Boosting worst-group accuracy without any group annotations

Vincent Bardenhagen[*], **Alexandru Țifrea**[*], **Fanny Yang**
Department of Computer Science
ETH Zurich, Switzerland
{vbardenha,tifreaa,fan.yang}@ethz.ch

Abstract

Despite having good average test accuracy, classification models can have poor performance on subpopulations that are not well represented in the training data. In this work, we introduce a criterion to estimate the accuracy on these populations. This allows us to design a procedure that achieves good worst-group performance and unlike previous procedures requires no group labels. We provide a sound empirical investigation of our procedure and show that it recovers the worst-group performance of methods that use oracle group annotations.

1 Introduction

Machine learning models have been remarkably successful on a broad range of prediction tasks, when performance is measured with the average error on a holdout i.i.d. test set. However, good performance on average does not guarantee that the error is uniformly low on different subpopulations. In applications where the accuracy on the subpopulations matters (e.g. instances where fair prediction is critical), metrics other than the average validation error are more informative.

One common way for such subpopulations to emerge is when a subset of the features is spuriously correlated with the target variable in data sampled from the training distribution [Geirhos et al., 2020, Minderer et al., 2020]. If prediction models use these spurious features for the decision rule, then they produce incorrect outputs on samples for which the spurious correlation does not hold, leading to subpopulations (groups) which are consistently predicted incorrectly. Ideally, we want that estimators have both good average accuracy, and also good worst-group performance. When the training distribution has imbalanced subpopulations (see Figure 1 Right), the Bayes-optimal classifier has poor performance on groups that cannot be predicted correctly using the spurious feature. Unsurprisingly, estimating the Bayes-optimal classifier via empirical risk minimization (ERM) also suffers from poor worst-group performance, with the detrimental impact of overparameterization being studied at length in Sagawa et al. [2020b].

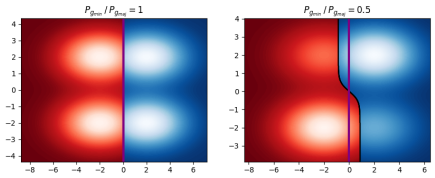

Figure 1: The Bayes classifier (**black**) aligns with the best worst-group classifier (**violet**) only for balanced training groups (left). For imbalanced groups, the Bayes classifier has low worst-group performance (right).

A number of works have made progress on mitigating the consequences of the group imbalance problem, but all these approaches require some degree of access to clean group membership labels for the training or validation data. Some of the most effective methods are inspired by distributionally-

[*]Equal contribution.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

robust optimization (DRO) [Sagawa et al., 2020a] or importance weighting/sampling [Sagawa et al., 2020a,b, Menon et al., 2021], They rely crucially on strict regularization and introduce a number of additional hyperparameters that need to be carefully tuned with respect to worst-group performance. As a consequence, all of these methods require that oracle group labels are available for some data points.

However, collecting group labels is often a difficult or sometimes straight-out impossible task in practice. For instance, in real-world applications it is often not clear what exactly the spurious features are (e.g. they could be as inconspicuous as the laterality artifacts of an X-ray device [Zech et al., 2018]), which makes it impossible to determine how to partition data into groups. Furthermore, even if we know what the spurious attributes are, it might not be possible to collect them during the data acquisition process. Public or private institutions are often not allowed to hold records on sensitive information such as race, religion, sexual orientation etc.

Despite a surge in approaches that can achieve good worst-group accuracy without group annotations for the training data, the problem of *model selection* without requiring validation group labels remains a largely unaddressed challenge. All prior works either use an annotated validation set for hyperparameter tuning [Nam et al., 2020, Liu et al., 2021] or they avoid model selection altogether and use hyperparameters identified by other works, known to lead to good worst-group performance [Sohoni et al., 2020, Creager et al., 2021].

Contributions. In this work we address this issue and propose a model selection criterion that can be applied without any group labels. To this aim we adapt a two-stage procedure introduced in prior work (e.g. [Nam et al., 2020, Liu et al., 2021, Bahng et al., 2020]) and devise proxy criteria for hyperparameter selection for the models trained during both stages. Our experiments reveal that our method significantly outperforms baselines that use no group annotated data and achieves similar performance to approaches that require group annotations on a variety of prediction tasks for image and tabular data sets.

2 Problem setting

In this section we describe the setting that we assume throughout our work. We consider classification problems in which the covariates are drawn from a mixture of groups, similar to Sagawa et al. [2020b]. All the samples from one group are assigned the same ground truth label, but a class can comprise several groups. Elements from different groups differ by features that are not predictive of the class label for all samples. However, these features can be spuriously correlated with the class labels for samples coming from certain groups.

In Figure 1 we illustrate the setting for the simple case of 2D binary data with 2 groups per class, balanced classes and isotropic Gaussian group conditional distributions. The marginal distribution of covariates is given by $x \sim P_X = \sum_{y \in \mathcal{Y}} \sum_{g \in \{0,1\}} P[X|G=g, Y=y] P_{g|y} P_y$. If the groups are balanced (i.e. $P_{0|y} = P_{1|y}, \forall y \in \mathcal{Y}$), then the Bayes classifier f_{Bayes} has simultaneously both optimal average and worst-group error. However, if some groups are significantly more represented in the training distribution than others (i.e. $P_{0|y} >> P_{1|y}$ for some $y \in \mathcal{Y}$), then f_{Bayes} is optimal with respect to the average error, but does not have the best error among all possible classifiers on the minority groups (Figure 1 right). This problem is particularly challenging when the spurious feature is highly correlated with the class label for the majority groups, meaning that a classifier could easily rely on spurious attributes for its predictions. This relationship between the worst-group and the average accuracy for scenarios with group imbalance is reminiscent of other well-studied trade-offs such as the one involving the standard and the adversarial robust risk [Tsipras et al., 2019].

Related work. As Figure 1 shows, even when given infinite training data and a well-specified hypothesis class that includes the Bayes classifier, ERM still arrives at a solution with subpar worst-group performance. Instead, with group labels for the training samples, we change the original training distribution (e.g. via importance weighting/sampling) such that the Bayes classifier of this new distribution has optimal worst-group performance. Prior works propose implicit (e.g. GDRO [Sagawa et al., 2020a]) and explicit (e.g. generating minority points [Goel et al., 2020], removing majority points [Menon et al., 2021, Sagawa et al., 2020b]) ways to "alter" the training distribution. Furthermore, a number of recent approaches do not use group annotations for the training data (e.g. Liu et al. [2021],Nam et al. [2020],Creager et al. [2021]). However, since these approaches are sensitive to the choice of some crucial hyperparameters (e.g. regularization strength, weight values for importance weighting/sampling etc), effective model selection is of utmost importance. All prior

approaches rely on access to group annotations for the validation set and select hyperparameters that lead to optimal worst-group accuracy.

