The Impact of Coreset Selection on Spurious Correlations and Group Robustness

Amaya Dharmasiri[1] **William Yang**[1] **Polina Kirichenko**[2]

Lydia T. Liu[1] **Olga Russakovsky**[1]

[1]Princeton University [2]FAIR at Meta

{amayadharmasiri, williamyang, ltliu, olgarus}@princeton.edu,
polkirichenko@meta.com

Appendix

This appendix includes further details on the experiment setup and analysis of the paper "The Impact of Coreset Selection on Spurious Correlations and Group Robustness".

- A - Details of all the datasets used in our extensive analysis, including their bias levels
- B - Implementation details of the analysis pipeline, model details and hyperparameters
- C.1 - Additional results corresponding to Section **??** of the main paper
- C.2 - Additional results corresponding to Section **??** of the main paper
- C.3 - Additional results corresponding to Section **??** of the main paper
- C.4 - Further exploration into difficult coresets in small data regime

Our results and code are publicly available here

A Datasets, Characterization scores, and Policies

cMNIST (c-Mn) [1]- A simple synthetic version of the MNIST [2] dataset where colors have been added to the images of the numbers. Each digit is spuriously correlated with a specific color.

Waterbirds (WB) [3]- constructed by placing images from the Caltech-UCSD Birds-200-2011 [4] dataset over backgrounds from the Places [5] dataset. The task is to classify whether a bird is a landbird or a waterbird, where the spurious attribute is the background (water or land).

Urbancars [6]- The task is the classification of car images into urban cars and country cars. There are 2 different spurious attributes that result in three different sub-datasets. **Urbancars-C (UC-C)** has a co-occurring object from urban and country contexts as the spurious feature, whereas **Urbancars-B (UC-B)** has backgrounds as the spurious feature. **Urbancars-A (UC-A)** has both spurious features resulting in 8 different subgroups.

Metashift (MSh) [7] MetaShift is a general method of creating image datasets from the Visual Genome project [8]. We use the Cat vs. Dog dataset, where the spurious attribute is the image background. Cats and more likely to be indoors, and dogs are more likely to be outdoors. We use the "unmixed" version according to the original implementation.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

Nicospurious (Nic-S) [9, 10] NICO++ is a large-scale benchmark for domain generalization. We only use their training dataset, which consists of 60 classes and 6 common attributes (autumn, dim, grass, outdoor, rock, water). To transform this dataset into the spurious correlation setting, we use the method followed by [10]

Civilcomments (CC) [11]- a text classification task where the goal is to classify a given comment as "toxic" or "neutral". Following prior works [12] we use the coarse version of the dataset where the presence of the spurious feature entails the comment containing mentions of any of these categories: male, female, LGBT, black, white, Christian, Muslim, other religion. The presence of this spurious feature is correlated with the label "toxic".

MultiNLI (MNL) [13, 3]- is also a text classification task where a pair of sentences belongs to one of the three classes: Negation, Entailment, and Neutral. Spurious feature is the presence of negation words such as "no" or "never" and it is spuriously correlated with the Negation class.

Celeb-A hair (Cel-A) [14]- We select to implement the binary classification on the hair-color attribute to "Blond" and "non-Blond". The gender of the person is claimed to be the spurious feature. The correlation is between Female gender and being blonde.

Dataset	#Classes	#Attributes	MaxGroup	MinGroup	Bias Level
cMNIST [1]	10	10	5890 (10.71%)	13 (0.02%)	10.38
Waterbirds [3]	2	2	3498 (72.95%)	56 (1.17%)	3.67
Urbancars-C [6]	2	2	3800 (47.50%)	200 (2.50%)	1.90
Metashift [7]	2	2	789 (34.67%)	196 (8.61%)	1.41
Civilcomments [11]	2	2	148186 (55.08%)	12731 (4.73%)	1.45
Nico-spurious [10, 9]	6	6	3030 (32.53%)	6 (0.06%)	11.06
Urbancars-B [6]	2	2	3800 (47.50%)	200 (2.50%)	1.90
Urbancars-A [6]	2	4	3610 (45.12%)	10 (0.12%)	1.99
MultiNLI [3, 13]	3	2	67376 (32.68%)	1521 (0.74%)	2.28
Celeb-A hair [14]	2	2	71629 (44.01%)	1387 (0.85%)	1.62

Table 1: D**ataset statistics including the number of classes, attributes, the largest/smallest subgroups, and bias levels.**

B Experiment settings, models, and hyperparameters

B.1 Class balancing

Prior work has shown that sample importance scores when used directly as a coreset selection can cause unintended class imbalances in the resulting coreset [15]. Another work [16] also perform a form of class balancing to ensure that none of the underrepresented classes are completely excluded from the selected subset. Since our experiments involve very small coreset sizes, and since the class labels are readily available, we implement a uniform class balancing strategy.

However, it should be noted that the datasets originally have imbalanced class distributions. Therefore, we calculate the ideal number of samples that each class should represent for a desired coreset size (equal proportions from all available classes), then if a particular class does not have enough samples, we iteratively divide the shortfall among the remaining classes until a distribution as close as possible to uniform is obtained.

