Better than Balancing: Debiasing through Data Attribution

Saachi Jain[*] **Kimia Hamidieh**[*] **Kristian Georgiev**[*]
Marzyeh Ghassemi, Aleksander Mądry
MIT
{saachij,hamidieh,krisgrg,mghassem,madry}@mit.edu

Abstract

Spurious correlations in the training data can cause serious problems for machine learning deployment. However, common debiasing approaches which intervene on the training procedure (e.g., by adjusting the loss) can be especially sensitive to regularization and hyperparameter selection. In this paper, we advocate for a *data-based perspective* on model debiasing by directly targeting the root causes of the bias within the training data itself. Specifically, we leverage data attribution techniques to isolate specific examples that disproportionally drive reliance on the spurious correlation. We find that removing these training examples can efficiently debias the final classifier. Moreover, our method requires no additional hyperparameters, and does not require group annotations for the training data.

1 Introduction

The composition of the training dataset has a substantial impact on the features that a model learns, and thus its reliability during deployment [5, 11]. In particular, the training dataset might contain spurious correlations—features which are statistically associated but causally irrelevant for the final target. Such reliance on spurious correlations can lead to poor generalization, especially on underrepresented subpopulations that do not share the same spurious patterns [31, 6, 3].

Many current debiasing strategies [26, 13] intervene after training (e.g., by fine-tuning on a smaller, balanced held-out set [13]). These post-hoc approaches assume that a heavily biased model retains strong enough features that can be retrofitted to create an unbiased classifier. Other approaches try to adjust the loss during training [24, 4], or utilize sample re-weighting [16, 19] to improve the worst-group performance. However, such approaches tend rely heavily on carefully tuned hyperparameters and, in many settings, do not even outperform basic empirical risk minimization (ERM) [7].

Fundamentally, these approaches do not address the root cause of the bias: the training data itself. Indeed, recent work has shown that simple balancing can perform on par with more complex approaches [10]. We thus hypothesize that taking a *data-based approach*, by eliminating biases within the training data, can be a more effective strategy for debiasing. Unfortunately, data balancing requires group labels of the training data (which might not be available). Moreover, for highly skewed datasets, balancing the data can require either removing large parts of the dataset or oversampling a very small number of examples, preventing the model from learning useful features [2, 27].

Balancing assumes that all majority examples equally contribute to the underlying bias. But is this really the case? In particular, if we can identify which examples from the dataset are driving the model's reliance on the spurious correlation, we can more efficiently debias the model without removing a large fraction of the dataset. This motivates the question:

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

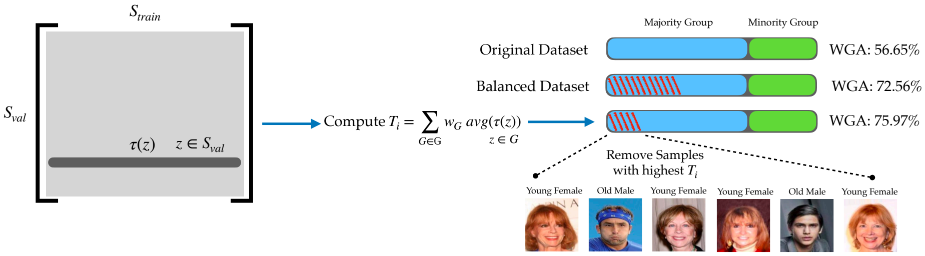

Figure 1: We use TRAK [22] to identify examples that most drive a model's bias. By removing those examples, we can more efficiently improve worst-group accuracy over approaches such as balancing.

How can we pinpoint the training examples that disproportionately drive model biases?

Our Contributions In this work, we investigate using data attribution to measure the impact of individual training datapoints on a model's biases. In particular, we leverage TRAK [22], a framework for approximating the counterfactual impact of training data on model predictions, to identify points that encourage a model to rely on a spurious correlation. Evaluating our framework on a variety of tasks, we demonstrate that our approach can:

- **Pinpoint influential examples for a specified model bias.** We find that typically only a small fraction of training examples drive reliance on spurious correlations.
- **Efficiently debias models by removing a small number of influential training points.** Our approach outperforms balancing and other common debiasing approaches [24, 16, 13].
- **Discover biases by examining the data attribution matrix.** We demonstrate that biases can often be extracted directly from the attribution matrix. By leveraging this observation, we can automatically discover (and then intervene on) hard subpopulations within the data.

Finally, we leverage our framework to discover and mitigate biases within the ImageNet dataset. Our method surfaces coherent color and co-occurrence biases. We then debias the model according to these failures, and improve accuracy on the identified populations.

