22], speech [7], and natural language processing [13]. However, extensive studies have shown that deep neural networks, trained via empirical risk minimization (ERM), are vulnerable: some small and carefully-crafted perturbations in the input space can cause malfunctions and huge performance drops [18, 26]. The essential reason behind performance drop is that the models rely on *weakly correlated* or *spurious correlations* [52] — heuristics between labels and inputs that hold for most training examples, but are not inherent to the task of interest, such as the strong associations between the background and the label on the Waterbirds dataset [54] (Figure 2 in the Appendix).

Adversarial training (AT) [50, 18, 24, 34] is by far one of the most effective ways to learn models against small perturbations [41, 35]. The idea behind AT is simple and straight-forward — adding adversarial noise to the input space during training and therefore achieving better *adversarial robustness* than models trained without AT (Appendix B). Previous works have shown various advantages of AT, including mitigation of the performance drop on noisy input [34, 42] or use as a regularization technique [36].

Another general line of approach for learning robust models is distributionally robust optimization (DRO) [2]. Instead of learning to minimize an ERM objective, DRO aims at *distributional robustness* via optimizing the performance on the worst-case distributions (Appendix B). Previous works have

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

demonstrated that DRO is certified to be effective against small perturbations. For example, in [48], adversarial robustness is cast as a form of distributional robustness in a Wasserstein ball. In this work we will study *group DRO*, which has been shown to reduce reliance on spurious correlations [44, 63]. From a view of representation learning, AT and group DRO attempt to improve model robustness of difference aspects: *adversarial robustness* and *group-distributional robustness*. A critical question is whether we can develop a model to incorporate both of the two types of robustness at the same time.

In the following sections, we explore the connections between the two types of robustness of AT and group DRO, and propose the *Adversarial group DRO* algorithm, which leverages the advantages of both to further improve model robustness. More specifically, we leverage the pre-selected group knowledge in group DRO and the projected gradient descent to learn a group mixture distribution formulated as a minimax problem, robust to group shifts and perturbations. Experimental results on two datasets with pre-selected spurious features — Waterbirds [54, 44] and CelebA [33] datasets — demonstrate the effectiveness of the proposed algorithm.

Our contributions can be summarized as follows,

- We propose *Adversarial group DRO*, an efficient online optimization algorithm that combines group DRO and AT to improve model robustness.
- Our algorithm shows superior results than simply employing either AT or group DRO, and can mitigate performance drop for robust models on standard dataset like CIFAR-10, showing different types of robustness can be complementary.
- We provide intuitions and supporting evidence on *the learned robust representations* through various types of analysis.

2 Proposed algorithm

In this section, we set up the basic notations and then describe the proposed algorithm. We denote $\mathcal{D}$ as the dataset, and $\langle x, y\rangle$ as a data sample (the image and the corresponding label). $f(\cdot;\theta)$ denotes a deep neural network, which takes an $\langle x, y\rangle$ pair as input. θ is the set of parameters of the neural network. $\mathcal{L}(\cdot,\cdot)$ denotes a generic loss function (e.g., cross-entropy loss).

The proposed *Adversarial group DRO* (see the Appendix for full algorithm) algorithm combines group DRO and AT (cf. Appendix B for background knowledge) to incorporate both adversarial and distributional robustness. Our plan is to train the model under a dynamically changing group mixture distribution where the constituent distributions are adversarially perturbed. Thus, our model is exposed to both *distributional shifts* (in our case *group shifts*) and *adversarial perturbations*.

2.1 Relation between Adversarial Training and DRO

We emulate Eq. (7) to combine DRO and AT and study the connections of the two types of robustness. We add perturbations into the DRO setup (Eq. (5)), and then find the model that optimizes the risk over all the maximally perturbed uncertain distributions:

$$\min_{\theta} \left\{ R(\theta) := \sup_{Q\in\mathcal{Q}} \mathbb{E}_{(x,y)\sim Q} \left[\max_{\delta\in\Delta} \mathcal{L}(f(x+\delta), y; \theta) \right] \right\} \tag{1}$$

In our case, DRO carries the group-mixture distributions, so Eq. (1) can be together considered with Eq. (6) and find the group adversarial model

$$\theta_{AdvDRO} = \operatorname*{argmin}_{\theta} \left\{ \max_{g\in\mathcal{G}} \mathbb{E}_{(x,y)\sim P_g} \left[\max_{\delta\in\Delta} \mathcal{L}(f(x+\delta), y; \theta) \right] \right\}. \tag{2}$$

2.2 Adversarial group DRO algorithm

Training group DRO and AT jointly can be tricky, as previous works fail for *group DRO* due to the difficulty in gradient estimation in a stochastic fashion [17, 20, 44] or assume convexity and therefore are not generalizable [48]. We propose an online algorithm that provides an efficient way to train Eq. (2). Due to space limit, we describe the algorithm 1 in the Appendix.

