ConceptDrift: Uncovering Biases through the Lens of Foundation Models

Cristian Daniel Păduraru
Bitdefender, Romania
University of Bucharest

Antonio Bărbălau
Bitdefender, Romania
University of Bucharest

Radu Filipescu
Bitdefender, Romania
University of Bucharest

Andrei Liviu Nicolicioiu
Mila, Montreal, Canada
University of Montreal, Canada

Elena Burceanu
Bitdefender, Romania
ILDS, Romania

{cpaduraru,ext-abarbalau,rfilipescu,eburceanu}@bitdefender.com,
andrei.nicolicioiu@mila.quebec

Abstract

Datasets and pre-trained models come with intrinsic biases. Most methods rely on spotting them by analyzing misclassified samples, in a semi-automated human-computer validation. In contrast, we propose **ConceptDrift**, a method that analyzes the weights of a linear probe, learned on top of a foundation model. We capitalize on the weight update trajectory, which starts from the embedding of the textual representation of the class, and proceeds to drift towards embeddings that disclose hidden biases. Different from prior work, with this approach we can pin-point unwanted correlations from a dataset, providing more than just possible explanations for the wrong predictions. We empirically prove the efficacy of our method, by significantly improving zero-shot performance with biased-augmented prompting. Our method is not bound to a single modality, and we experiment in this work with both image (Waterbirds, CelebA, Nico++) and text datasets (CivilComments).

1 Introduction

Deep neural networks, and especially fine-tuned versions of foundation models, are commonly deployed in critical areas such as healthcare, finance, and criminal justice, where biased predictions can have significant societal consequences [1]. Despite their impact, these models are often employed in their natural black-box state, i.e. as highly non-linear, multi-layered decision processes, lacking transparency or interpretability. Even if the pretrained model has been validated by the community, the dataset leveraged in the fine-tuning process can, and usually does, imprint the model with new biases. This issue is particularly concerning as biases from these datasets can lead to undesired outcomes [6], reinforcing existing inequalities or creating new forms of discrimination. This scenario finds its representation in subpopulation shift setups, where biases can naturally occur in samples.

Within the context of subpopulation shift setups, efforts employing foundation models [13, 37] have been recently made towards identifying and preventing biases. However, these methods limit themselves to data analysis alone. For instance, Kim et al. [13] focus on investigating misclassified validation samples. Their method relies on validating the presence of a given object within the set of mistakes and its absence from within the set of the correctly classified samples, in order to label it as a bias. The actual internal decision-making process of the model is never investigated nor referred to.

As an example, a method focusing on analyzing misclassified samples, such as B2T [13], is restricted to highlighting only the biases present in the validation set. Furthermore, some of the biases found

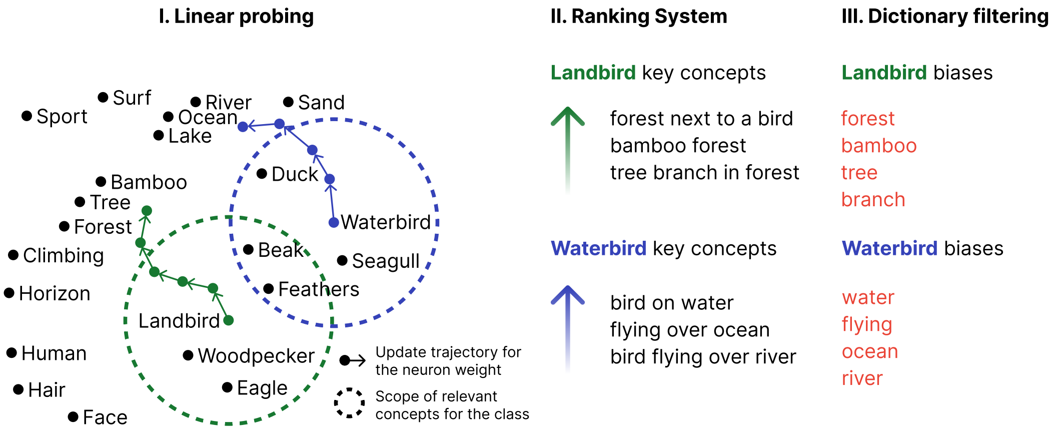

Figure 1: Illustration of ConceptDrift for the Waterbirds benchmark. The model's classification weights drift from the embedding of the textual representation of the class, outside the scope of relevant concepts, towards biases. We propose a novel embedding-space scoring system, capitalizing upon this drift, to identify which concepts factor in the final decision of the model and leverage a dictionary-based approach to delineate biases.

in the dataset might not have been imprinted upon the model weights. As an example, fine-tuning a ViT-L-14 CLIP [25] model on the O2O-Hard setup from Spawrious [19], a dataset specifically designed to instill biases at train-time and expose them at test-time, results in a 96% test-time accuracy. This demonstrates that biases in the data need not necessarily translate to biases in the model, and that model investigation is imperative in confirming whether or not a bias seen in the data is a contributing factor in the decision making process of the model.