3 Proposed method

We now sketch a general framework that covers a number of two-stage approaches that have been proposed for learning unbiased classifiers. We then describe a procedure for selecting hyperparameters without using group annotations.

3.1 Overview of two-stage approaches for learning unbiased predictors

First stage – Detecting error-prone samples. In the absence of group annotations for the training set, the first step consists in identifying a stand-in for the minority groups. Recall that the minority samples are also the ones for which the spurious correlation does not hold. Assuming that the spurious feature is easier to pick up from the data than the core features (e.g. spurious features have a significantly higher signal-to-noise ratio [Nam et al., 2020]), then one can recover the minority points by using a biased classifier that relies more heavily on the spurious attribute for its predictions. One can achieve the desired biased predictors by using carefully tailored loss functions [Nam et al., 2020], or through strong regularization, such as early stopping [Bahng et al., 2020, Liu et al., 2021]. Training the biased first-stage predictors amounts to solving the following optimization problem:

$$\hat{f}_{t_1,\theta_1} = \underset{f \in \mathcal{F}(t_1,\theta_1)}{\arg\min}\ \mathcal{L}(f, S), \tag{1}$$

where T_1 denotes the set of possible early stopping times and Θ_1 is the set of all hyperparameter configurations that we consider.

Ideally, the biased classifier $\hat{f}_{t_1,\theta_1}$ produces incorrect predictions at inference time on most of the minority samples. Thus, one can use the *error set* of $\hat{f}_{t_1,\theta_1}$ as a surrogate for the minority groups. We assign group label $g = 1$ to points in the error set of $\hat{f}_{t_1,\theta_1}$ to get the annotated training set:

$$\bar{S}(\hat{f}_{t_1,\theta_1}) = \{(x_i, y_i, g_i) : (x_i, y_i) \in S, g_i = \mathbb{1}[\hat{f}_{t_1,\theta_1}(x_i) \neq y_i]\} \tag{2}$$

Similarly, if group labels are not provided for the validation set, we can annotate it to obtain $\bar{V}(\hat{f}_{t_1,\theta_1})$.

Second stage – Correcting the mistakes of the biased predictor. Once we have identified the error-prone training samples, we can use a procedure like importance sampling/weighting or GDRO to mitigate the group imbalance and find classifiers that perform well uniformly across groups. For instance, one could use an importance weighted loss $\mathcal{L}_{\text{IW}}$ to up-weigh minority samples, in a procedure that resembles boosting algorithms [Breiman, 1996]. Once again, these techniques introduce additional hyperparameters (e.g. regularization strength, importance weight values etc) and choosing good values for them is essential to achieving satisfactory worst-group performance. The optimization problem for training the de-biased predictor takes the following form:

$$\hat{f}_{t_1,\theta_1,t_2,\theta_2} = \underset{f \in \mathcal{F}(t_1,\theta_1,t_2,\theta_2)}{\arg\min}\ \mathcal{L}_{\text{IW}}(f, \bar{S}(\hat{f}_{t_1,\theta_1})), \tag{3}$$

where T_2 and Θ_2 are the set of early stopping times and the hyperparameter grid for the second stage.

3.2 Prior work: Model selection with oracle group labels

Prior work [Nam et al., 2020, Liu et al., 2021] propose to tune hyperparameters by maximizing the worst-group validation accuracy directly, since group annotations are available on the validation set. In this setting, there are two possible strategies for hyperparameter tuning, as illustrated in Table 1. One option is to consider separate model selection criteria for the two stages [Nam et al., 2020, Bahng et al., 2020]. Hyperparameters for the first stage are chosen so as to bias the classifier towards making mistakes on minority-group samples, while predicting majority points well. Then second-stage hyperparameters are selected such that worst-group accuracy is maximized. Alternatively, one can choose both first and second-stage parameters using a single objective: improving the final performance on minority groups [Liu et al., 2021].

3.3 Proposed model selection without group annotations

Model selection for the first stage. We propose to fix the regularization strength, and select the remaining hyperparameters so that the *average* validation accuracy is maximized, which implicitly promotes more biased classifiers. Concretely, for a given model class, we only search for empirical risk minimizers in the set of predictors that can be obtained after only one epoch of gradient descent training. This severe early stopping restricts the set of possible classifiers to simple functions that can

Prior work (known validation group labels)	**Ours** (no group annotations)
Stage-wise model selection $t_1^*, \theta_1^* \in \underset{t_1, \theta_1 \in \mathrm{T}_1 \times \Theta_1}{\arg\max}\ \mathrm{PredictionBias}(\hat{f}_{t_1,\theta_1}, \bar{V}_{\text{oracle}})$ $t_2^*, \theta_2^* \in \underset{t_2, \theta_2 \in \mathrm{T}_2 \times \Theta_2}{\arg\max}\ \mathrm{WgAcc}(\hat{f}_{t_1^*,\theta_1^*,t_2,\theta_2}, \bar{V}_{\text{oracle}})$	$t_1^* = 1$ $\theta_1^* \in \underset{\theta_1 \in \Theta_1}{\arg\max}\ \mathbf{AvgAcc}(\hat{f}_{t_1,\theta_1}, V)$
Simultaneous model selection for both stages $t_1^*, \theta_1^*, t_2^*, \theta_2^* \in \underset{\substack{t_1,\theta_1 \in \mathrm{T}_1 \times \Theta_1 \\ t_2,\theta_2 \in \mathrm{T}_2 \times \Theta_2}}{\arg\max}\ \mathrm{WgAcc}(\hat{f}_{t_1,\theta_1,t_2,\theta_2}, \bar{V}_{\text{oracle}})$	$t_2^*, \theta_2^* \in \underset{t_2,\theta_2 \in \mathrm{T}_2 \times \Theta_2}{\arg\max}\ \mathcal{C}(\hat{f}_{t_1^*,\theta_1^*,t_2,\theta_2}, \{\bar{V}(\hat{f}_i)\}_{i=1}^K)$

Table 1: Model selection using prior approaches (**Left**) and with our method that uses no group annotations (**Right**). We **highlight** the important modifications that we propose.

achieve better validation accuracy by relying on the easier to learn, spurious features, and ignoring the core features altogether. To construct the error set of the training data, we perform two-fold cross-validation and repeatedly extract samples with incorrect predictions from the holdout fold.