Table 1: **Test results (%).** Our *adversarial group DRO* algorithm improves on the robust metrics on both clean and perturbed test set without sacrificing much of the average accuracy. Note that ERMs do not know group information during training. The difference between Batch and Group is that in Batch we do not consider group information for adversarial perturbation, we perturb the whole batch uniformly, while in Group we take group weights into account to learn adversarial noises.

Metric	Perturbation	CIFAR-10				Waterbirds				CelebA			
		ERM		GDRO		ERM		GDRO		ERM		GDRO	
		w/o AT	AT	w/o AT	AT	w/o AT	AT	w/o AT	AT	w/o AT	AT	w/o AT	AT
Average Acc.	Batch	92.8	91.2	91.9	91.2	97.3	97.3	96.4	96.1	95.8	96	94.8	95.2
	Group (ours)	-	-	**92.3**	**92**	-	-	96.2	96	-	-	94.8	95.2
Adversarial Acc.	Batch	73.4	87	67.3	87.7	0	38.3	0.6	32.4	73	95.1	36.1	95.3
	Group (ours)	-	-	**69.9**	**88.2**	-	-	0.2	**33.6**	-	-	29.4	93.5
Robust Acc.	Batch	87.2	84	86.5	82.8	73.5	75.2	86.2	86.2	70.5	73	86.6	86.6
	Group (ours)	-	-	85	**86.2**	-	-	85.8	**89.1**	-	-	**90.8**	**86.6**
Robust Adv. Acc.	Batch	53.7	77	53.6	78.6	2.2	55.5	17.8	60.8	1.6	37.7	5.5	83.3
	Group (ours)	-	-	**56**	**79.2**	-	-	**17.9**	**64.5**	-	-	**10.2**	**83.8**

Building on top of existing algorithms for *group DRO* [44] and AT [31, 30], Algorithm 1 leverages prior knowledge of group information and learns which groups to amass stronger perturbations. Typical AT adds perturbations to the input space uniformly, while our algorithm performs the AT phase and optimizes the DRO part in turns, which allows us to update the q distribution over groups and weigh perturbations. Essentially, we are learning an adversarial distribution that generates the strongest perturbations to add to each group.

Note that we can also rewrite Eq. (2) as

$$\hat{\theta}_{AdvDRO} = \underset{\theta}{\operatorname{argmin}} \left\{ \max_{q \in \mathcal{Q}} \sum_{g=1}^{m} q_g \mathbb{E}_{(x,y)\sim \hat{P}_g} \left[\max_{\delta \in \Delta} \mathcal{L}(f(x+\delta), y; \theta) \right] \right\}, \tag{3}$$

where $\mathcal{Q} = \{\sum_{g=1}^{m} q_g P_g : \sum_{g=1}^{m} q_g = 1, q_g \geq 0 \ \ \forall g \in \mathcal{G}\}$, so in practice, we can use mini-batches which contain a mixture of different groups. And in an end-to-end manner, the algorithm dynamically learns to perform under an "uncertain distribution" perturbed and mixed with groups and encodes the *group-distributional robustness* and *adversarial robustness*.

A representation learning view. Although AT and group DRO achieve different kinds of robustness, they both aim to learn robust models. From a representation learning perspective, the model adopting these approaches together should learn correlations that rely less on the spurious ones — either explicit (as in Figure 2) or implicit (innate within the dataset or added by adversarial noise).

3 Experiments

3.1 Experimental Setup

Datasets. To demonstrate the effectiveness of the *Adversarial Group DRO* (Algorithm 1), we conduct extensive experiments on three image benchmark datasets Waterbirds, CelebA & CIFAR-10, and study the connections between *adversarial robustness* and *group-distributional robustness*. In Appendix A, we detail the datasets.

Methods. We train models with objectives described in Section 2 & B, which are (1) *ERM*, (2) *adversarial ERM* (advERM, i.e., AT), (3) *group DRO* (GDRO), and (4) our algorithm *Adversarial group DRO* (advGDRO). We expect the models to gain *adversarial robustness* from ERM to advERM (so is GDRO to advGDRO) and gain an additional *distributional robustness* from advERM to advGDRO, and thus continuously rely on fewer spurious correlations moving from ERM to advGDRO. For convenience, we will simply call *group DRO* as DRO in the following sections. For full implementation details, see Appendix C.

Robust metrics. We evaluate on average accuracy and average adversarial accuracy to compare *adversarial robustness*. For *distributional robustness*, we use robust accuracy, which can be quantified by measuring the worst-case performance among all groups. Finally, we measure robust adversarial

accuracy for a combined *adversarial* and *distributional robustness*. These metrics are of interest for ERM, AT, GDRO and advGDRO. If we can improve on robust adversarial accuracy, then we essentially will improve both types of robustness.

3.2 Comparisons and Analysis

We first compare advGDRO with the three baseline methods, i.e. ERM, AT and GDRO, on the three benchmark datasets to illustrate the benefits of *adversarial group DRO*. The experimental results are summarized in Table 1 & 2 (in the Appendix). Recall that subgroup information is available to models only *during training*.