B.2 Baselines

Once the class-balancing has been applied and the number of samples to be picked from each class is calculated, the Random (R) selection policy uses uniform random selection on separate classes.

Random-groupbalanced (R-Gbal) baseline is implemented as an oracle baseline since it utilizes the group labels of each samples, that we assume we do not have in the current setting. First, based on the class-balancing constraint, we calculate the number of samples that should/could be sampled

from each class. Then, within each class, we calculate the number of samples from each group that can be sampled such that the group distribution within each class is as close to uniform as possible. If a group does not have enough samples to create a uniform distribution, the shortfall is iteratively divided equally among the remaining groups until they run out of samples. This way, for a given size of coreset, the R-Gbal baseline selects the most group-balanced coreset possible without repeating the same samples (oversampling).

B.3 Training surrogate model

For datasets Waterbirds, Urbancars, Metashift, Nicospurious, Celeb-A hair, we trained a ResNet50 [17] initialized with pretrained weights from Imagenet to calculate the sample-level scores for the learning-based selection methods. Following the setting proposed by [18], we trained the models with SGD with a constant learning rate of 0.001, momentum of 0.9, batch size 32 and a weight decay of 0.01. Following the previous work [19], for EL2N and Uncertainty, we trained the model for 20 epochs, and for Forgetting, we trained for 200 epochs

For cMNIST, we trained a ResNet18 [17] initialized with pretrained weights from Imagenet to calculate the sample-level scores for the learning-based selection methods. We trained the models with SGD with a constant learning rate of 0.001, momentum of 0.9, batch size 32 and a weight decay of 0.01. For EL2N and Uncertainty, we trained the model for 20 epochs, and for Forgetting, we trained for 200 epochs

For Civilcomments, MultiNLI, we trained a pretrained Bert [20] model with Adam with learning rate 1e-5 and momentum 0.9 to calculate the sample-level scores for the learning-based selection methods. For EL2N and Uncertainty, we trained the model for 5 epochs, and for Forgetting, we trained for 20 epochs

For embedding-based methods: **SelfSup** and **SupProto**, we used the same model and pretrained weights as above, but did not train the feature extractor on the specific dataset; instead we extract the features for each sample from the penultimate layer. For the fine-tuned versions of the embedding-based scores: **SelfSup (finetuned)** and **SupProto (finetuned)** were first fine-tuned with supervision using the same training setting as EL2N, and the features are then extracted from the penultimate layer.

B.4 Training downstream model

The same training recipe and models as the surrogate model were used here, except for the number of epochs trained. Since we compare models trained on a variety of coreset sizes, we keep the number of training iterations for each model constant. We train each model for a specific number of epochs such that the total number of iterations is equal to the number of iterations had the model been trained on the complete dataset for n epochs. (Eg: for a coreset of size 2%, and n is 100, the scaled number of training epochs would be $100/0.02 \simeq 5000$). We set n for each dataset as follows: n =100 for **cMNIST, Waterbirds, Urbancars, Metashift, Nicospurious**, n =50 for **Celeb-A h**, and n =10 for **Civilcomments, MultiNLI**.

All models were trained with standard ERM with Stochastic Gradient Descent. All individual trainings were done on RTX-3090 GPUs with 24GB of VRAM. Total estimated compute for all experiments of this work is around 7,500 GPU hours.

C Extended results

C.1 Embedding-based characterization scores run a lower risk of inadvertently exacerbating bias compared to learning-based characterizations

Here we present the extended results corresponding to Section **??** from the main paper. Figure 1 contains the average precision evaluation of each characterization score on each dataset, when