Our approach does not require subpopulation annotations for the training dataset, and only (optionally) leverages subpopulation labels for a small, held out validation set. By targeting biases within dataset itself, our work takes a first step toward a *data-based perspective* on debiasing.

2 Debiasing datasets with data attribution

Spurious correlations can encourage models to rely on features that do not reliably generalize during deployment. As a running example, consider classifying "young" versus "old" faces from the CelebA dataset [18], where age and gender are correlated during training (i.e., the faces of young women and old men are overrepresented). A model trained on such a skewed dataset might learn to conflate gender with age, and thus struggle to correctly predict the age of old women and young men.

How can we train a classifier which can accurately classify the age of a face regardless of the gender? In this paper, we target such biases at their source: by finding (and removing) training examples that drive the targeted bias. To do so, we leverage *data attribution* techniques [11, 22, 14], which enable us to quantify the impact of a particular training point on the model's predictions.

While any data attribution technique could fit in our framework, here we use TRAK [22], which forms a linear approximation of the neural network in terms of the model's gradients and then estimates the leave-one-out influence of each example. In this section, we'll briefly describe our problem setup, and then explain such data attribution methods can be used to debias models.

2.1 Background and setup

We consider a setting where each example $z = (x, y)$ belongs to a pre-defined group $G \in \mathcal{G}$. For instance, in our running example, $\mathcal{G}$ would be faces of old women, young women, old men, and young men. We are given a training dataset $S_{\text{train}} = \{(x_1, y_1), ..., (x_n, y_n)\}$ which does *not* contain any group annotations. Additionally, at least initially, we also assume access to a validation set S_{val}

where we *do* know which group $G \in \mathcal{G}$ each validation example $z \in S_{\text{val}}$ belongs to (i.e., $z \in G$). In Section 3, we will consider the setting where these validation group labels are not available.

Our goal then is to train a classifier to maximize the *worst group accuracy*: i.e., the minimum accuracy on the test examples that belong to each of the groups in $\mathcal{G}$. Finally, we leverage TRAK, a scalable data attribution method. We provide a brief overview of TRAK in Figure B.

2.2 Measuring group alignment

A natural way to use data attribution to address model biases is to eliminate examples that negatively influence that negatively influence the worst-performing groups. However, not all "minority groups" may be equally hard for the model. Returning to our running example, while both old women and young men do not align with the age/gender correlation, the model may still perform better on old women than young men (e.g., due to data availability).

Thus, for each group, we first compute a weight w_G that captures how badly the model is underperforming on that group. Specifically, for a group $G \in \mathcal{G}$, let $S_{\text{val}}^G = S_{\text{val}} \cap G$ be the validation examples in that group. Following Sagawa et al. [24], we assign a group's weight according to its average loss[2]: $w_G = \exp\left(\frac{1}{|S_{\text{val}}^G|} \sum_{z \in S_{\text{val}}^G} \log p(z)\right)$ where $p(z)$ is the probability assigned by the model to example z for the correct class. Then, for each training example z_i, we compute the *group alignment* score T_i by examining the (weighted) aggregated influence of z_i on each of the groups:

$$T_i = \sum_{G \in \mathcal{G}} \frac{w_G}{|S_{\text{val}}^G|} \sum_{z \in S_{\text{val}}^G} \tau(z)_i.$$

By weighting the groups according to their loss, we prioritize removing negative influencers for particularly underperforming groups.

Debiasing the dataset Then, to remove bias from the training set, we can simply remove the top K examples with the most negative T_i as shown in Figure 1. One way to choose the number of examples K is to search for the best K that maximizes worst group accuracy on the validation set. However, as a heuristic, we instead choose K to remove all examples with a negative group alignment score $T_i < 0$. As will show in Figure 2, this heuristic tends to slightly over-estimate the best K.

2.3 Spurious attribute discovery

To compute τ, we relied on validation group labels which defined the targeted "bias." However, in many real-world settings, we might not even know which biases are impacting our model. How can we use data attribution to discover (and then intervene on) biases within the training data?

We make a key observation: strong biases often cause noticeable variations in the patterns of attribution scores. Let's return to the case of predicting age in the presence of a gender skew. Here, the gender of a validation example has a large impact on the set of training examples it relies on. Thus, we can isolate intra-class variations in the attribution matrix to automatically discover biases.