Model selection for the second stage. To approximate minority group accuracy we could measure the accuracy on the error set of the biased model trained during the first stage. A naive approach that uses the accuracy on this error set as a criterion for hyperparameter tuning, leads to suboptimal results. The reason is that one error set contains minority points, but also many *majority* samples that are predicted incorrectly (see Appendix G.2 for more details). Instead, we propose to use an ensemble of biased first-stage predictors and the intersection of the error sets that they produce as a proxy for the validation minority group. This procedure runs the risk of removing some of the few minority points present in the validation set. Therefore, instead of the actual intersection of the error sets, we use a "soft" version of the set intersection operator: the average of the error set accuracies over the ensemble. For an ensemble of K models, each producing different group annotations for the validation set $\bar{V}(\hat{f}_i)$ we can write the model selection criterion $\mathcal{C}(f_\theta, \{\bar{V}(\hat{f}_i)\}_{i=1}^K)$ as follows, for an arbitrary function f_θ with hyperparameters θ:

$$\mathcal{C}(\hat{f}_{t_1^*,\theta_1^*,t_2,\theta_2}, \{\bar{V}(\hat{f}_i)\}_{i=1}^K) = \frac{1}{K}\sum_{i=1}^{K} \frac{\sum_{(x,y,g)\in\bar{V}(\hat{f}_i)} \mathbb{1}[\hat{f}_{t_1^*,\theta_1^*,t_2,\theta_2}(x) = y | g = 1]}{|\{(x,y,g) \in \bar{V}(\hat{f}_i) : g = 1\}|} \tag{4}$$

Recall that the group label g is 1 for the samples that belong to the error set of the biased predictor trained in the first stage. By using this criterion we amplify the contribution of samples that are consistently predicted incorrectly by biased classifiers, while not discarding any of the samples in the error sets. In our experiments, we fix the ensemble size to 4, and show in Appendix G how the performance of the proposed approach varies when we use smaller ensembles.

4 Experiments

We now show empirical results obtained with our procedure and provide insight into the role played by our modifications to each of the stages.

Data sets. We highlight that one inconspicuous pitfall when developing new methodology is inadvertent overfitting, that is tailoring a method to particular data sets. In order to prevent this, we only use two data sets for designing and analyzing our approach (i.e. Corrupted MNIST and Waterbirds), and then run it on a number of other new data sets from various domains, ranging from tabular data (e.g. the Adult data set) to natural images (e.g. CelebA). We stress that we do not change our approach in any way in order to improve performance on these unseen data sets.

Baselines. There is no previous work that tackles the challenge of worst subgroup performance without using *any* subgroup information. Thus, we compare our method to baselines with different degree of required group label information: ERM w/o group labels, ERM tuned with a validation set with group annotations (ERM WG), JTT and GDRO. Detailed information on the tuning of the baselines can be found in Appendix D.

Main results. The results of our empirical analysis are summarized in Table 2. We point out that our method consistently outperforms the ERM baseline, the only other method that does not use any group annotations for training or model selection. Moreover, our approach often achieves a

worst-group accuracy comparable to JTT, which has the advantage of oracle group label knowledge for the validation data.

We note that with a group-annotated validation set even simple ERM training (ERM WG) can sometimes match the performance of the more sophisticated two-stage approaches like JTT.

Table 2: Average and worst-group test accuracy for methods that use varying degrees of group-annotated data.

		Corrupt-MNIST		Waterbirds		CelebA		Color MNIST		Adult		Poverty	
	Tuning	Avg	Wg	Avg	Wg	Avg	Wg	Avg	Wg	Avg	Wg	Avg	Wg
No group labels	ERM	99.6	71.2	97.9	74.9	94.3	60.7	99.8	82.6	80.1	41.6	87.6	**55.6**
	Ours	99.0	**96.5**	97.5	**78.5**	88.0	**78.9**	99.3	**96.6**	81.2	**68.0**	86.3	50.0
Val group labels	ERM WG	99.5	79.8	97.6	**86.7**	93.1	77.8	99.7	84.4	78.9	61.2	87.7	51.5
	JTT	99.1	**91.3**	93.3	**86.7**	88.0	**81.1**	98.3	**94.8**	77.8	**63.3**	64.5	**60.5**
Train & val group labels	GDRO	98.0	93.1	89.6	89.4	94.3	92.2	98.2	93.1	76.8	71.4	73.5	67.5

4.1 Ablation studies

In this section we break down our approach and give insights into why accurate model selection is possible even without any access to group labels.

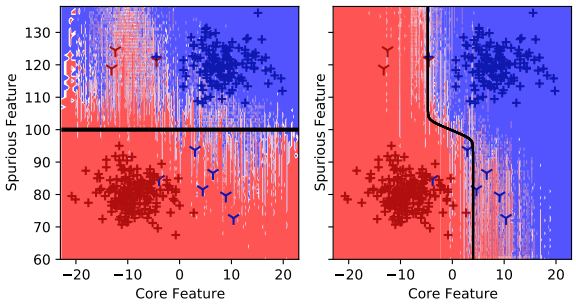

Figure 2: Prediction regions of a classifier (background) after one epoch (left) and at convergence (right). The solid black line shows shows a classifier that relies on the spurious feature for prediction (left) and the Bayes classifier (right).

Extreme regularization leads to biased classifiers. We begin by showing that neural networks trained according to our proposed first stage procedure are indeed biased towards making more mistakes on minority points. We train a LeNet5 model to perform even/odd digit binary classification on the Color MNIST dataset. We project the input space into a 2D representation where one of the coordinates correlates with the core features, and the other with the spurious features (see Appendix C for more details on how we obtained the projection). As shown in Figure 2, the classifier trained for one epoch and with hyperparameters tuned for average validation accuracy is more aligned with a predictor that relies exclusively on spurious features. Therefore, our first-stage classifiers will produce incorrect predictions primarily on the minority samples. Conversely, a classifier trained to convergence will resemble the Bayes optimal predictor that does not make mistakes uniformly over the minority groups, leading to error sets that contain fewer representative minority samples.

Precision and recall of detecting minority samples. We show in Appendix F the precision and recall of our first-stage predictors and how they compare to those of JTT. In particular, we note that our procedure for hyperparameter tuning achieves similar precision compared to JTT. On the other hand, JTT has the advantage of using oracle group labels on the validation set, and hence, obtains slightly better recall. We stress, however, that ultimately our goal is not to accurately detect minority points, but rather to improve prediction on minority subpopulations. As both JTT and our method show, an imperfect first-stage error set (i.e. with low precision and/or recall) can still significantly improve worst-group performance in the second stage on the data sets that we have considered.