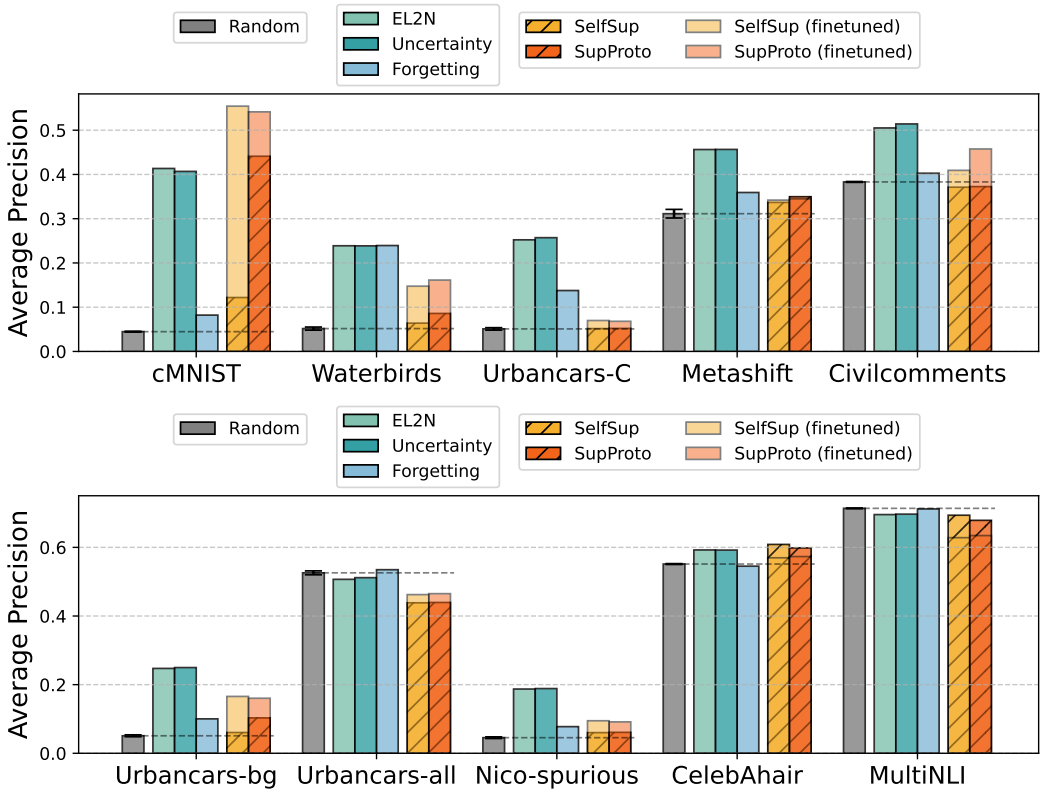

Figure 1: **Classifying bias-conflicting samples using characterization scores.** We measure the Average Precision of three learning-based methods (EL2N [19], Uncertainty [21] and Forgetting [22]) and two embedding-based methods (SelfSup [16] and SupProto [23]) at classifying bias-conflicting vs bias-aligning samples. The shaded bars on the embedding-based methods represent the results for scores generated from fine-tuned embeddings.

evaluated as a predictor for detecting bias conflicting samples. The random baseline is calculated by randomly ordering all the samples and then thresholding them at each level to calculate average precision, whereas the error bars represent the standard deviation. Therefore the average precision on random selection represents the overall proportion of bias-conflicting samples in the dataset. We see that across all datasets, leaning-based characterizations capture a much stronger signal that distinguishes bias conflicting samples from bias-aligning samples. We stipulate that this strong correlation between the characterization score and the bias-alignment of the samples can in turn cause inadvertent bias exacerbation when used as a metric for data selection. On the more challenging real-world datasets (Urbancars [6], Metashift [7], and Civilcomments [11], and Nico-spurious [10]), embedding-based methods do not appear to order the samples according to their bias levels (i.e., have near-random AP); even finetuning these embeddings (depicted by the shaded bars) does not significantly change these findings. It is also noteworthy that for datasets with more natural and complex spurious features (Urbancars-all [10], CelebAhair [14], and MultiNLI [13, 3], learning-based and embedding-based characterizations seem to capture signals of comparable strength.

C.2 Coreset bias level is not a consistent indicator of downstream robustness

In this section we include the extended results corresponding to Section **??** of the main paper. Figure 2 shows bias-levels, worst-group accuracy, and average accuracy for Difficult and Easy selection policies using EL2N [24] scores along with the baselines. The observations we outlined in the main paper consistently appear across all the datasets of the analysis. Coresets selected using the Difficult policies consistently have lower level of bias compared to those selected using the Easy policy. However, in the middle column we see that it does not always lead to improved robustness: **the Difficult coresets lead to more robust classifiers only when the coreset size is "sufficiently large."** What constitutes "sufficiently large" appears to further vary empirically between the 10 datasets used in this analysis. Furthermore, in the small data regime, the robustness of models for Difficult coresets becomes unintuitively low, despite the bias levels being the lowest out of all policies.

Corresponding numerical results (along with Median and Stratified selection policies) are shown in Table 2 and Table 3 respectively for moderate coreset sizes and very small coreset sizes. In the moderately sized coresets (40% and 60%), Difficult selections of EL2N scores yield high robustness, however this pattern is not consistent in the small coresets of 10% and 5%.

Extended results for all scoring methods and all selection policies for 40% selection rate is shown in Table 4