Specifically, for each class c, we stack the attribution scores $\tau(z)$ for all validation examples of that class. We then use the top principal component of the resulting matrix to cluster the validation examples into different "groups." Indeed in Appendix Figure 5, we find that performing this clustering on the young vs. old example does identify the underlying bias of gender. We can then use these pseudo-annotations for the group labels when computing the group alignment scores.

3 Results

In Section 2, we defined the group alignment score T_i, which captures a training example's contribution to the targeted bias. In this section, we use these alignment scores to efficiently debias the underlying model. We consider three different datasets that contain a planted spurious correlation: `CelebA-Age` [18, 12], `CelebA-Blond` [18], `Waterbirds` [25], and `MultiNLI` [30]. For each dataset, we compute TRAK and calculate the group alignment scores T_i. We then retrain the model after removing training examples with negative T_i (See Appendix D.2 for experimental details).

[2] Unlike GroupDRO [24], we compute w_G only once with the model's converged parameters

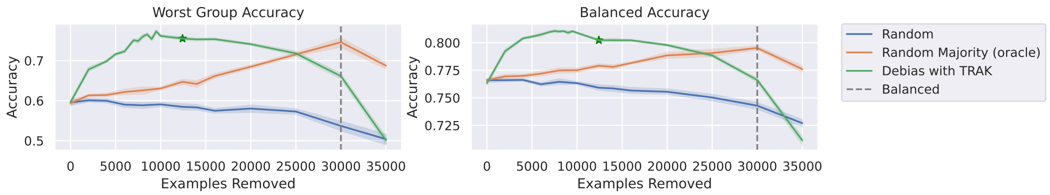

Figure 2: Worst group and balanced accuracies on `CelebA-Age` after using TRAK to remove K training examples (most negative T_i first). The green star marks the K used by our heuristic ($T_i < 0$). Debiasing with TRAK efficiently improves worst group accuracy.

Group Info Train / Val	**Method**	**Worst Group Accuracy** `CelebA-Age`	 `CelebA-Blond`	 `Waterbirds`	 `MultiNLI`
✗/ ✗	ERM	56.65	45.86	57.85	67.19
	Auto-TRAK (ours)	**75.97**	83.77	81.04	74.7
✗/ ✓	JTT [16]	60.95	81.61	63.61	72.6
	DFR* [13]	70.37	88.40	**88.96**	74.7
	TRAK (ours)	**75.55**	**90.03**	87.15	81.54
✓/ ✓	RWG [10]	**75.64**	88.40	81.21	68.41
	SUBG [10]	68.49	88.26	85.46	67.76
	GroupDRO [24]	74.80	**90.61**	72.47	77.7

Table 1: Balanced accuracy and worst-group accuracy on `CelebA-Age` , `CelebA-Blond` , and `Waterbirds` . A $*$ indicates that the method uses validation group labels for model finetuning, in addition to hyperparameter tuning.

Identifying the drivers of model biases How well does T_i isolate the "drivers" of the model's bias? To answer this question, we iteratively remove training examples from `CelebA-Age` starting with the most negative T_i and measure the worst-group and balanced accuracy (See Figure 2). `CelebA-Age` has 40K "majority" examples and 10K "minority" examples; thus, naive balancing requires removing 30K training examples. In contrast, by isolating *which* specific majority examples contribute to the bias, our method is able to debias the classifier by removing only 10K examples

We also note that our heuristic of removing examples with negative T_i (the green star in Figure 2) slightly over-estimates the best number of examples to remove. Thus, while this heuristic gives a decent starting point for K, actually searching for the best K might further improve performance.

Debiasing the model in the presence of validation group labels We use TRAK to debias the classifier for each dataset, leveraging the provided validation group labels to compute the group alignment scores. In Table 2, we compare against several baselines, each of which requires either only validation group labels (✗/ ✓) or both training and validation group labels (✓/ ✓). Further information about each of these baselines can be found in Appendix D.1. We find that debiasing with TRAK improves worst-group accuracy over all other baselines on both CelebA datasets. On Waterbirds, our method out-performs all methods except DFR.[3]

Discovering biases through the TRAK matrix We now consider the case where validation group labels are not accessible. To address this setting, we create pseudo-annotations for the validation set by dividing each class into two groups based on the top principal component of the TRAK matrix. These pseudo-annotations are then used when computing the group alignment scores (Auto-TRAK in Figure 2)[4]. Note that Auto-TRAK is the only method that does not require either train or validation group labels. Despite this, Auto-TRAK achieves competitive worst-group accuracy in our experiments.

[3] For WaterBirds, there are more examples for the smallest group in the val split (133) than the training split (56). Since DFR directly fine-tunes on the validation set, it has a distinct advantage here over all other methods.