Correlation between the $\mathcal{C}$ criterion and worst-group accuracy. We show in Appendix G that the criterion in Equation 4 correlates well with the actual metric that we are interested in, i.e. worst-group accuracy. Furthermore, we analyze the impact of using an ensemble of first-stage models when computing the proxy metric $\mathcal{C}$ and find that this procedure is necessary for selecting the optimal early-stopping time.

5 Discussion and future work

In summary, we introduce a model selection criterion that requires no group label information and show that it successfully achieves good worst-group performance on a variety of real-world data sets. Our current approach relies on a number of implicit assumptions. For instance, when selecting the error set in the first stage, data with large amounts of label noise could potentially render our method infeasible. We leave a more thorough investigation on ways to prevent the catastrophic effects of noise for future work. Moreover, we implicitly assume that we know which classes contain minority

groups, which helps especially with model selection in the second stage of our method. Relaxing this assumption is another interesting direction for future research.

References

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *ICML*, 2020.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning, 2021.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *ArXiv*, abs/2004.07780, 2020.

Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation, 2020.

Liangxiao Jiang and Chaoqun Li. Scaling up the accuracy of decision-tree classifiers: A naive-bayes combination. *J. Comput.*, 6(7):1325–1331, 2011.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9012–9020, 2019.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts, 2021.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning, 2020.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Yann LeCun et al. Lenet-5, convolutional neural networks.

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, 2021.

Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=jphnJNOwe36`.

Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. Automatic shortcut removal for self-supervised representation learning. In *ICML*, 2020.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier, 2020.

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, 2020a.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations, 2020b.

Nimit S. Sohoni, Jared A. Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems, 2020.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv: Machine Learning*, 2019.

John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric K. Oermann. Confounding variables can degrade generalization performance of radiological deep learning models. 2018.

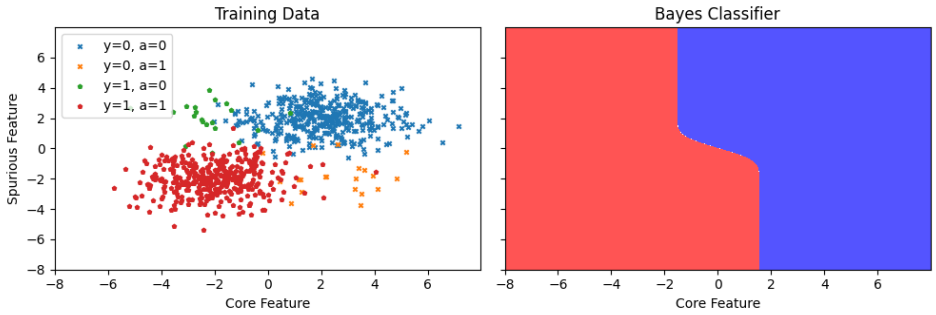

Figure 3: Synthetic data & Bayes classifier from the Gaussian mixture model with minority subgroups.

A Optimality in the presence of subgroups

In this section we conceptually and theoretically analyse a data model for the subgroup performance challenge. In this discussion we have two goals: First, analyse the fundamental performance limit of the Bayes classifier for certain data generating processes with rare subgroups. Second, analyse to what extend this fundamental limit is relevant for the behaviour of over-parameterized models and in that light contribute a different perspective to the question of memorization and feature learning raised in Sagawa et al. [2020b] and Menon et al. [2021]. To achieve these goals, we introduce and parameterize an explicit example of the data generating process in form of a Gaussian mixture model (GMM). For this GMM we demonstrate that the Bayes classifier has low subgroup performance and show that an over-parameterized random feature model learns a decision rule that is very similar to the Bayes classifier in the Gaussian mixture setting with rare subgroups.

B Explicit data model

Following the general description of the data generating process in 2 we define a concrete parametric example. We take the data generating model used in Sagawa et al. [2020b] to get sample data for experiments on implicit memorization. In this case each conditional distribution is a Gaussian with diagonal co-variance matrix. This "Gaussian mixture with minority subgroups" is defined as the following.

Definition B.1 (Gaussian Mixture with Minority Subgroups)**.** *The data of our Gaussian mixture with minority subgroups $x \in \mathbb{R}^d$ with $d \in 2\mathbb{Z}^+$ is distributed as a mixture of four Gaussian with each of them distributed according to $p_x(x|s,c) \sim \mathcal{N}(\mu_{s,c}, \Sigma)$. Of the d dimensions $d/2$ are spurious features $x_s \in \mathbb{R}^{d/2}$ and $d/2$ are core features $x_c \in \mathbb{R}^{d/2}$. The mean values $\mu_{s,c}$ are dependent on the core $c \in \{0,1\}$ and the spurious category ($s \in \{0,1\}$). For simplicity of calculation we assume symmetry around $\vec{0}$ with the centers $\mu_{0,0} = [\vec{k}, \vec{k}]$, $\mu_{1,1} = [-\vec{k}, -\vec{k}]$, $\mu_{1,0} = [-\vec{k}, \vec{k}]$, $\mu_{0,1} = [\vec{k}, -\vec{k}]$. The variance $\Sigma = \operatorname{diag} \vec{\sigma}_s, \vec{\sigma}_c$ is a diagonal matrix with the spurious variance $\sigma_s < \sigma_c$. The class conditional distribution are mixtures of two of the groups each:*

$$p(x|y=0) = p_x(x|a=1,c=0)p(a=1|c=0) + p_x(x|a=0,c=0)p(a=0|c=0) \quad (5)$$
$$p(x|y=1) = p_x(x|a=1,c=1)p(a=1|c=1) + p_x(x|a=0,c=1)p(a=0|c=1)$$

We define that each class label contains a minority group $0 < p(a=1|c=0) \ll p(a=0|c=0)$ and $0 < p(a=0|c=1) \ll p(a=1|c=1)$.