	Baselines		EL2N [24] scores				SelfSup [16] scores			
Dataset	R	R-Gbal	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas
WB	58.2 (96.6)	83.3 (97.7)	**74.6** (98.0)	73.1 (97.3)	50.5 (96.4)	30.2 (95.2)	50.9 (96.1)	51.0 (96.4)	69.2 (97.1)	37.1 (90.2)
c-Mn	62.0 (99.8)	84.4 (99.9)	**87.2** (99.9)	74.0 (99.9)	0.0 (99.0)	0.0 (98.4)	83.0 (99.7)	83.5 (99.8)	44.0 (99.7)	0.0 (98.7)
CC	77.4 (88.8)	72.0 (89.6)	63.0 (84.3)	51.7 (91.8)	63.9 (80.5)	43.2 (68.1)	77.8 (87.7)	78.4 (88.3)	70.3 (90.4)	**79.3** (86.0)
MSh	73.8 (88.3)	78.5 (87.2)	**78.5** (90.6)	63.1 (89.3)	66.5 (88.5)	58.6 (86.4)	73.3 (88.3)	69.2 (90.3)	74.3 (89.1)	56.9 (87.5)
Nic-S	44.0 (93.9)	40.0 (94.9)	**44.0** (95.8)	44.0 (95.3)	34.0 (95.1)	32.0 (93.7)	44.0 (95.0)	16.0 (93.3)	34.0 (93.4)	36.0 (94.3)
UC-C	52.0 (86.8)	61.6 (83.7)	54.0 (89.2)	51.2 (87.6)	46.4 (70.3)	49.6 (63.7)	42.0 (86.7)	48.4 (88.4)	**54.4** (85.0)	44.0 (81.7)
UC-B	44.8 (90.2)	64.0 (87.6)	**52.0** (90.1)	44.8 (89.5)	26.4 (90.4)	15.6 (91.8)	43.2 (87.9)	48.0 (88.8)	41.6 (90.6)	26.4 (89.4)
UC-C	16.8 (96.4)	50.4 (96.4)	**23.2** (97.2)	19.2 (97.0)	6.4 (94.9)	7.2 (94.4)	11.2 (96.5)	17.6 (97.0)	20.8 (96.6)	9.6 (96.5)
MNL	60.5 (78.8)	57.6 (78.6)	46.0 (61.5)	**65.6** (74.7)	61.8 (79.0)	46.0 (78.1)	55.2 (79.4)	60.8 (79.6)	55.7 (78.2)	58.3 (78.8)
Cel-A	58.9 (93.4)	71.1 (92.6)	38.3 (94.9)	47.8 (95.1)	68.9 (93.2)	**79.0** (88.0)	41.7 (94.4)	66.1 (92.6)	63.3 (94.4)	78.9 (88.8)

(a) 40 percent

	Baselines		EL2N [24] scores				SelfSup [16] scores			
Dataset	R	R-Gbal	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas
WB	67.3 (97.3)	77.6 (98.0)	73.2 (97.9)	75.6 (97.4)	28.6 (95.3)	26.5 (94.9)	73.7 (97.6)	64.5 (97.1)	**75.9** (97.5)	58.4 (94.8)
c-Mn	78.9 (99.5)	79.8 (99.5)	63.1 (99.5)	57.3 (99.6)	1.0 (99.1)	0.0 (99.0)	78.0 (99.7)	**81.3** (99.8)	65.0 (99.7)	0.8 (99.4)
CC	72.4 (90.1)	70.8 (89.7)	55.4 (92.1)	53.7 (92.0)	55.5 (75.7)	57.3 (76.8)	69.4 (90.0)	**71.2** (90.0)	69.7 (90.5)	61.6 (91.1)
MSh	75.4 (90.7)	75.4 (90.0)	70.8 (90.7)	70.8 (90.2)	70.7 (89.1)	67.5 (88.5)	75.9 (90.7)	75.4 (90.8)	**76.4** (89.7)	61.5 (89.1)
Nic-S	42.0 (94.7)	40.0 (95.2)	40.0 (96.2)	38.0 (95.3)	34.0 (95.6)	32.0 (94.9)	40.0 (96.3)	34.0 (95.4)	34.0 (95.0)	**44.0** (94.9)
UC-C	49.2 (87.5)	56.4 (85.2)	**54.4** (89.3)	53.2 (89.3)	49.6 (74.9)	49.2 (73.7)	48.4 (87.7)	52.4 (90.1)	52.8 (88.6)	50.8 (85.4)
UC-B	48.8 (88.7)	58.0 (88.0)	**53.6** (90.7)	49.6 (89.7)	26.0 (90.9)	24.4 (91.0)	47.6 (89.9)	49.2 (89.4)	49.6 (90.4)	39.2 (90.9)
UC-A	22.4 (97.2)	32.8 (97.5)	**23.2** (97.3)	21.6 (97.5)	10.4 (96.0)	9.6 (95.9)	16.8 (97.1)	20.8 (96.9)	20.8 (97.2)	18.4 (97.3)
MNL	58.0 (80.1)	65.5 (80.5)	65.3 (73.9)	65.0 (79.6)	55.2 (80.4)	54.5 (80.4)	**67.2** (79.6)	63.7 (80.7)	65.9 (79.4)	66.0 (79.5)
Cel-A	73.3 (93.9)	56.7 (95.3)	61.1 (95.2)	46.1 (95.2)	68.9 (94.3)	**71.7** (93.0)	63.3 (94.1)	37.2 (95.8)	55.6 (95.4)	58.9 (94.5)

(b) 60 percent

Table 2: **Worst-group accuracies and (Average accuracies) for different selection policies. For moderate coreset sizes: 40% and 60%.** The highest values of worst-group-accuracies are bolded, with second highest values underlined. The least robust, indicated by the least value for worst-group accuracy is shaded in brown. In general, Difficult selection policies with EL2N scores yield robust classifiers.