[4] For `MultiNLI` , we chose the PCA component by inspection that captures examples with/without negation.

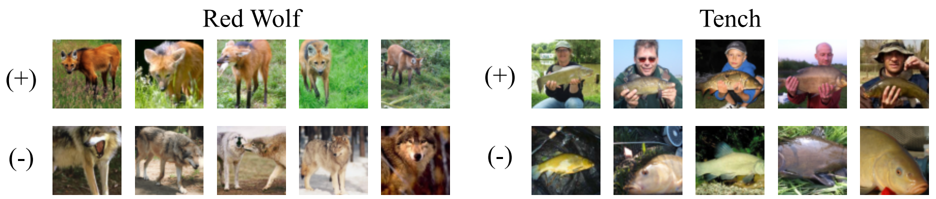

Figure 3: For four ImageNet classes, the most positive or negative examples according to the top PCA direction of the TRAK matrix. Our method identifies coherent color and co-occurrence biases.

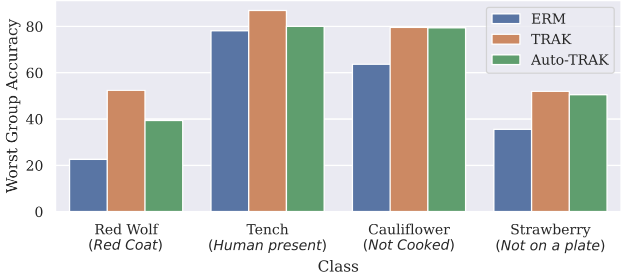

Figure 4: Worst Group Accuracy for ImageNet classes after intervening with either TRAK/Auto-Trak.

4 Case Study: Finding and Mitigating ImageNet Failures

In this section, we consider deploying our method to both discover and mitigate biases within ImageNet-trained models. Specifically, we first identify potential biases for specific ImageNet classes by examining the first principal component in the TRAK matrix. We then remedy the identified failure by using Auto-TRAK to remove the training examples that drive that bias.

Identifying ImageNet Biases We compute the TRAK matrix over the training dataset for a held out validation split (10% of the training set). Focusing on four ImageNet classes (as in Jain et al. [12]), we then use the first principal component of the TRAK matrix to identify potential biases. In Figure 3, we display the most extreme training examples according to the top principal component for each class. Our method identifies semantically color and co-occurrence biases (e.g., tench fishes with our without humans or yellow/white cauliflowers that are either cooked or uncooked.)[5]

Mitigating ImageNet Biases with Auto-TRAK For each of the four targeted ImageNet classes, we seek to mitigate the identified failure modes with TRAK. In order to evaluate the efficacy of our approach, we hand-label the 50 test images for each targeted class according to a human description of identified bias. In Figure 4, we display the worst group accuracy on the test images of the targeted class (with groups hand-labeled according to a human description of the identified bias) after using TRAK (with hand-labeled validation groups) or Auto-TRAK (deriving group validation labels from the top principal components) to remove examples. [6] Both TRAK and Auto-TRAK are able to improve worst group accuracy over the ERM model without significantly impacting the overall ImageNet accuracy (see Appendix Table 3).

5 Conclusion

In this work, we propose a simple method for debiasing models by isolating training examples which disproportionately contribute to the model's predictions on underperforming groups. Our method does not require training group labels, and does not rely on carefully-tuned hyperparameters. By targeting biases at their source, our work takes a first step toward a data-centric approach on debiasing.

[5] Our identified biases match the challenging subpopulations in Jain et al. [12].

[6] Here, we only consider the target class when computing the loss weighting. As a result, the heuristic overestimates the number of examples to remove. Thus, we instead search for the best number to remove using our held out validation split.

References

[1] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. “Learning imbalanced datasets with label-distribution-aware margin loss”. In: *Advances in neural information processing systems* 32 (2019).

[2] Rhys Compton, Lily Zhang, Aahlad Puli, and Rajesh Ranganath. “When More is Less: Incorporating Additional Datasets Can Hurt Performance By Introducing Spurious Correlations”. In: *arXiv preprint arXiv:2308.04431* (2023).

[3] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. “AI for radiographic COVID-19 detection selects shortcuts over signal”. In: *Nature Machine Intelligence* 3.7 (2021), pp. 610–619.

[4] John Duchi and Hongseok Namkoong. “Learning models with uniform performance via distributionally robust optimization”. In: *arXiv preprint arXiv:1810.08750*. 2018.