DROs achieves better group-distributional robustness over ERMs and advGDRO further achieves adversarial robustness. Across the datasets, advGDRO gains at most 13.9% over advERM on robust accuracy; and GDRO improves over ERM by up to 15.7% and advDRO gains at most 46.1% over advERM on robust adversarial accuracy. The improvements show that advGDRO achieves superior robustness on both clean and perturbed data, and verifies that DROs guarantee better robust performances. The fact that advGDRO consistently outperform other methods on the robust adversarial accuracy demonstrates the effectiveness of our algorithm in improving both types of robustness.

***Adversarial group DRO* mitigates performance gap.** Another interest of our work is to mitigate the performance drop that comes with AT [34] on a standard dataset like CIFAR-10. We observe that with mild perturbation our algorithm mitigates the gap on average accuracy from 1.6% to 0.8%. In addition, our algorithm surprisingly improves the adversarial accuracy over advERM by 1.2%, where advERM is designed to optimize against adversarial perturbations.

Incorporating group weights increases adversarial robustness. A key benefit of Algorithm 1 is to leverage group information to learn the adversarial distribution for Eq. (3); however, our algorithm also has the flexibility of perturbing without group weights. To illustrate the effect of group information, we compare the models that are trained with and without group updates. Table 1 & 2 (in the Appendix) show improvements on GDRO and the efficacy of using group updates for perturbation is most obvious when combined with advGDRO. When group information is incorporated, the performance on both worst-group robust measures is consistently improved. The robust accuracy group updates reach a performance gain up to 3.4% and 3.7% on robust adversarial accuracy.

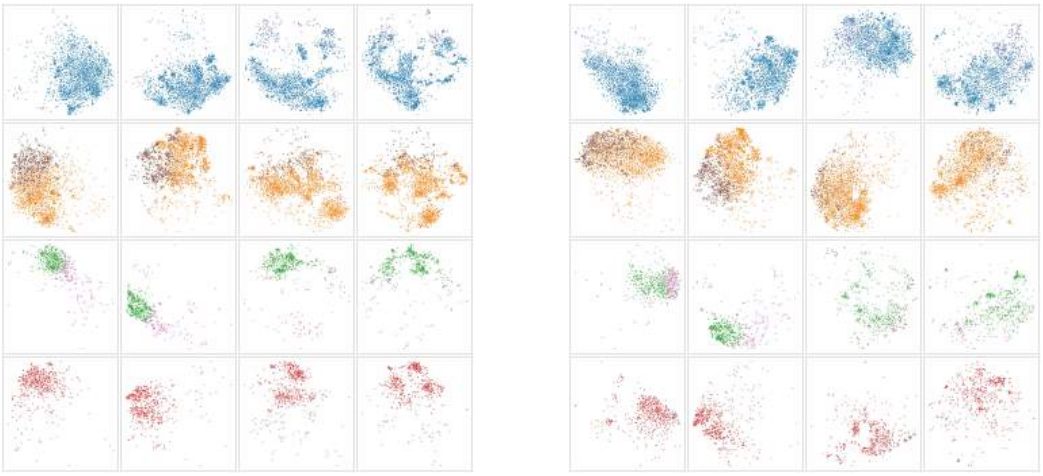
4 Visualization Analyses

We discuss the effect of our *Adversarial group DRO* algorithm through the lens of representations in this section. Furthermore, we show test examples corrected by more robust models to analyze what is driving the improvements, and plot CNN kernels to see what representations the models learn and draw connection with [55] in Appendix F & G respectively.

Representation changes show learning to disentangle. We use *t-SNE* [53] to visualize the representations of the last ResNet [22] layer output (before `fc` layer) in Figure 1. On a clean test set (Figure 1(a)), we can observe the change of data point distributions from ERM to advGDRO — over ERMs, the dataset representations have only one cluster; however, going into DROs, each group forms into more disentangled clusters and the disentanglement is most obvious on advGDRO. As indicated in [46], disentanglement aligns with the goal of *robustness*, i.e., our advGDRO moves toward learning meaningful representations that is robust. On perturbed test set (Figure 1(b)), though not as obvious as Figure 1(a), a similar trend can be observed – the data points become more sparsely scattered as models become more robust. We hypothesize it is because perturbations by nature add noises to images, resulting in more spurious correlations, and thus harder to disentangle; in other words, Figure 1(b) explains the performance drop and the limit of robust learning.

5 Conclusion

In this paper, we propose an algorithm for robust representation learning and explore the connections between *group-distributional robustness* and *adversarial robustness*. By achieving improved



(a) clean test set (b) perturbed test set

Figure 1: **t-SNE visualizations on Waterbirds.** Output of the last CNN layer (before fc). For (a) & (b), each row represents a different group; the columns from left to right are (1) ERM, (2) advERM, (3) GDRO, and (4) advGDRO. The majority color stands for the correction predictions, and the minority, wrong predictions. In (a), the data points tend to spread into distinctive clusters as the training method becomes more robust, and we believe that our algorithm may help the representations become more disentangled, bringing about better performance. While in (b), the trend is not as obvious, but still the data points become more spread out on the plane; we think such phenomenon is because perturbed data add more spurious correlations, and the robust training has its limit.

performances via our algorithm over benchmark datasets, we have made a step toward that goal, showing they can be complementary. Our results show that by utilizing group weighting end-to-end to learn the "uncertain distribution" we can further enhance the two types of robustness. On the representation side, when models are trained robustly, we observed that the representations learned show *disentanglement* on the 2D t-SNE embedding space, and therefore more robust and meaningful.