Dataset	Baselines		EL2N [24] scores				SelfSup [16] scores			
	R	R-Gbal	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas
WB	44.7 (95.3)	89.0 (92.2)	31.9 (85.2)	**51.1** (95.5)	42.7 (95.3)	24.8 (94.8)	35.1 (94.9)	26.3 (94.2)	41.3 (95.0)	29.1 (93.9)
c-Mn	0.0 (99.2)	95.3 (99.2)	**88.6** (99.9)	8.0 (99.4)	0.0 (97.1)	0.0 (96.9)	36.9 (99.5)	24.7 (99.4)	0.0 (98.2)	0.0 (96.5)
CC	66.6 (77.3)	75.0 (81.2)	7.2 (11.1)	15.9 (19.1)	58.0 (91.0)	29.6 (56.0)	42.4 (62.6)	62.5 (75.9)	68.3 (78.9)	**70.7** (80.8)
MSh	55.4 (85.4)	75.4 (87.5)	6.2 (33.0)	21.2 (51.2)	**56.9** (85.4)	50.3 (83.0)	55.0 (85.1)	46.2 (84.8)	52.3 (86.4)	38.5 (83.5)
Nic-S	34.0 (90.6)	60.0 (88.8)	34.0 (90.2)	**38.0** (89.9)	28.0 (93.1)	8.0 (88.7)	26.0 (92.3)	32.0 (93.0)	30.0 (90.0)	16.0 (82.2)
UC-C	44.4 (75.5)	52.8 (60.5)	27.2 (59.7)	44.8 (76.0)	47.6 (66.8)	46.0 (58.7)	21.2 (67.3)	40.8 (75.8)	**48.0** (70.7)	38.8 (74.6)
UC-B	21.6 (89.3)	72.0 (78.1)	14.0 (70.3)	28.4 (85.1)	17.2 (90.0)	9.2 (90.4)	9.2 (75.2)	**34.4** (84.6)	20.4 (89.9)	15.6 (86.0)
UC-A	6.4 (95.1)	62.4 (88.8)	11.2 (76.3)	**15.2** (93.2)	6.4 (94.5)	3.2 (92.5)	0.8 (82.3)	8.8 (93.2)	9.6 (95.5)	4.8 (95.2)
MNL	32.5 (69.4)	62.3 (69.4)	3.2 (17.5)	22.2 (30.7)	41.7 (74.9)	37.7 (64.4)	25.1 (60.1)	34.3 (68.4)	**51.2** (71.1)	29.7 (70.5)
Cel-A	47.2 (94.8)	83.3 (92.0)	40.0 (82.1)	51.1 (89.2)	41.1 (93.6)	80.6 (88.0)	**81.4** (86.7)	64.4 (94.0)	62.2 (93.2)	80.9 (85.3)

(a) 5 percent

Dataset	Baselines		EL2N [24] scores				SelfSup [16] scores			
	R	R-Gbal	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas
WB	40.5 (95.3)	88.6 (95.6)	**68.5** (96.7)	67.4 (96.3)	47.2 (95.9)	29.1 (95.1)	40.8 (94.8)	34.7 (95.3)	55.9 (96.0)	23.9 (92.4)
c-Mn	24.8 (99.4)	95.0 (99.5)	**87.9** (99.9)	47.0 (99.8)	0.0 (98.1)	0.0 (97.3)	69.4 (99.6)	73.4 (99.6)	0.0 (98.8)	0.0 (96.5)
CC	77.9 (85.3)	72.2 (80.3)	5.0 (10.0)	18.7 (21.9)	**80.6** (88.4)	34.8 (58.1)	57.0 (73.4)	66.6 (79.3)	68.3 (79.8)	68.4 (79.3)
MSh	58.5 (88.4)	81.7 (88.8)	11.4 (42.9)	63.1 (82.2)	**67.5** (87.3)	54.5 (84.7)	57.1 (85.4)	56.9 (87.7)	58.5 (86.8)	38.5 (84.7)
Nic-S	32.0 (94.2)	40.0 (81.2)	36.0 (95.4)	**50.0** (92.3)	30.0 (94.0)	16.0 (91.2)	30.0 (94.5)	26.0 (93.5)	26.0 (91.9)	28.0 (88.3)
UC-C	48.8 (75.4)	60.0 (67.0)	**53.6** (83.0)	46.0 (80.0)	46.4 (65.9)	47.6 (57.4)	27.6 (73.0)	46.4 (81.8)	48.8 (74.8)	40.4 (75.9)
UC-B	32.4 (88.8)	76.4 (82.1)	**44.4** (86.4)	28.0 (88.1)	21.6 (90.6)	11.2 (90.7)	17.6 (80.6)	36.0 (88.1)	26.0 (89.9)	17.6 (88.1)
UC-A	8.8 (95.7)	64.8 (92.7)	**27.2** (94.3)	13.6 (94.9)	5.6 (94.4)	4.8 (92.4)	3.2 (88.9)	11.2 (96.4)	11.2 (95.5)	6.4 (94.8)
MNL	58.4 (73.6)	63.2 (73.2)	7.7 (19.5)	12.7 (27.6)	46.5 (77.8)	45.6 (70.6)	38.6 (67.7)	**48.6** (73.7)	44.9 (74.1)	43.6 (74.2)
Cel-A	75.6 (92.0)	76.1 (92.0)	41.7 (90.1)	63.3 (91.8)	72.8 (92.6)	70.2 (79.4)	**77.2** (84.8)	76.7 (91.4)	48.3 (92.1)	76.6 (82.1)

(b) 10 percent

Table 3: **Worst-group accuracies and (Average accuracies) for different selection policies. For very small coreset sizes: 5% and 10%.** The highest values of worst-group-accuracies are bolded, with second highest values underlined. The least robust, indicated by the least value for worst-group accuracy is shaded in brown. Difficult selection suffers from a large drop in both average and worst-group accuracies, especially in EL2N. Selection policies that incorporate less difficult samples tend to yield comparatively higher robustness.