[5] Vitaly Feldman. “Does Learning Require Memorization? A Short Tale about a Long Tail”. In: *Symposium on Theory of Computing (STOC)*. 2019.

[6] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence*. 2020.

[7] Ishaan Gulrajani and David Lopez-Paz. “In search of lost domain generalization”. In: *arXiv preprint arXiv:2007.01434* (2020).

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. 2015.

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

[10] 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*. PMLR. 2022, pp. 336–351.

[11] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. “Datamodels: Predicting Predictions from Training Data”. In: *International Conference on Machine Learning (ICML)*. 2022.

[12] Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. “Distilling Model Failures as Directions in Latent Space”. In: *arXiv preprint arXiv:2206.14754* (2022).

[13] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. “Last layer re-training is sufficient for robustness to spurious correlations”. In: *arXiv preprint arXiv:2204.02937* (2022).

[14] Pang Wei Koh and Percy Liang. “Understanding Black-box Predictions via Influence Functions”. In: *International Conference on Machine Learning*. 2017.

[15] Yoonho Lee, Huaxiu Yao, and Chelsea Finn. “Diversify and disambiguate: Learning from underspecified data”. In: *arXiv preprint arXiv:2202.03418* (2022).

[16] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. “Just Train Twice: Improving Group Robustness without Training Group Information”. In: *International Conference on Machine Learning*. 2021.

[17] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. “Just Train Twice: Improving Group Robustness without Training Group Information”. In: *International Conference on Machine Learning (ICML)*. 2021.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep Learning Face Attributes in the Wild”. In: *International Conference on Computer Vision (ICCV)*. 2015.

[19] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. “Learning from Failure: Training Debiased Classifier from Biased Classifier”. In: *Neural Information Processing Systems (NeurIPS)*. 2020.

[20] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. “Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation”. In: *arXiv preprint arXiv:2204.02070* (2022).

[21] Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. “Agree to disagree: Diversity through disagreement for better transferability”. In: *arXiv preprint arXiv:2202.04414* (2022).

[22] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. "TRAK: Attributing Model Behavior at Scale". In: *Arxiv preprint arXiv:2303.14186*. 2023.

[23] Daryl Pregibon. "Logistic Regression Diagnostics". In: *The Annals of Statistics*. 1981.

[24] 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*. 2020.

[25] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. "An investigation of why overparameterization exacerbates spurious correlations". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8346–8356.

[26] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. "Editing a classifier by rewriting its prediction rules". In: *Preprint*. 2021.

[27] Roy Schwartz and Gabriel Stanovsky. "On the limitations of dataset balancing: The lost battle against spurious correlations". In: *arXiv preprint arXiv:2204.12708* (2022).

[28] Nimit S Sohoni, Maziar Sanjabi, Nicolas Ballas, Aditya Grover, Shaoliang Nie, Hamed Firooz, and Christopher Ré. "Barack: Partially supervised group robustness with guarantees". In: *arXiv preprint arXiv:2201.00072* (2021).

[29] Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton Van den Hengel. "Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16761–16772.

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

[31] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. "Noise or signal: The role of image backgrounds in object recognition". In: *arXiv preprint arXiv:2006.09994* (2020).

[32] Jingzhao Zhang, Aditya Menon, Andreas Veit, Srinadh Bhojanapalli, Sanjiv Kumar, and Suvrit Sra. "Coping with Label Shift via Distributionally Robust Optimisation". In: *arXiv preprint arXiv:2010.12230* (2020).

[33] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. "Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations". In: *arXiv preprint arXiv:2203.01517* (2022).

A Appendix

B Background on Data Attribution and TRAK

Let $\mathcal{Z}$ be the input space and $\mathcal{S}$ be a training set of interest. For a given training subset $S' \subset S$ and target example z, data attribution techniques seek to estimate the attribution score of z — that is, the change in the model's prediction on z when the model is trained on the subset S'. More formally, let $f(z, \theta(S))$ be the model's output function on example z. Then we can define $\tau(z)_i$ as the attribution score of the ith training example z_i on target example z

$$\tau(z)_i = f(z, \theta(S)) - f(z, \theta(S \backslash z_i)).$$

While $\tau(z)$ is relatively straightforward to compute for linear models [23], computing this influence is far more challenging for neural networks. Thus, in order to approach this problem, TRAK first approximates $f(z; \theta(S))$ as a linear model on top of the gradients $\nabla_\theta(f; \theta^*)$ of the original neural network after convergence. We can then plug this approximation into the estimate for linear classifiers to approximate $\tau(z)$. After simplification, the TRAK estimate of the influence of z is

$$\tau(z) = -\phi(z)^T(\Phi^T\Phi)^{-1}\Phi\mathbf{Q}$$

where $\phi(z) = \nabla f(z; \theta(S^*))$ are the (randomly projected) gradients on example z, Φ are the stacked training gradients $\Phi = [\phi(z_1), ..., \phi(z_n)]$, and $\mathbf{Q}$ is a normalization matrix.