In sum, our work provides a connection for future studies in the robustness of distribution shifts and adversarial training, and on a broader level, the pursuit of learning robust representations.

References

[1] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang. Recent advances in adversarial training for adversarial robustness, 2021.

[2] A. Ben-Tal, D. den Hertog, A. D. Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013. ISSN 00251909, 15265501. URL `http://www.jstor.org/stable/23359484`.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

[4] D. Bertsimas and J. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific, 1997.

[5] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[6] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. J. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *ArXiv*, abs/1902.06705, 2019. URL `http://arxiv.org/abs/1902.06705`.

[7] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 4960–4964. IEEE, 2016. doi: 10.1109/ICASSP.2016.7472621. URL https://doi.org/10.1109/ICASSP.2016.7472621.

[8] J. Chen, X. Zhang, R. Zhang, C. Wang, and L. Liu. De-pois: An attack-agnostic defense against data poisoning attacks. In *IEEE Transactions on Information Forensics and Security*, 2021.

[9] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, 2018.

[10] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv*, abs/1712.05526, 2017. URL http://arxiv.org/abs/1712.05526.

[11] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[12] E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010. URL https://EconPapers.repec.org/RePEc:inm:oropre:v:58:y:2010:i:3:p:595-612.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[14] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 647–655, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/donahue14.html.

[15] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su. Adversarial distributional training for robust deep learning. In *Advances in Neural Information Processing Systems*, 2020.

[16] J. Duchi and H. Namkoong. Learning models with uniform performance via distributionally robust optimization. *ArXiv*, 2020.

[17] J. C. Duchi and H. Namkoong. Learning models with uniform performance via distributionally robust optimization. *ArXiv*, abs/1810.08750, 2018.

[18] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, volume abs/1412.6572, 2015.

[19] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *ArXiv*, abs/1708.06733, 2017. URL http://arxiv.org/abs/1708.06733.

[20] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *ICML*, 2018.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016.

[23] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[24] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvari. Learning with a strong adversary, 2016.

[25] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 2019.

[26] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*, volume abs/1707.07328, 2017. URL http://arxiv.org/abs/1707.07328.

[27] H. N. John Duchi, Tatsunori Hashimoto. Distributionally robust losses against mixture covariate shifts, 2019. URL https://web.stanford.edu/~hnamk/papers/DuchiHaNa19.pdf.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[29] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017. URL https://arxiv.org/abs/1607.02533.

[30] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2017.

[31] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao. Adversarial training for large neural language models. *ArXiv*, abs/2004.08994, 2020.

[32] Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. In *NDSS*, 2018.

[33] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. *ICCV*, pages 3730–3738, 2015.

[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2018.

[35] P. Maini, E. Wong, and J. Z. Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1909.04068.

[36] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019. doi: 10.1109/TPAMI.2018.2858821.

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

[38] H. Namkoong and J. C. Duchi. Variance-based regularization with convex objectives. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/5a142a55461d5fef016acfb927fee0bd-Paper.pdf.

[39] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *Society for Industrial and Applied Mathematics*, 19:1574–1609, 01 2009. doi: 10.1137/070704277.

[40] Y. Oren, S. Sagawa, T. B. Hashimoto, and P. Liang. Distributionally robust language modeling. In *EMNLP*, volume abs/1909.02060, 2019. URL http://arxiv.org/abs/1909.02060.

[41] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of tricks for adversarial training, 2021.

[42] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *ArXiv*, abs/2002.10716, 2020.

[43] J. Rauber and M. Bethge. Fast differentiable clipping-aware normalization and rescaling. *ArXiv*, abs/2007.07677, 2020. URL https://arxiv.org/abs/2007.07677.

[44] S. Sagawa, P. W. Koh, T. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *ArXiv*, abs/1911.08731, 2019.

[45] H. Salman, M. Sun, G. Yang, A. Kapoor, and J. Z. Kolter. Black-box smoothing: A provable defense for pretrained classifiers. In *NeurIPS*, volume abs/2003.01908, 2020. URL https://arxiv.org/abs/2003.01908.

[46] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Towards causal representation learning, 2021.

[47] S. Shafieezadeh-Abadeh, P. M. Esfahani, and D. Kuhn. Distributionally robust logistic regression. In *NIPS*, 2015.

[48] A. Sinha, H. Namkoong, A. Sinha, and J. Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[49] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. doi: 10.1109/TEVC.2019.2890858.

[50] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Jan. 2014. 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014.