Dataset	Baselines		EL2N [19]				Uncertainty [21]				Forgetting [22]				SelfSup [16]				SupProto [23]			
	R	R-gbal	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas	Diff	Strat	Med	Eas
C-Mn	62.0	84.4	**87.72**	74.0	0.0	0.0	37.8	**82.3**	1.0	0.0	**9.9**	4.0	0.0	0.0	83.0	**83.5**	44.0	0.0	**87.0**	48.5	0.0	0.0
WB	58.2	83.3	**74.6**	73.1	50.5	30.2	**72.6**	71.2	51.0	28.2	75.4	**79.6**	53.8	52.0	50.9	51.0	**69.2**	37.1	**66.5**	58.8	59.4	30.0
UC-c	52.0	61.6	**54.0**	51.2	46.4	49.6	54.8	**55.2**	46.4	50.4	51.2	**54**	31.6	31.6	42.0	48.4	**54.4**	44.0	46.8	**50.4**	50.0	44.8
MSh	73.8	78.5	**78.5**	63.1	66.5	58.6	**73.8**	69.2	70.2	65.4	71.7	**76.4**	46.6	48.7	73.3	69.2	**74.3**	56.9	67.0	**73.8**	72.3	58.5
CC	77.4	72.0	63.0	51.7	**63.9**	43.2	60.2	64.2	61.0	**75.2**	60.9	65.3	**82.0**	79.5	77.8	78.4	70.3	**79.3**	76.2	**77.1**	74.1	75.2
Nic-s	44.0	40.0	**44.0**	**44.0**	34.0	32.0	**44.0**	42.0	36.0	30.0	44.0	40.0	**46.0**	40.0	**44.0**	16.0	34.0	36.0	38.0	**42.0**	**42.0**	38.0
UC-b	44.8	64.0	**52.0**	44.8	26.4	15.6	52.8	**53.2**	26.0	15.6	47.6	50.8	**54.0**	**54.0**	43.2	**48.0**	41.6	26.4	**53.2**	52.4	36.8	13.6
UC-a	16.8	50.4	**23.2**	19.2	6.4	7.2	22.4	**26.4**	8	7.2	20.0	**23.2**	16.0	16.0	11.2	17.6	**20.8**	9.6	**20.8**	19.2	17.6	5.6
MNL	60.5	57.6	46.0	**65.6**	61.8	46.0	61.5	**64.0**	53.8	49.2	**59.5**	57.8	52.4	52.4	55.2	**60.8**	55.7	58.3	-	**63.9**	62.6	55.4
Cel-h	58.9	71.1	38.3	47.8	68.9	**79.0**	42.2	42.2	62.8	**78.9**	27.2	42.8	**64.4**	**64.4**	41.7	66.1	63.3	**78.9**	50.0	59.4	63.9	**82.2**

Table 4: **Worst-group accuracies of different selection policies within each scoring methods (learning-based: EL2N [19], Uncertainty [21], Forgetting [22], and embedding-based: SelfSup [16], SupProto [23]) at 40% selection rate.** The highest worst-group-accuracies within each scoring method for each dataset are bolded.