For this model it is simple to calculate the Bayes classifier and analyse its properties. Using Bayes rule we have

$$C^{\text{Bayes}}(x) = \arg\max_{y \in \{0,1\}} p(y|x) = \arg\max_{y \in \{0,1\}} p(x|y)p(y) \quad (6)$$

In this simple case of binary classification, the classifier can be rewritten as an inequality.

$$C^{\text{Bayes}}(x) = \begin{cases} \{0\} & \text{if } p(x|y=0)p(y=0) > p(x|y=1)p(y=1) \\ \{1\} & \text{if } p(x|y=0)p(y=0) < p(x|y=1)p(y=1) \\ \{0,1\} & \text{if } p(x|y=0)p(y=0) = p(x|y=1)p(y=1) \end{cases} \quad (7)$$

Table 3: Performance of overparameterized models in comparison to the Bayes Classifier in the Gaussian Mixture with minority subgroup dataset

	Bayes		Overparameterized	
Minority group ratio	Average	Worst	Average	Worst
0.5	0.843	0.836	0.798	0.766
0.7	0.858	0.693	0.814	0.645
0.9	0.928	0.445	0.905	0.425
0.99	0.991	0.076	0.989	0.027

The worst subgroup performance for this case transforms to:

$$\min_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim P_g} \left[C^{\text{Bayes}}(x) = y \right] = \min_{c \in \{0,1\}, a \in \{0,1\}} \int_{x \in \mathbb{R}^d} C^{\text{Bayes}}(x) p(x|c,a)\, dx \tag{8}$$

The integration cannot be solved analytically for Gaussian distributions, but numerical approximations are simple especially for low dimensions. In high dimensions one can use Monte Carlo simulations to get a good approximation of the integral. We compute numerical values for $d = 2$ through numerical double integration computation. In the first step we analyse the worst-group and average accuracy values that Sagawa et al. [2020b] observed for overparameterized models. While they attributed low worst-group performance to the overparameterized models, we hypothesize that the accuracy values are not a model phenomenon but correspond to properties of the underlying data distribution and the optimization objective. To have exact results we carry out the analysis for $d = 2$. The analysis is to change the relative probability of minority points in the range from $[0.5, 0.99]$ and change the variance σ_c. The resulting graph is displayed in (4). The general tendency aligns with Sagawa et al. [2020b] who described that increasing label spurious feature correlation leads to a sharp drop in worst subgroup performance of overparameterized models. Thus, the behaviour previously attributed to overparameterization properties arises for the Bayes classifier (compare 4). The same holds true for the variance analysis where the beavhiour of the overparameterized model and the Bayes classifier align. Furthermore, we analyse the average accuracy if only the core or spurious feature can be used for classification. Additionally, we compute numerical approximations through sampling for the exact settings from Sagawa et al. [2020b] with $d = 200$ and evaluate the overparameterized random feature model from their work (a logistic regression on a random feature model of size 3500) in terms of worst subgroup and average accuracy. We then compare it to the Bayes classifier. We find that the over-parameterized model almost attains the performance of the Bayes classifier in terms of worst subgroup as well as average accuracy (compare 3). Consequently, we conclude that worst subgroup performance can be a consequence of Bayes Error minimization on the data distribution.

The experiments and analysis demonstrate that in the synthetic setting of Sagawa et al. [2020b] the Bayes Classifier has low worst subgroup performance. In addition, they demonstrate that the low worst subgroup performance of overparameterized models is not due to problematic model behaviour but due to the underlying data distribution. If improving worst subgroup performance is the target, it is important to realize that there usually is a trade-off with average accuracy. Thus, the optimization target needs to be shifted away from the cross-entropy loss as a proxy for the Bayes Error to achieve good worst subgroup performance.

C Mapping images to spurious-core representation

In a general setting it is computationally infeasible to analyse distributional properties over image spaces. Especially, if the data is not generated with full control over the manifold. However, we utilize our assumptions on the underlying data distribution and limited control over the data generating process to get a 2D- representation of core and spurious features for some image datasets. This analysis is only feasible if core and spurious feature can be manipulated separately, this is the case for the worst subgroup datasets of Color-MNIST and waterbirds. To facilitate the analysis, it is necessary to generate a mapping from the image domain x to the core feature x_c and the spurious feature x_s which are assumed to be one dimensional. Given this mapping it is possible to estimate the distributions $p(x_c)$ and $p(x_s)$, visualize the Bayes Classifier, compute its accuracy and compare it

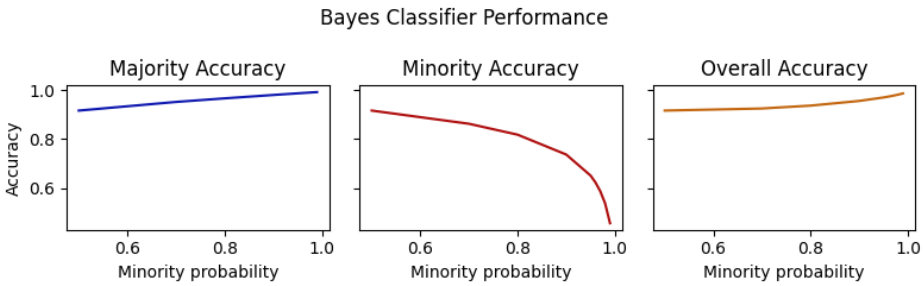

Figure 4: Bayes optimal classifier accuracy for different label spurious factor correlations. The minority accuracy is strongly affected by the correlation while the majority accuracy is less affected. The overall accuracy increases with an increase in label spurious feature correlation.

with the visualization and accuracies of CNN classification models. Furthermore, it is possible to get an intuition of the model behaviour in the first stage of our procedure.

C.1 Identifying Core and Spurious Feature Distribution

First, we train a mapping f from the core features to a one-dimensional representation. The closer f represents the ranking of the ground truth conditional distribution $p(y|x_c)$ meaning $p(y=1|x_c^1) > p(y=1|x_c^2) \implies f(x_c^1) > f(x_c^2)$ of the labels given the input, the better. In practice, we take the training images from MNIST, add a neutral background color to all images and train a LeNet5 LeCun et al. to minimize the cross-entropy loss on the odd / even classification task. For waterbirds, we take the cropped bird images and place them on a black background. We take the non-normalized output of the last layer to get the mapping

$$f : [0,1]^{\dim_1 \times \dim_2} \to \mathbb{R} \quad f(x) = x_c \tag{9}$$

Second, for Color-MNIST we define a bijection between the spurious feature color and a one-dimensional space by replacing the one color channel of the background with a value between 60 and 120.

$$g^{-1} : [60,120] \to [0,255]^3 \quad g^{-1}(\lambda) = [100, \lambda, 100] \tag{10}$$

For waterbirds we repeat the first step, but instead of having a classifier learning the core feature, this time the classifier learns the spurious feature. We take only the background images of water and land and train a classifier to classify land and water correctly with a mapping:

$$g : [0,1]^{\dim_1 \times \dim_2 \times 3} \to \mathbb{R} \quad g(x) = x_s \tag{11}$$

Third, we combine these two mappings and create a complete mapping of the input images to a 2d representation with the spurious feature and the core feature on different axis.

$$h : [0,1]^{\dim_1 \times \dim_2 \times 3} \to (\mathbb{R},\ [0,1]) \quad h(x) = \begin{pmatrix} f(x) \\ g(x) \end{pmatrix} = \begin{pmatrix} x_c \\ x_s \end{pmatrix} \tag{12}$$

Fourth, we create an artificial grid of images by combining the input factors. For Color-MNIST it is simple to add a different background color to the images from the validation set. For waterbirds we repeat the original data creation pipeline from Sagawa et al. [2020a] and place the birds in the centre of the landscapes of the background. We assume that x_c and x_s remain identical when we combine the inputs. For Color-MNIST this assumption is satisfied. For waterbirds some issues arise when the bird in the foreground covers important parts of the background, or the background itself contains birds or birdlike objects. However, this issue arises rarely.