[51] J. Szurley and J. Z. Kolter. Perceptual based adversarial audio attacks. *ArXiv*, 2019.

[52] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

[53] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[54] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[55] H. Wang, X. Wu, P. Yin, and E. Xing. High-frequency component helps explain the generalization of convolutional neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8681–8691, 2020.

[56] E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, 2018.

[57] C. Xiao, B. Li, J. yan Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3905–3911. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/543. URL https://doi.org/10.24963/ijcai.2018/543.

[58] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le. Adversarial examples improve image recognition. In *CVPR*, June 2020.

[59] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

[60] H. Zhang, H. Chen, C. Xiao, B. Li, D. S. Boning, and C. Hsieh. Towards stable and efficient training of verifiably robust neural networks. *ArXiv*, abs/1906.06316, 2019. URL http://arxiv.org/abs/1906.06316.

[61] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[62] S. Zhao, X. Ma, Y. Wang, J. Bailey, B. Li, and Y.-G. Jiang. What do deep nets learn? class-wise patterns revealed in the input space. *ArXiv*, abs/2101.06898, 2021.

[63] C. Zhou, X. Ma, P. Michel, and G. Neubig. Examining and combating spurious features under distribution shift. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Algorithm 1: Adversarial group DRO

Input: Step sizes: $\eta_q, \eta_\theta, \eta_\delta$; T: total number of iterations, ϵ: perturbation bound, Π: projection function, σ^2: variance of the noise initialization; K: the number of perturbation estimation steps, P_g for each $g \in G$

for $t = 1, ..., T$ **do**
- $g \sim$ Uniform(1,...,m) ; `// choose a group`
- $x, y \sim P_g$; `// sample batch`
- $\delta \sim \mathcal{N}(0, \sigma^2 I)$; `// sample noise`
- **for** $k = 1, ..., K$ **do**
  - $g_{adv} \leftarrow q_g^{(t-1)} \operatorname{sign}(\nabla_\delta \mathcal{L}(f(x+\delta), y; \theta^{(t-1)}))$; `// Get gradient direction`
  - $\delta \leftarrow \Pi_{\|\delta\| \le \epsilon}(\delta + \eta_\delta g_{adv})$; `// Ascent step and projection back to` L_p `ball`
- **end**
- $q' \leftarrow q^{(t-1)}; q'_g \leftarrow q'_g exp(\eta_q \mathcal{L}(f(x+\delta), y; \theta^{(t-1)}))$; `// update group weights`
- $q^{(t)} \leftarrow q' / \sum_{g'} q'_{g'}$; `// re-normalize`
- $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta q_g^{(t)} \nabla_\theta \mathcal{L}(f(x+\delta), y; \theta^{(t-1)})$; `// update model`

end

Output: model θ

Table 2: **Test results on CIFAR-10.** ($\epsilon = 8/255$)

Metric	Perturbation	CIFAR-10			
		ERM		GDRO	
		w/o AT	AT	w/o AT	AT
Average Acc.	Batch	92.1	89.7	89.7	89.6
	Group (ours)	-	-	89.6	89.6
Adversarial Acc.	Batch	22.3	79.2	31.5	77.2
	Group (ours)	-	-	**31.6**	**77.7**
Robust Acc.	Batch	66.1	73.8	78.5	79.3
	Group (ours)	-	-	**78.9**	**81.5**
Robust Adv. Acc.	Batch	2.5	51.4	19.4	59.8
	Group (ours)	-	-	14.8	**61.3**

A Datasets

Waterbirds & CelebA. Following [44], Waterbirds and CelebA both contain four groups (two classes $\times$ two spurious correlations), which are $Y \in$ {landbird, waterbird} and $attr \in$ {land, water} for Waterbirds, and $Y \in$ {female, male} and $attr \in$ {non-blonde, blond} for CelebA, with each group having an unbalanced number of examples. Table 3 in the Appendix presents detailed statistics and usages.

Table 3: We study datasets where a spuriously-correlated attribute is present and evaluate the effectiveness of our *adversarial group DRO* algorithm on average and on robust metrics.

Dataset	Split	Subgroup Size $(Y, attr)$			
Waterbirds		landbird, land	landbird, water	waterbird, land	waterbird, water
	train	3498	184	56	1057
	val	467	466	133	133
	test	2255	2255	642	642
CelebA		non-blonde, female	non-blonde, male	blonde, female	blonde, male
	train	71629	66874	22880	1387
	val	8535	8276	2874	182
	test	9767	7535	2480	180

(a) Training example 1 (y : waterbird; $attr$: water background).

(b) Training example 2 (y : landbird; $attr$: land background).

(c) Test example (y : waterbird; $attr$: land background).

Figure 2: **Example of spurious attributes.** Note that the correlations *water* or *land* between bird type y and background $attr$ (short for *attribute*) does not hold at test time.