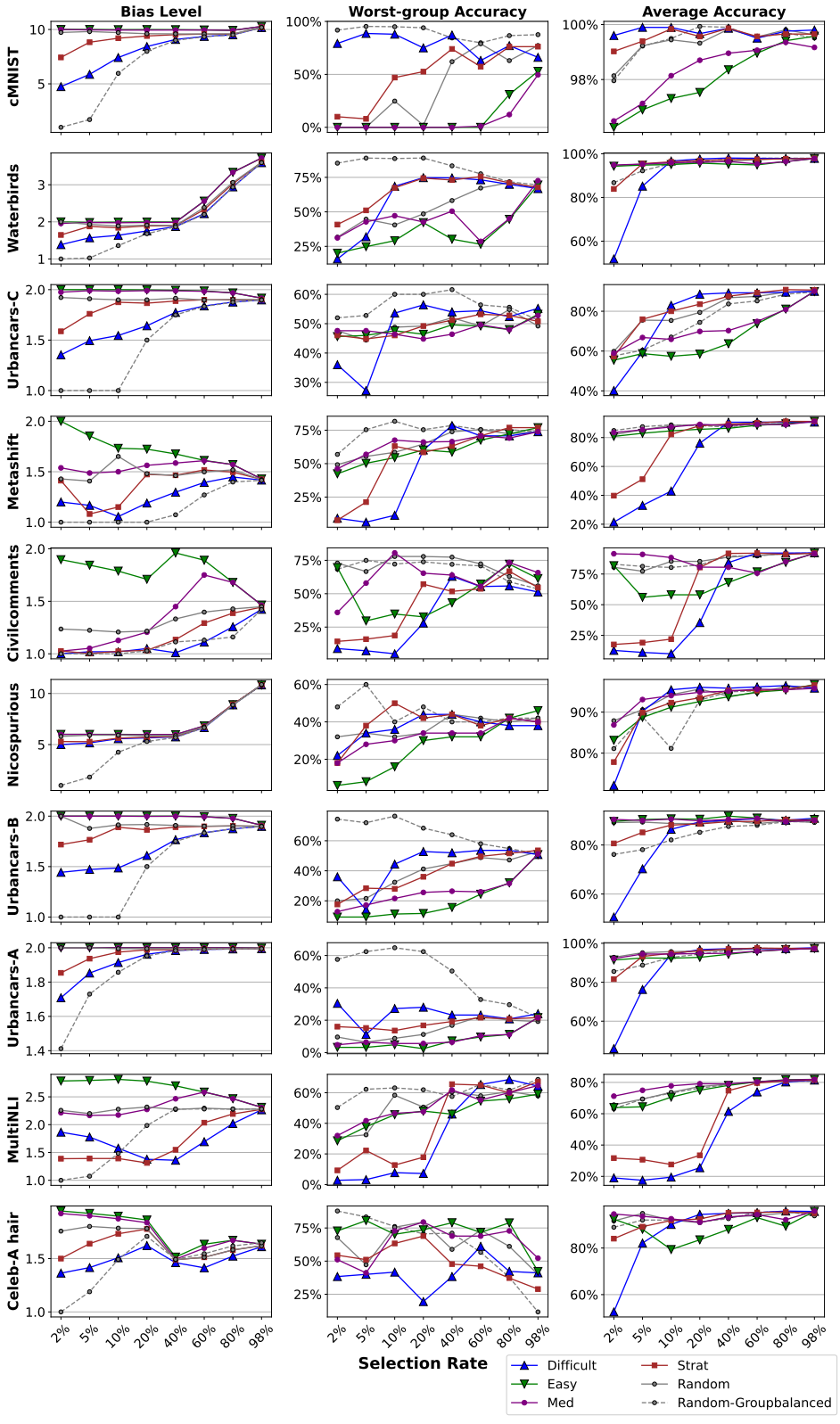

Figure 2: **Data bias and classifier accuracies for different selection policies using EL2N scores.** Selecting the Difficult samples typically results in less biased coresets and corresponding more robust (highest worst-group accuracy) classifiers than Easy samples at higher selection rates. The Difficult samples also lead to more robust models than Random selection. However, as coreset size gets smaller, we see a significant drop in average and worst-group accuracies for Difficult samples. In such settings, Stratified and Median policies, which are consistently more biased than Difficult, counter-intuitively yield higher robustness.

C.3 Trading off most difficult bias-conflicting samples to improve robustness

This section includes the extended results corresponding to Section **??** of the main paper. Difficult* selection policy is a simple heuristic where a small percentage (3%) of the highest scoring samples is removed from Difficult selection. Bias levels, worst-group accuracy, and average accuracy for this heuristic policy along with the rest of the policies are applied on EL2N scores for all the datasets of the analysis as shown in Figure 3. We can see that all methods: Difficult*, Stratified, and Median make the coresets progressively more biased compared to Difficult selection. However, they result in improved robustness (worst-group accuracy) in cases where Difficult policy has catastrophically low robustness.