C.2 Analysis of the Representation Quality

The analysis of the mapping can be conducted somewhat like the analysis of disentanglement features in VAE. However, here images are not created but sorted in a learned representation space. The first method is to create a traversal starting with images that have a very low expression of the core feature over ambiguous examples to very high expression of the core feature. We can visually

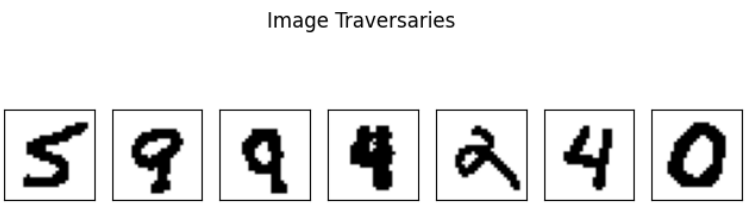

Figure 5: Image traversal for MNIST from clearly odd numbers to clearly even numbers with a ambiguous region in the middle

check if the hierarchy matches with the human perception. The second method is to check for model consistency. Namely, how strong do models learned with different initialization differ in their recovered distribution.

The traversal in 5 shows that the middle image is clearly ambiguous and the intended number could be even (4) or odd (9). And the numbers to the right and left of the centre image look less stereotypical and clearly written than the ones at the corners. We check three model random seeds and compute the pairwise Pearson correlation for the data points conditioned on the ground truth class label. This is important as for us it is not only relevant if the models learn a similar classification rule, but also if the feature expression within each class carries some ground truth. We find significant correlation of above 80%. The results for these consistency check appear promising and indicate that the learned mapping can indeed be used for qualitative and with some uncertainty even quantitative analysis.

D Hyperparameter Tuning

D.1 Baselines

For the ERM baselines we tune over learning rate, batch size, weight decay and if the dataset has class imbalance over reweighted and standard cross entropy loss. For the baselines tuned for average accuracy we select the hyperparameters and early stopping time with respect to average accuracy on the validation set. We then run three seeds of the best hyperparameter (7, 9, 15) and report accuracy values on the test set.

For the ERM baselines with worst-group early stopping we use the same tuning procedure. The only adaptation is that we select the model and early stopping time based on the best worst subgroup accuracy on a validation set. It is important to recognize that the high worst-group accuracy might not be consistent over runs with different seeds. In this setting high worst subgroup accuracy arises as a random deviation from the optimization target. Thus, hyperparameter search to a certain extent is the search over the best seed and parameter configuration. However, the good worst-group performance in the validation set generalizes to the test data, given sufficient minority samples in the validation set.

For the JTT baselines with worst-group early stopping we have a larger grid. In addition to the standard optimizer parameters, learning rate, batch size and weight decay there are the method specific selection epoch and upweighting factor. Selection epoch is the epoch from which training mistakes of the first stage are selected for upweighting in the second stage. The upweighting factor is the weighting that these samples receive during training of the second stage. We select the best model at best early stopping time based on worst subgroup performance on the validation set.

For the GDRO baselines with worst-group early stopping we perform a grid search that in addition to the standard optimizer parameters, learning rate, batch size and weight decay selects the best value for generalization adjustment. The generalization adjustment is an additional group specific upweighting factor for the loss that increases the importance of groups with few samples.

D.2 Ours

For our method hyperparameter tuning is carried out in two stages. In the first stage we fix the model class to models that can be trained from scratch with gradient descent in one epoch. The metric is overall accuracy on the validation set. The grid consists of different values for learning rate, batch size, optimizer, and weight decay. We carry out hyperparameter tuning in a twofold cross validation because the models in our final ensemble are also trained on half of the data and evaluated on the other half.

In the second stage we tune model selection and early stopping for our proxy metric. In addition to the standard parameters of learning rate, weight decay and batch size, we tune for the best upweighting factor for the train error set samples. To improve precision, we experiment with forming a hard intersection of two train error sets to select the samples to be upweighted. The proxy metric only selects this option for the poverty dataset that is in general very hard to optimize and might contain a lot of label noise in the minority groups.

For the results of our approach, we rerun the second stage with four different seeds (7,9,15,22) with the same hyperparameters and select the best early stopping epoch based on the proxy metric. The resulting standard deviations are in Table 4.

Table 4: Average and worst-group test accuracy for our method over four seeds for the second stage. Values in brackets are the standard deviation.

Corrupt-MNIST		Waterbirds		CelebA	
Avg	Wg	Avg	Wg	Avg	Wg
99.0 (0.5)	96.5 (1.0)	78.5 (5.8)	97.5 (0.1)	88.0 (2.6)	78.9 (2.9)
Color MNIST		Adult		Poverty	
Avg	Wg	Avg	Wg	Avg	Wg
99.5 (0.2)	96.5 (0.5)	81.2 (1.1)	68.0 (2.8)	50.0 (3.6)	86.2 (0.8)

E Label Noise/ No subgroups

By design our procedure assumes a minority group in each label category. If this assumption is violated, the error set for the class without minority subgroups will consist of samples that are not classified correctly by the "biased" classifier for other reasons than the misleading spurious feature. One possibility is the presence of samples with uniform label noise. In the smooth intersection proxy, the samples classified wrongly by all the ensemble models will dominate the metric. Optimizing for this metric will inevitably lead to poor average accuracy and hardly improvement on the worst subgroups and the proxy metric. The violation of the "minority in each class assumption" might be identified by a sharp decrease in average accuracy without much of an increase in the proxy metric. If this violation is detected in one class, the proxy metric should be limited to the other classes. If there are minority subgroups in the class uniform label noise is less of a problem. The error set will be a mixture of minority and label noise samples. The label noise samples are in general impossible to optimize for. Thus, the proxy metric will be affected by noise from these points, but it will still be correlated with the minority accuracy. The boosting and upweighting of the label noise samples during training seem to be even less of a problem. In CelebA, where we almost exclusively upweight label noise for the class of non-blond people the achieved overall and average accuracy are still high. In future research it would be interesting to transform these qualitative insights from CelebA to a quantitative insight on other datasets. Especially the hypothesis that the poverty dataset contains a large amount of label noise and how this influences the results, would be interesting to analyse. For this direction one can utilize synthetic datasets where the amount of label noise can be controlled.