CIFAR-10. Without manual spurious correlations to form groups, we treat each class of CIFAR-10 as a group in our experiments. Notice that the groups are different from class labels especially for Waterbirds and CelebA, and by nature the groups are different from CIFAR-10 classes since they contain manually crafted spurious features. We hold out 10% of the training set as validation data.

B Background

In this section, we set up basic notations again and then present the frameworks adopted in this work with brief discussions on their respective issues and connections.

We denote $\mathcal{D}$ as the dataset, and $\langle x, y \rangle$ as a data sample (the image and the corresponding label). $f(\cdot; \theta)$ denotes a deep neural network, which takes an $\langle x, y \rangle$ pair as input. θ is the set of parameters of the neural network. $\mathcal{L}(\cdot, \cdot)$ denotes a generic loss function (e.g., cross-entropy loss).

B.1 Empirical Risk Minimization

Typical machine learning algorithms adopt Empirical Risk Minimization (ERM) framework during training, where we learn a model parameterized by θ minimizing the empirical risk of $\mathcal{L}(\cdot, \cdot)$ under an empirical distribution $\hat{P}$ derived from training data $\mathcal{D}_{train}$:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\hat{P}} \mathcal{L}(f(x), y; \theta) \tag{4}$$

The underlying assumption is that the training and test set are sampled from the same distribution, i.e. *i.i.d.*, and thus we expect the model to generalize on the test set if it has been optimized during training. An issue with ERM is when the model encounters a data distribution that is different from $\hat{P}$ at test time, the performance drops rapidly [59]. The problem setup where the empirical training distribution $\hat{P}$ is different from test data sampled from some different distribution $\hat{P_T}$ is commonly called *distribution shift*.

B.2 Distributionally Robust Optimization (DRO)

To mitigate the issue arising from ERM, a natural solution is to use DRO [2], which instead minimizes the worst expected risk over a family of distributions $\mathcal{Q}$:

$$\min_{\theta\in\Theta} \left\{ R(\theta) = \sup_{Q\in\mathcal{Q}} \mathbb{E}_{(x,y)\sim Q} \left[\mathcal{L}(f(x), y; \theta)\right] \right\} \tag{5}$$

where $\mathcal{Q}$ is the uncertain set and $R(\theta)$ is the worst-distribution risk. Since $\mathcal{Q}$ encodes all possible distributions at test time, the model is expected to be robust to distributional shifts. A common choice for $\mathcal{Q}$ is a divergence ball around the training distribution which includes a wide range of distributional shifts.

However, [27] showed that having such a divergence ball could result in overly pessimistic models, and a more realistic setting called *group DRO* [23, 40, 44] is adopted in our work. Formally, we define

the $\mathcal{Q}$ in *group DRO* a coarse-grained mixture models where P is a mixture of m groups containing P_g where $g \in \mathcal{G} = \{1, ..., m\}$, and optimize Eq. (5) with $\mathcal{Q} = \{\sum_{g=1}^{m} q_g P_g : \sum_{g=1}^{m} q_g = 1, q_g \geq 0 \ \forall g \in \mathcal{G}\}$. This formulation allows us to learn models that are robust to *group shifts*. Equivalently, since the unique optimal solution of a linear program happens at a vertex [4], we can rewrite the inner optimization of Eq. (5) as

$$R(\theta) = \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim P_g}[\mathcal{L}(x, y; \theta)] \tag{6}$$

In practice, we leverage prior knowledge on specific tasks or data to define the groups and the corresponding uncertain distributions. For instance, based upon the bird categories and spurious background attribute, we have four groups for Waterbirds — {landbird, land; waterbird, land; landbird, water; waterbird, water}.

A nice application of *group DRO* is to avoid the reliance on spurious correlation [44, 63], and we hypothesize this can be improved by another robust training method, the adversarial training.

B.3 Adversarial Training

Different from *group-distributional robustness* in group DRO, AT aims at *adversarial robustness* against adversarial examples by finding the model that minimizes the loss of the maximally perturbed input so that $f(x + \delta) \neq f(x)$:

$$\hat{\theta}_{AT} = \operatorname*{argmin}_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \Delta} \mathcal{L}(f(x + \delta), y; \theta) \right], \tag{7}$$

where δ is the perturbation and Δ is the perturbation distribution. Δ is designed to be limited in a small boundary to be imperceptible to human eyes [18, 30]. For example, given a small budget ϵ, $\Delta := \{\delta : \|\delta\|_p \leq \epsilon\}$ where $\|\cdot\|_p$ is the L_p norm. In our work we conduct a number of projected gradient steps to solve for the inner maximization [34, 31]

$$\begin{aligned} g_{adv} &\leftarrow \text{sign}(\nabla_{\delta^{(t)}} \mathcal{L}(f(x + \delta^{(t)}), y; \theta)) \\ \delta^{(t+1)} &\leftarrow \Pi_{\|\delta^{(t)}\| \leq \epsilon}(\delta^{(t)} + \eta_\delta g_{adv}) \end{aligned} \tag{8}$$

where $\Pi(\cdot)$ is the projection function, η_δ is the adversarial step size, and initial perturbation $\delta^{(0)}$ is sampled from a normal distribution, $\mathcal{N}(0, \sigma^2 I)$. We refer readers to [1] for a recent survey on AT.