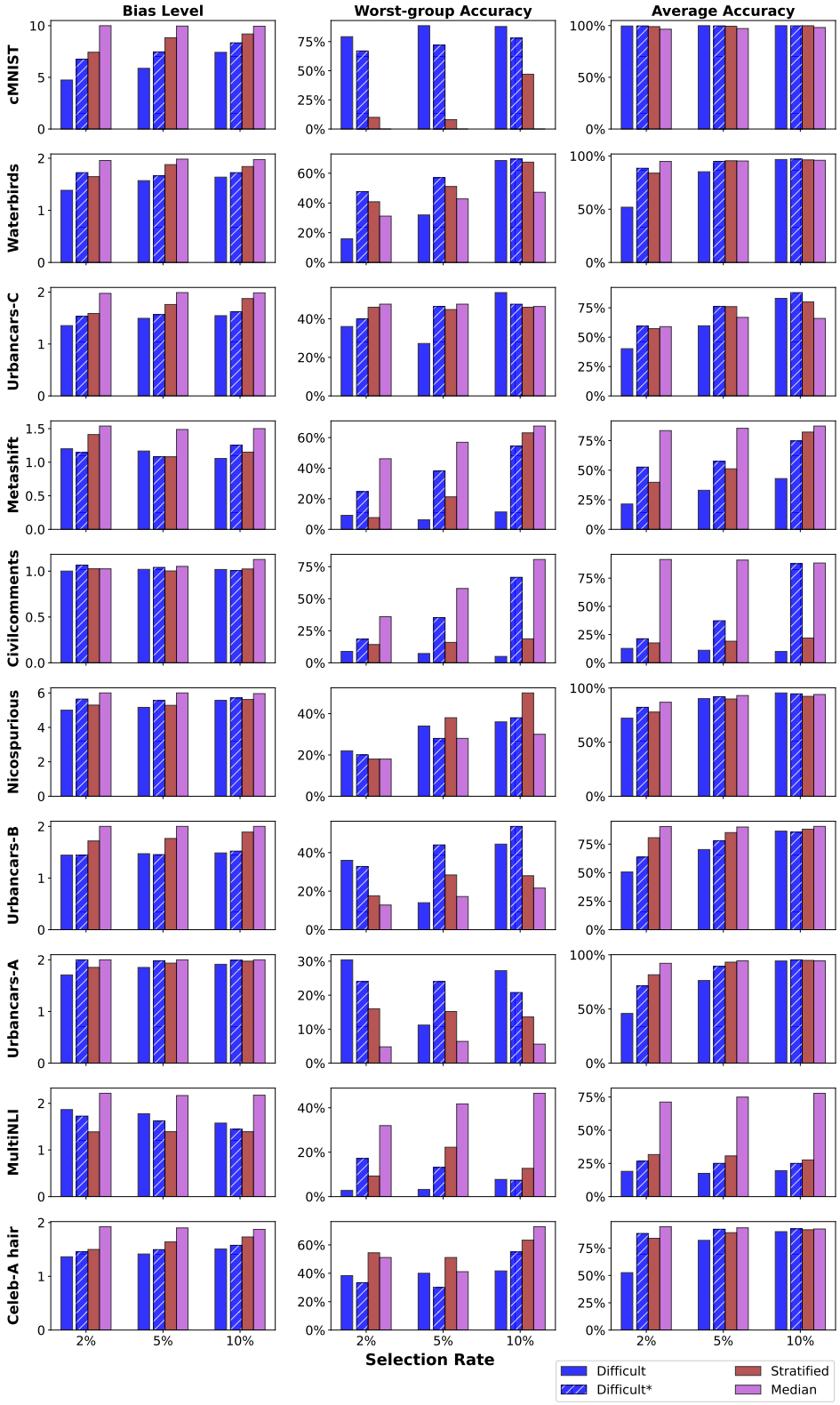

Figure 3: **Trading off most difficult minority samples to achieve higher robustness.** Difficult*, Median and Stratified selection policies often make the selected coreset slightly biased, however, they improve the robustness of the downstream models, in settings where Difficult selection has a catastrophic drop in performance.

C.4 Further exploration into catastrophic loss of accuracy for difficult coresets in small data regime

We inspect train-test accuracy gaps for models trained on small-data regime coresets and observe that all models achieve 100% training accuracy. The drop in average accuracy for difficult coresets in the small data regime therefore suggests a high level of overfitting compared to other policies, also consistent with claims of Sorscher et al [16] as we discussed in Section **??**.

It has been hypothesized that the catastrophic drop in accuracy for difficult coresets could be due to the coreset achieving low coverage on the data space. Zhen et al. [25] utilized the concept of p-partial r-cover to quantify this phenomenon, where r is some radius around each data of the coreset and p is the proportion of training data covered. We used their p-partial r-cover as a metric to investigate whether this pattern persists in datasets with strong spurious correlations. Using the features of ImageNet pre-trained ResNet-50, we selected a radius r where it would cover 95% of the training data. Using the r, we obtained the measured p from the selected coresets shown in Table 5. Significant decrease in p in difficult selected coresets at 5% and 2% across the datasets confirms the drop in coverage.

Selection Rate	Random (baseline)	Difficult	Easy
2%	80.3	**45.4**	82.3
5%	85.7	**58.8**	86.1
20%	93.3	92.4	90.9

(a) Waterbirds [3]

Selection Rate	Random (baseline)	Difficult	Easy
2%	61.0	**7.9**	71.1
5%	74.7	**25.0**	77.4
20%	89.6	80.0	88.4

(b) Metashift [7]

Table 5: **p-partial r-cover achieved by Difficult and Easy selection policies for Waterbirds [3] and Metashift [7] datasets.** At low selection rates, Difficult selection yeilds to significantly less coverage than Easy of Random selections (bolded).

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020.

[2] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[3] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[4] 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.

[5] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva. Places: An image database for deep scene understanding, 2016.

[6] Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others, 2023.

[7] Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. In *International Conference on Learning Representations*.

[8] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[9] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16036–16047, 2023.

[10] Sriram Yenamandra, Pratik Ramesh, Viraj Prabhu, and Judy Hoffman. Facts: First amplify correlations and then slice to discover bias. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4804, 2023.

[11] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification, 2019.

[12] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy, 2022.

[13] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018.

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[15] Artem Vysogorets, Kartik Ahuja, and Julia Kempe. Robust data pruning: Uncovering and overcoming implicit bias. *arXiv preprint arXiv:2404.05579*, 2024.

[16] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[18] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*.

[19] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[21] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*.

[22] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2018.

[23] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2022.

[24] Tejas Pote, Mohammed Adnan, Yigit Yargic, and Yani Ioannou. Classification bias on a data diet. In *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023.

[25] Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *The Eleventh International Conference on Learning Representations*.