F Precision Recall

Our procedure is targeted at achieving high majority accuracy in the first stage. Thus, avoiding many majority points to be present in the error set and achieving high precision. However, for some datasets one epoch is enough for a well optimized classifiers to pick up on both the core and the spurious signal. In this case, minority points are also classified correctly which results in a lower recall. In the development procedure we analysed using an ensemble of models to create error sets on the training data, but there was no clear improvement across datasets. Thus, we decided to keep the procedure as simple as possible.

		Class 0		Class 1	
		Precision	Recall	Precision	Recall
Poverty	Ours	0.57	0.09	0.49	0.72
	JTT	0.27	0.39	0.69	0.54
CelebA	Ours	-	-	0.16	0.38
	JTT	-	-	0.09	0.94
Waterbirds	Ours	0.23	0.80	0.16	0.70
	JTT	0.27	0.85	0.191	0.875

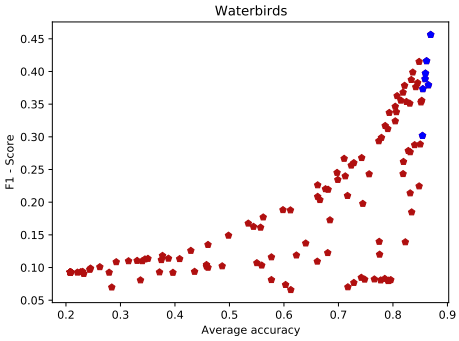

Figure 6: Waterbirds: F1 score for the identification of minority points with different first stage hyperparameters. The 5% highest overall accuracy runs are marked in blue.

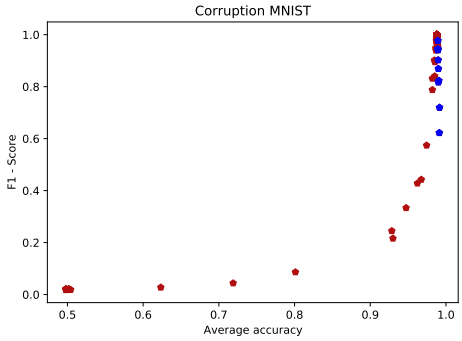

Figure 7: Corruption MNIST: F1 score for the identification of minority points with different first stage hyperparameters. The 5% highest overall accuracy runs are marked in blue.

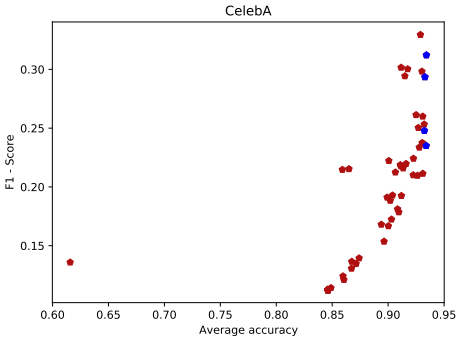

Figure 8: CelebA: F1 score for the identification of minority points with different first stage hyperparameters. The 5% highest overall accuracy runs are marked in blue.

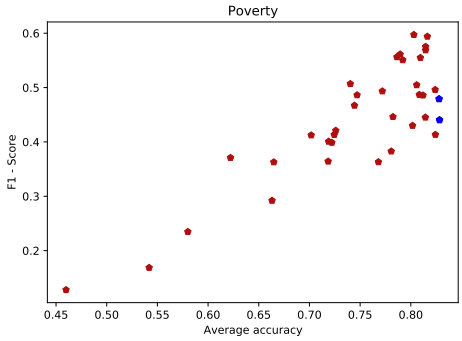

Figure 9: Poverty: F1 score for the identification of minority points with different first stage hyperparameters. The 5% highest overall accuracy runs are marked in blue.

G Proxy Metric

The goal of the proxy metric used for model selection in the second stage is to provide a reasonable surrogate for the worst-group accuracy. Note that a good model selection metric does not necessarily need to be a good estimate of the exact worst-group accuracy. Instead for hyperparameter selection it suffices if the two quantities have a similar ranking especially for large values. We split the analysis of the metric into two sections. First, we analyse the correlation across the whole range of parameters and epochs during the hyperparameter search. In this section small and large values of the proxy metric and the ground truth receive the same importance. Second, we analyse the selection of the early stopping time, given the selected optimizer parameters and thus zoom in on the important range of values.

G.1 Identification of hyperparameters

To analyse the hyperparameter selection performance we compute the rank correlation between the different metrics and the ground truth on the validation set. We use the worst-group validation accuracy as the ground truth. We consider models obtained for all the configurations in the hyperparameter search grid. Our proposed surrogate model selection criterion greatly outperforms the average validation accuracy criterion, as indicated in the table below. The number of first-stage biased models that are used for computing the criterion $\mathcal{C}$ introduced in Section 3.3 does not influence the rank correlation with the ground truth significantly. However, as we will see in the next section, the ensemble plays a role for selecting the regularization strength.

Table 5: Correlation between ground truth worst-group accuracy and different proxy metrics

Model selection criterion	Corrupt-MNIST	Waterbirds	CelebA	Color MNIST	Adult
Average val. acc.	0.59	0.93	-0.198	0.50	0.65
1 model	**0.72**	**0.94**	**0.90**	**0.99**	**0.81**
4-model ensemble	0.70	0.93	0.89	0.96	0.7

G.2 Early stopping time

The right early stopping time is crucial for the performance of the procedure. Given good optimization hyperparameters the model learns majority as well as minority samples well. Worst-group accuracy fluctuates relatively strongly and thus selecting an epoch with good worst-group accuracy is crucial. To compare the metrics on this task, we compute the validation accuracy if the best hyperparameter run is early stopped on either of the metrics. The displayed results are averaged over the four seed runs of the second stage. The result is that the epoch selection with our selected metric is more accurate across most datasets and the fluctuation between the seeds is smaller. For waterbirds this does not hold a possible reason is the extremely small amount of just 18 points in the smallest minority group in the validation set. Accordingly, the indirect up-weighting of the points that appear across validation error sets generally makes the selection procedure more stable.

Table 6: Average and worst-group validation accuracy for optimal hyperparameter models early stopped based on different metrics. Values in the brackets are standard errors over the four different seeds of the second stage.