We endeavor to expand upon the current usage of foundation models, beyond the restricted scope of simple data analysis, and propose a new direction for bias identification within the context of subpopulation shift setups. Our method focuses on investigating the skewness of the model's weights towards detecting and prioritizing spurious features as part of the decision-making process of the investigated model. We hereby propose a novel protocol for uncovering biases using foundation models such as CLIP [25] and mGTE [36], leveraging the topology of their embedding space to identify and name biases instilled by linear probing. Our protocol, dubbed ConceptDrift, is illustrated in Fig. 1. We showcase how, during training, the weights of the final classification layer drift away from the textual representation of their associated class, towards representations of spurious attributes. We propose a ranking system based on embedding-space arithmetic to extract keywords from concepts which factor in the activation of class neurons, and leverage a dictionary-based approach to delineate concepts outside the semantic scope of the classes, as biases.

We summarize our **main contributions** as follows:

1. We introduce **ConceptDrift**, a method capable to pin-point concepts relevant for the decision-making process of a model. We are the first to propose a weight-space approach for identifying the biases of fine-tuned foundation models, diverging from the current data-restricted protocols.

2. We propose a novel, embedding-space scoring method, able to reveal concepts which discriminatively impact the class prediction.

3. We show how our procedure is suited to assist in bias investigation. We reveal previously untapped biases on four datasets: Waterbirds [31], CelebA [17], Nico++[35] and CivilComments [5], showcasing significant improvements in terms of zero-shot bias prevention, upon state-of-the-art bias identification methods. Validated over image and text data, it can work on other modalities, with a foundation model with text processing capabilities as well.

2 Our Method

For a standard classification task $\{(x_j, y_j)\} \subset \mathcal{X} \times \mathcal{Y}$, we propose a method for pin-pointing concepts that are erroneously correlated to the task's classes. In order to achieve this, we train a linear layer on top of a frozen, pre-trained representations of the input data, obtained from a foundation model M. Next, we find concepts $(c_i)_{1 \leq i \leq q}$ in textual form, that are present in the training data and strongly influence the predictions of the classifier.

We require that the model M is capable of embedding both the concepts c_i and the input samples x_j into the $\mathbf{R}^D$ vector space, such that their cosine similarity $cos(M(c_i), M(x_j))$ is high when the concept represented by c_i is present in sample x_j.

The main steps of our method are the following:

Step 1: Initialization We initialize the weights w_k, $1 \leq k \leq |\mathcal{Y}| = N$, of the linear layer with the embedding of the corresponding class name, extracted by the model M, for each class k.

Step 2: Drifting towards biases, through learning We perform ERM [29] training on our dataset of interest, while keeping the weights w_k on the unit sphere. Through learning, the weights in the linear layer naturally shift from the original initialization, towards concepts that can effectively distinguish the samples of different classes. In an ideal, unbiased dataset w.r.t. the foundation model, the learned weights would be the embeddings of the class names. But in all the other cases, concepts used for classification drift, like visually presented in Fig. 1.

Step 3: Dataset concepts extraction For image classification task, we first use a captioning model to obtain descriptions of the images in the dataset. Next, for both image and text classification, we extract concepts from the captions or directly from the text samples.