C Additional Related Work

A variety of approaches have been proposed to mitigate learning spurious correlations or shortcuts from training data, and learn classifiers which optimize the model to be robust to group shifts. Many approaches leverage group information during training to combat spurious correlations or improve robustness to shifts in group proportions between train and test distributions. For example, some methods minimize the worst-group loss during training [24, 32, 9], reweight or subsample to balance majority and minority groups [10], use a balanced validation set to re-train the last layer [13], or impose regularization around minority points [1].

In the absence of group labels during training, several works aim to identify the minority group examples [17, 33], learn multiple diverse classifiers [15, 21, 29], or use partially available group labels [28, 20]. However, all approaches require group information for model selection. In our approach Auto-TRAK, we do not have access to group annotations for training or hyperparameter selection. In TRAK, we assume having access to a small validation set annotated with group labels.

D Details of Experiments

D.1 Experimental Setup

In this section, we describe the datasets, models and evaluation procedure that we use throughout the paper.

Datasets. In order to cover a broad range of practical scenarios, we consider the following image classification and text classification problems.

- `Waterbirds` [25] is a binary image classification problem, where the class corresponds to the type of the bird (landbird or waterbird), and the background is spuriously correlated with the class. Namely, most landbirds are shown on land, and most waterbirds are shown over water.
- `CelebA-Blond` [18] is a binary image classification problem, where the goal is to predict whether a person shown in the image is blond; the gender of the person serves as a spurious feature, as 94% of the images with the "blond" label depict females.
- `CelebA-Age` [18, 12] is a binary image classification problem, where the goal is to predict whether a person shown in the image is young; the gender of the person serves as a spurious feature. For this task, we specifically subsample the training set such that the ratio of samples in the majority vs. minority groups is 4:1.

Methods. We benchmark our approach against the following methods:

- **ERM** is simple empirical risk minimization on the full training set.
- **RWG** [10] is ERM applied to random batches of the data where the groups are equally represented with a combination of upsamping and downsampling such that the size of the dataset does not change.
- **SUBG** [10] is ERM applied to a random subset of the data where we subsample all groups such that they have the same number of examples.
- **GroupDRO** [24] trains that minimizes the worst-case performance over pre-defined groups in the test dataset.
- **Just Train Twice (JTT)** [17] trains an ERM model with upsamping initially misclassified training examples by an initial ERM model.
- **DFR** [13] trains an ensemple of linear models on a balanced validation set, given ERM features.

D.2 Training Details

In this section, we detail the model architectures and hyperparameters used by each approach. We used the same model architecture across all approaches: Randomly initialized ResNet-18 [8] for CelebA and ImageNet-pretrained ResNet-18s for Waterbirds. We use the GroupDRO implementation by Sagawa et al. [24] and DFR implementation by Kirichenko et al. [13].

For all approaches, we tune hyperparameters for ERM-based methods (ERM, DFR, and FAIR-TRAK) and re-weighting based methods (RWG, SUBG, GroupDRO and JTT) separately. For RWG, SUBG, GroupDRO and JTT, we early stop based on highest worst-group accuracy on the validation set as well. We optimize all approaches with Adam optimizer.

For the CelebA dataset, we all methods with learning rate $1e-3$, weight decay $1e-4$, and batch size 512. We train RWG, SUBG, GroupDRO and JTT with learning rate $1e-3$, weight decay $1e-4$, and batch size 512. We train all models for the `CelebA-Age` task to up to 5 epochs and all models for `CelebA-Blond` task up to 10 epochs.

For the Waterbirds dataset, we train the approaches that use the ERM objective (including FAIR-TRAK) with learning rate $1e-4$, weight decay $1e-4$, and batch size 32. We train RWG, SUBG, GroupDRO and JTT with learning rate $1e-5$, weight decay 0.1, and batch size 32. We train all models to up to 20 epochs.

For all other hyperparameters, we use the same hyperparameters as Kirichenko et al. [13] for DFR and the same hyperparameters as Liu et al. [16] for JTT.