Model selection criterion	Corrupt-MNIST		Waterbirds		CelebA		Color MNIST		Adult	
	Wg	Avg	Wg	Avg	Wg	Avg	Wg	Avg	Wg	Avg
Average val. acc	91.1 (7.4)	99.7 (0.0)	84.7 (2.1)	98.1 (0.1)	43.4 (23.9)	94.2 (0.6)	93.6 (1.5)	99.7 (0.0)	31.3 (8.3)	85.4 (0.1)
1 model	98.0 (0.8)	98.1 (0.8)	82.0 (6.6)	97.3 (0.3)	72.7 (9.6)	83.7 (4.8)	98.0 (0.6)	98.6 (0.6)	50.3 (7.4)	82.6 (2.0)
2-model ensemble	98.6 (0.5)	98.9 (0.5)	**85.9 (3.4)**	97.6 (0.3)	75.9 (5.1)	85.3 (3.4)	98.0 (0.6)	98.5 (0.5)	53.4 (2.9)	83.7 (0.4)
4-model ensemble	**98.9 (0.5)**	99.2 (0.4)	81.9 (3.5)	97.1 (0.2)	**77.1 (4.6)**	85.9 (3.1)	**98.2 (0.0)**	99.3 (0.4)	**62.2 (3.1)**	78.4 (2.7)

G.3 Effect of ensembling in the proxy metric

Next, we try to get an intuition why the ensembling is beneficial when selecting early stopping time. To do so we compare precision and recall for the points that appear only in one error set and those that appear in all four error sets (are in the intersection). The relative frequency of minority points in the

intersection is larger than in the individual error sets. This shows in a generally increased precision. However, the absolute frequency is also strongly reduced resulting in a lower recall. The agreement between error sets from different models is larger for minority points than for majority points. Thus, their importance is increased through the ensembling in our metric, which might explain the better results.

Table 7: Precision and recall of the individual error sets and their intersection

		Class 0		Class 1	
		Precision	Recall	Precision	Recall
Waterbirds	One error set	0.28	0.86	0.16	0.57
	Filtered error set	0.52	0.86	0.23	0.33
CelebA	One error set	-	-	0.18	0.50
	Filtered error set	-	-	0.26	0.31
Corrupt-MNIST	One error set	0.82	0.81	0.75	0.39
	Filtered error set	1	0.65	0.66	0.19
Color-MNIST	One error set	0.98	0.93	0.93	0.84
	Filtered error set	1	0.84	1	0.43

H Datasets

Waterbirds: Designed by Sagawa et al. [2020a] to analyse ERM behaviour with respect to spurious and core features the waterbirds dataset is a binary classification task of birds in to land birds and water birds. The dataset is synthetically created by extracting these birds from their original photos and placing them in front of water or land backgrounds. The spurious feature is the background with waterbirds in front of water in 95% of their occurrence and land birds in front of land in 95% of their occurrence. While the classifier must learn features related to coloring and form of the bird to correctly classify the images, the easy shortcut is to follow the color and structure of the background. The official publication comes with train, validation and test split and the experiments follow these official splits with one adaptation. The official validation and test sets are balanced with respect to subgroups for accurate estimation of worst subgroup performance. To analyse our approach under realistic conditions an i.i.d. validation set is necessary that was created by splitting the test set in half randomly. One half is used as a new test set. The other half is under sampled according to the training distribution to create this i.i.d. validation set. To ensure comparability no data augmentation is used, and the images are rescaled and cropped to 224 x 224 pixels. Afterwards, the images are normalized with the mean and standard deviation values of the ImageNet pretraining.

CelebA The CelebA dataset Dua and Graff [2017] is a collection of face images of celebrities annotated with a variety of attributes, face landmark points and many more. Following Sagawa et al. [2020a] we take the classification of hair color into blonde and non-blonde as an example for the worst subgroup question. The spurious feature in this situation is the annotated gender. In contrast to waterbirds there is only one rarely occurring subgroup: blond males. The easy feature to pick up is the gender (hair length) and the core feature hair colour is harder to learn for the classifier. To ensure comparability no data augmentation is used. The images are cropped to have a square format and then rescaled to the 224 x 224 pixels resolution. The last preprocessing step is a normalization with the image net mean and standard deviation values.

Poverty The poverty dataset Koh et al. [2021] is a collection of 8 channel satellite imagery from various regions in Africa with annotated information regarding country and if the picture is taken in an urban/ rural area. While the original task is to predict a continuous poverty index and generalize over country related domains, we create a new target and a new train, val, test split. We bin the continuous index into the classes "wealthy" and "poor" by taking 0 as a decision boundary and discarding images with a value between [-0.1,0.1] and try to classify images into these categories. Furthermore, we identify the urbanization label as spurious feature and analyse classification performance on the subgroups. The train, val, test splits are generated i.i.d. from all available data with 50% of the data used for training. No data augmentation and no transformation of the inputs is applied.

C-MNIST The corruption biased MNIST dataset was developed on the basis of the work in Goel et al. [2020] that combine the standard MNIST dataset LeCun and Cortes [2010] and the zigzag corruption of Corrupted MNIST. The classification task is to tell if a digit is even or odd and the spurious feature is having or not having a corruption. In comparison to the previous dataset we made a few adoptions: we reduced the fraction of subgroup samples from 5% to 1% to make the task more difficult and increased the overall amount of samples to demonstrate that the root for the problem is not to be found in too few samples. The official publication of MNIST and corruption MNIST has a train test split and we only use data from the training split for training and split the testing data into validation, i.i.d. validation and testing. No data augmentation and no transformation of the inputs is applied.

Color-MNIST The colour biased MNIST dataset is created based on the idea of Kim et al. [2019]. They color the MNIST letters with different colors while leaving the background unaffected. In our adaptation we add color to the background and pose the binary classification task of even and odd numbers. The colored background is created as a spurious feature with two base colors with some random color noise, with frequent color and even/odd combinations and rare once. In the experiments we vary the label, color correlation and sometimes use monocolor MNIST for demonstrative experiments. The official publication of MNIST has a train test split and we only use data from the training split for training and split the testing data into validation, i.i.d. validation and testing. No data augmentation and no transformation of the inputs are applied.

Adult The Adult dataset is taken from Jiang and Li [2011] and the pre-processing and subgroup definition was replicated from Lahoti et al. [2020]. The dataset is a tabular dataset extracted from a census in the USA. The binary prediction tasks is to estimate if the individuals yearly income is above or below 50.000 USD based on the given characteristics such as education and marital status. We define the gender and race to be protected attributes and define the following subgroups per class:

$$\mathcal{G} = \{\{\text{male,non-black}\}, \{\text{female,non-black}\}, \{\text{male,black}\}, \{\text{female,black}\}\}$$