C Implementation Details

For a single experiment, we use two NVIDIA Tesla P100 GPUs. All our CNN models use SGD as optimizers. Due to resource limits, on CIFAR-10, we train ResNet-110 from scratch with a batch size of 128 and learning rate $\eta_\theta = 0.1$. On CelebA and Waterbirds, we use pre-trained ResNet-50 with a batch size of 110 and $\eta_\theta = 0.001$. To train robust models, we perturb input images with max perturbation boundary $\epsilon = 2/255$ on a L_∞ ball, initial Gaussian noise $\delta^{(0)}$ with $\sigma = \epsilon^2$, and a step size $\eta_\delta = 0.01$ for 5 steps. Additionally on CIFAR-10, we test Algorithm 1 with $\epsilon = 8/255$. On DROs we set group update rates $\eta_q = 0.01$. We did not fine-tune hyperparameters extensively and only set them to standard values used in previous work, and we believe that fine-tuning can further improve the results.

Model selection. All models are evaluated at the best early stopping epoch as measured by robust metrics on validation set. This way we make sure our results are not overfitting towards the robust metrics.

D Related work

Representations of neural networks. The success of a deep learning model generally depends its ability of learning more complex and high quality representations than traditional models [3]. [14] showed that intermediate layers of CNN tend to learn simple patterns and high level shapes like lines and corners [28]. These features are essential to the performance of CNNs [21]. Our work attempts to learn robust representations under an adversarial and uncertain setting.

Attacking DNNs. An active area of studying DNN behaviors is attacking the DNNs. Researchers have found that DNNs are susceptible to various types of *threat models*, which deceive the models and cause undesired behaviors of DNNs. One type is the backdoor attack — implanting malicious data into the training data [19, 32], the model learns incorrect behaviors and then consistently makes the wrong decisions at test time. Another situation is the adversarial attack — ever since [18], adversarial examples have been broadly studied and a wide range of attacks such as FGSMs and PGD [18, 30, 29, 34] have been proposed, and they give rise to a broader discussions of vulnerabilities of neural networks in computer vision [43, 49, 10, 57, 9], speech processing [51] and NLP tasks [26]. Meanwhile, understanding adversarial examples [25, 62] has been studied as well.

Toward robustness. In order for the models to defend against the deceptions or *threats*, researchers have set to work on closing the gap between adversarial accuracy and standard accuracy [34, 5], and a wide range of *defense methods* for different types of attack [8, 45, 56]. Adversarial training, discussed in Section B.3, is the most popular method against adversarial attacks. Previous works demonstrated its capability of working with other frameworks [42, 58, 36, 15], such as self-supervised learning, etc. Discussions about the trade-off between the robustness and generalization have been attempted [55, 61]. However, a universal method to fully prevent all the aforementioned attacks from occuring has not yet been developed [6].

Robust optimization. The community has also started to study robustness from an optimization point of view by proving certifiable bounds of the attacks [60, 11]. DRO has drawn attention, due to its nature to upper-bound the expected risk under an unknown test distribution [16, 2], and how the distributions are formed — either a coarse-grained group [23, 40, 44] as we adopted in this work, or other types [2, 48, 12]. Solving DRO problems under different setups [47, 37, 38] has also been proposed and studied.

E Computational efficiency

AT is previously known to require a longer training time till completion given a total number of epochs; we empirically find that in our setup the run time of our algorithm is only less than 1% slower than ERM and differs with AT by less than or around 5%, showing the efficiency of our approach.

F Corrections from models to models show less reliance on manual spurious correlation.

The worst-performing group that the models end up having *a posteriori* also give us some signals for the effect of a spurious attribute; for example, on Waterbirds, the most common worst groups are {waterbird, land; landbird, water}, which means the spurious attribute “background" is a factor that affects the model. Should our method mitigate this effect, we can correct mistakes made by a less robust model (e.g. ERM). Therefore, we plot out samples that are mistakenly predicted by a less robust model and yet corrected by a robust model. Figure 3 shows such samples on the Waterbirds. For example, in row 2, the group was {landbird, water} and advERM predicted them as Waterbirds but advGDRO can successfully make the right prediction; in row 1 & 3, advGDRO can make correct predictions on group {waterbird, land}. In other words, advGDRO is the most robust against spurious attributes and can prevent learning them.

G Robustness and filter smoothness

We plot the kernels of the first convolutional layer of CNN on CIFAR-10 and draw connection with [55] to further help to see what representations the model learns. [55] proposes that the filters should be smoother when the model is adversarially trained; and when kernels are regularized to be smoother, the model are stronger against FSGM and PGD. In Figure 4, we see that the kernels from advGDRO (Figure 4(d)) are smoother than others, meaning DRO indeed helps adversarial robustness. Notably, the fact that the kernels become smoother from Figure 4(b) to (d) is also congruent with our observations that DRO help getting the model more robust against adversarial attacks.