We report the performance of the models via Worst-group Accuracy, or Balanced Accuracy in Table 2, which is the average of accuracies of all groups. If all groups in the test set have the same number of examples, balanced accuracy will be equivalent to average accuracy.

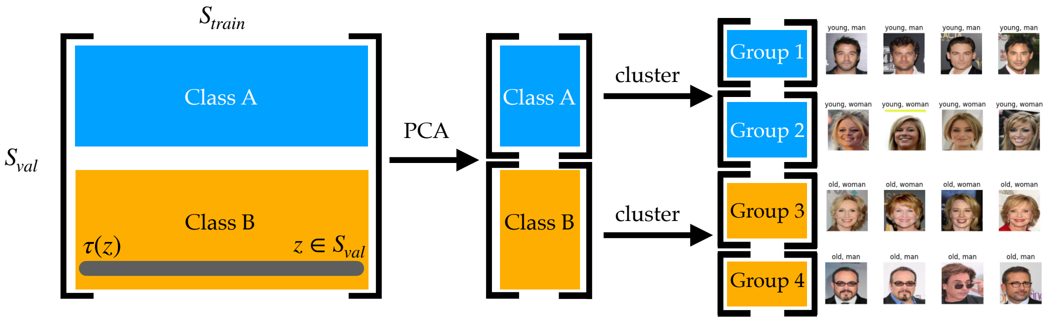

Figure 5: Procedure for discovering spurious attributes.

D.3 Details on Auto-Trak

To discover spurious attributes, we first compute the TRAK matrix for the validation set. We then compute the top principal components of the TRAK matrix and cluster the validation examples based on them. Finally, we use the clusters to create pseudo-annotations for the validation set.

E Omitted Results

E.1 Balanced Accuracies

Below we include the balanced accuracies for the experiments in Table 2.

Method	Group Info	CelebA-Age		CelebA-Blond		Waterbirds		MultiNLI	
	Train / Val	Balanced Accuracy	Worst Group Accuracy	Balanced Accuracy	Worst Group Accuracy	Balanced Accuracy	Worst Group Accuracy	Balanced Accuracy	Worst Group Accuracy
ERM	✗/✗	77.96	56.65	82.59	45.86	83.40	57.85	80.92	67.19
Auto-TRAK (ours)	✗/✗	80.05	**75.97**	91.01	83.77	90.36	81.04		
RWG [10]	✓/✓	80.66	**75.64**	90.42	88.40	86.51	81.21	78.61	68.41
SUBG [10]	✓/✓	77.57	68.49	91.30	88.26	86.97	85.46	73.64	67.76
GroupDRO [24]	✓/✓	80.88	74.80	91.83	**90.61**	86.51	72.47	81.4	77.7
JTT [16]	✗/✓	68.06	60.95	92.01	81.61	85.24	63.61	78.6	72.6
DFR [13]	✗/✓✓	80.69	70.37	91.93	88.40	90.89	**88.96**	82.1	74.7
TRAK (ours)	✗/✓	81.05	**75.55**	91.08	**90.03**	91.46	87.15	81.54	75.46

Table 2: Balanced accuracy and worst-group accuracy on CelebA-Age , CelebA-Blond , and Waterbirds . A double checkmark (✓✓) indicates that the method uses validation group labels for model finetuning, in addition to hyperparameter tuning.

E.2 ImageNet Accuracies

Below we included the detailed accuracies for the ImageNet experiment.

Class	**Method**	Class-Level		ImageNet-Level
(bias)		Balanced Accuracy	Worst Group Accuracy	Overall Accuracy
Red Wolf *(Red Coat)*	ERM	46.87	22.62	63.97
	TRAK	65.63	**52.38**	63.71
	Auto-TRAK	59.94	39.29	63.87
Tench *(Presence of human)*	ERM	85.10	78.12	63.97
	TRAK	90.73	**86.88**	63.84
	Auto-TRAK	86.67	80.00	63.97
Cauliflower *(Not Cooked)*	ERM	77.81	63.64	63.97
	TRAK	85.77	**79.55**	63.70
	Auto-TRAK	86.73	79.40	63.75
Strawberry *(Not on a plate)*	ERM	58.93	35.58	63.97
	TRAK	70.49	**51.92**	63.88
	Auto-TRAK	68.99	50.48	63.79

Table 3: Auto-TRAK identifies and mitigates biases in ImageNet. For four ImageNet classes, a bias was identified from inspecting the TRAK PCA directions. Then Auto-TRAK is applied in order to mitigate the bias for that class. Auto-TRAK is able to improve the worst group accuracy for the targeted class without significantly changing the overall ImageNet accuracy.

F Case Study: Interpreting the Flagged Data

What type of data does our method flag? In particular, do the examples we identify as driving the targeted bias share some common characteristics? To test this hypothesis, in Figure F.1 we inspect the data flagged by our method and identify subpopulations within the majority groups that are disproportionately responsible for the bias. Then, in Figure F.2 we retrain the model after excluding *all* training examples from the identified subpopulations and show that this is a viable strategy for mitigating the bias in the model's predictions.

F.1 Identifying subpopulations responsible for model bias

Consider the running example from Figure 1 where we train a model on the `CelebA-Age` dataset to predict whether a person is "young" or "old" in the presence of a spurious feature, ("man"/"woman") (in the `CelebA-Age` dataset, young women and old men are overrepresented). In this setup, we have access to a number of group annotations both in the training and validation sets. For instance, each training example has a label indicating whether the person is wearing eyeglasses.

While TRAK and Auto-TRAK do not require group annotations for the training set, we can use these annotations to inspect the data flagged by our methods. Specifically, we calculate the average attribution score of the training examples in each subpopulation (see Figure 6). We consider subpopulations within the cartesian product of labels and group annotations, e.g., subpopulations of the form ("young", "wearing eyeglasses"). I We find that subpopulations such as "5 o'clock shadow" and "busy eyebrows" have particularly negative attribution scores for the old class, while "gray hair" is particularly negative for the young class. In Figure 7, we show examples from the subpopulations with the most negative attribution scores. Indeed, we observe that a large fraction of the examples in these subpopulations contain labeling errors (e.g., platinum blond instead of gray hair).

F.2 Retraining without identfied subpopulations

In Figure F.1, we identified subpopulations that have overwhelmingly negative attribution scores. A natural, interpretable strategy for mitigating the bias in the model's predictions is to exclude these

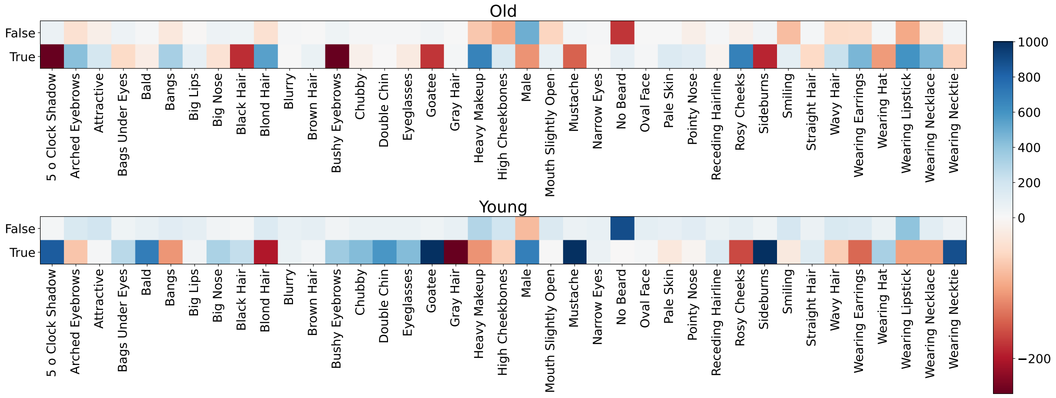

Figure 6: Average attribution score of the training examples in each subpopulation.

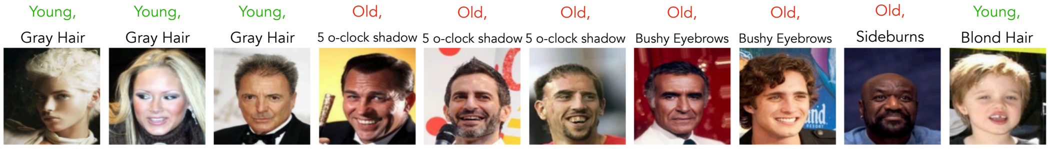

Figure 7: Randomly sampled examples from the subpopulations with the most negative attribution scores.

subpopulations from the training set. To explore this approach, we exclude the five subpopulations with the most negative attribution scores on average from the CelebA-Age dataset: "Young" + "Gray Hair", "Old"+ "5 o'Clock Shadow", "Old" + "Bushy Eyebrows", "Young" + "Blond Hair", and "Old" + "Sideburns"

After retraining the model on this modified training set, we get a worst-group accuracy of 68.4%—an approximately 12 percentage-points improvement over the worst-group accuracy of the original model (56.7%).