Figure 3: **advGDRO can correct mis-predictions from all other models.** Row 1 shows the image predictions corrected from ERM to advGDRO; row 2 shows the images corrected from advERM to advGDRO; row 3 shows the images corrected from GDRO to advGDRO. Title of each image is the *prediction of robust model* (✓)*/prediction of comparing model* (✗).

H Convergence of Algorithm 1

We study error ϵ_T of the average iterate $\bar{\theta}^{(1:T)}$ and then analyze the convergence rate:

$$\epsilon_T = \max_{q \in \mathcal{Q}} L(\bar{\theta}^{(1:T)}, q) - \min_{\theta \in \Theta} \max_{q \in \mathcal{Q}} L(\theta, q), \tag{9}$$

where $L(\theta, q) := \sum_{g=1}^{m} q_g \, \mathbb{E}_{(x,y)\sim P_g}[\max_{\delta \in \Delta} \mathcal{L}(f(x+\delta), y; \theta)]$ is the expected worst-case adversarial loss. Applying Danskin's theorem and results from [39, 44], we show in Proposition 1 that Algorithm 1 has a standard convergence rate of $O(1/\sqrt{T})$ in a convex setting.

Proposition 1. *Suppose that the loss $\mathcal{L}(\cdot; (x, y))$ is non-negative, convex, B_∇-Lipschitz continuous, and bounded by $B_\mathcal{L}$ for all (x, y) in $\mathcal{X} \times \mathcal{Y}$, and $\|\theta\|_2 \leq B_\Theta$ for all $\theta \in \Theta$ with convex $\Theta \subseteq R^d$. Then, the average iterate of Algorithm 1 achieves an expected error at the rate*

$$\mathbb{E}[\epsilon_T] \leq 2m \sqrt{\frac{10[B_\Theta^2 B_\nabla^2 + B_\mathcal{L}^2 \log m]}{T}}. \tag{10}$$

Proof. We prove Proposition 1 in two parts. First we prove that the inner-most maximization of (3) is convex and differentiable, and then by Proposition 2 of [44], we get the convergence guarantee.

Let $F(\theta) := \max_{\delta \in \Delta} \mathcal{L}(f(x+\delta), y, \theta)$. Since Δ is a compact convex set, by Danskin's theorem, if $Z_0(\theta) := \{\delta \in \operatorname{argmax}_{\delta \in \Delta} \mathcal{L}(f(x+\delta), y, \theta)\}$ is singleton for some θ, then $F(\theta)$ is convex and directionally differentiable. By Corollary C.2 of [34], $F(\theta)$ has an ascent direction, and in practice, we observe most of the elements of δ reach the boundary after the projected gradient steps.

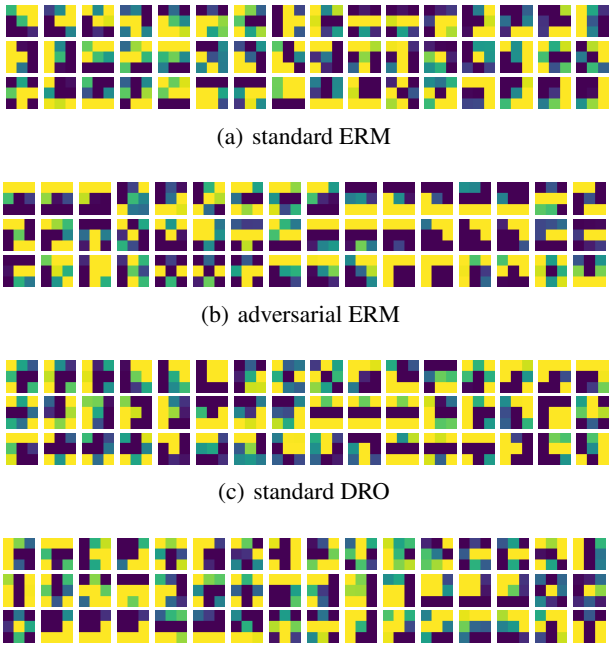
Now (3) can be written as a saddle-point problem,

$$\min_{\theta \in \Theta} \max_{q \in \mathcal{Q}} \sum_{g=1}^{m} q_g \mathbb{E}_{(x,y)\sim \hat{P}_g} \, [F(\theta)]. \tag{11}$$

By Proposition 2 of [44], we can use the result of [39] Eq.(3.23) to obtain a similar bound,

$$\mathbb{E}[\epsilon_T] \leq 2m \sqrt{\frac{10[B_\Theta^2 B_\nabla^2 + B_\mathcal{L}^2 \log m]}{T}}. \tag{12}$$

□



(a) standard ERM

(b) adversarial ERM

(c) standard DRO

(d) adversarial DRO

Figure 4: **Visualization of CNN kernels** (16 kernels each channel $\times$ 3 channels at the first layer). According to [55], adversarially robust model should have smoother kernels, and our method (*advG-DRO*) produces similar outcome.