Step 4: Rank the concepts For each class, we want to keep only the candidate concepts, which favour the prediction of that class with respect to another subset of classes. Since the weights w_k of each class are normalized, the prediction rule of the classifier can be formulated as:

$$\hat{y_j} = \arg\max_{k \in \mathcal{Y}} cos(w_k, M(x_j)). \tag{1}$$

This further motivates the need for w_k to point closer to samples in class k, than the weight of the other classes. Consider now a concept c_i, that has a high cosine with the weight w_k and a training example x_j containing the concept c_i. Based on the following inequality (proof in appendix A):

$$cos(M(x_j), w_k) \geq cos(M(x_j), M(c_i)) - \sqrt{2(1 - cos(w_k, M(c_i)))}, \tag{2}$$

it follows that: as long as our assumption from the beginning of this section holds, and w_k is highly similar with $M(c_i)$, then w_k is also guaranteed to have a high similarity with samples containing the concept c_i. Since we seek the concepts which favour the prediction of class k as opposed to at least one other class, we rank them by the difference in similarity of $M(c_i)$ with w_k, and the weight of any other class, w_p:

$$score_k(c_i) = cos(w_k, M(c_i)) - \underset{\substack{1 \leq p \leq N; \\ p \neq k}}{min} cos(w_p, M(c_i)) \tag{3}$$

Step 5: Filtering concepts Among the concepts with high rank, based on the score in Eq.3 we also expect to find those that refer to the class itself, or specific instances of it. We thus apply a filtering procedure to remove instances of the class from the keywords, leaving only associated attributes or keywords of completely different concepts.

3 Experimental analysis

Foundation models (FM) We used mGTE (gte-large-en-v1.5 [36]) for text embeddings in CivilComments [5], and OpenAI CLIP ViT-L/14 [25] for text and images in the other datasets.

We train the linear layer on L_2 normalized embeddings extracted by these models using the PyTorch [22] AdamW optimizer with a learning rate of $1e-4$, a weight decay of $1e-5$, a batch size of 1024 and a cosine annealing learning rate scheduler. We use the cross entropy loss with balanced class weights as the objective. The weights of the layer are normalized after each update and we

Table 1: Foundation Model (FM) Zero-shot prompting task. We modify the prompt using several bias-discovering methods, and evaluate the zero-shot performance of the FM. Notice how our ConceptDrift method significantly improves the accuracy for all datasets, over the baseline (prompt template w/o biases wildcard) and over the existing SoTA methods.

	Waterbirds		CelebA		Nico++		CivilComments	
	(Acc % ↑)		(Acc % ↑)		(Acc % ↑)		(Acc % ↑)	
Method	Worst	Avg.	Worst	Avg.	Worst	Avg.	Worst	Avg.
FM zero-shot [25]	35.2	90.7	72.8	87.4	57.7	88.4	33.1	83.4
FM w B2T [13]	48.1	86.1	72.8	88.0	-	-	-	-
FM w SpLiCE [4]	-	-	67.2	90.2	-	-	-	-
FM w Lg [37]	46.1	85.9	50.6	87.2	-	-	-	-
FM w **ConceptDrift** (ours)	**55.3**	84.7	**75.6**	88.4	**63.5**	86.2	**53.7**	69.0

Table 2: Model Ablation (Zero-shot prompting task). We variate the ranking score and the cut-off for concepts, revealing that both aspects could greatly influence the overall performance.

	Waterbirds		CelebA		Nico++		CivilComments		**Mean**	
	(Acc % ↑)		(Acc % ↑)		(Acc % ↑)		(Acc % ↑)		(Acc % ↑)	
Variations	Worst	Avg.	Worst	Avg.	Worst	Avg.	Worst	Avg.	Worst	Avg.
top-q concepts: 30	51.3	85.0	70.6	86.7	46.0	80.9	**54.0**	68.0	59.0	77.9
score: final - init weights	48.1	85.7	74.4	88.6	60.9	85.8	50.6	64.3	61.9	80.5
ConceptDrift * *top-q concepts*: 15 * *score*: classes difference	**55.3**	84.7	**75.6**	88.4	**63.5**	86.4	53.7	69.0	**65.8**	81.3

also learn a temperature to scale the logits. As early stopping criterion, we use the class-balanced accuracy on the validation set.

Keyword extraction For image captioning we use the GIT-Large model [32], trained on MSCOCO [15]. Next, to extract concepts we use YAKE [7], taking the top 256 n-gram concepts, for both $n = 3, 5$. For post-processing the selected concepts, we split them into individual words to remove stopwords, substrings from the class names, and hypernims or hyponims of the class concepts using WordNet [20] (*e.g.* 'seagull' for 'landbird' class). We remove keywords common for all classes, as they are usually in top because they are part of n-grams containing the class names.

3.1 Datasets

Waterbirds [31] is a common datasets for generalization and bias mitigation. It is created from CUB [33], by grouping different species of birds into two categories, *landbirds* and *waterbirds*, each being associated with a spurious correlation regarding its background, land and water respectively.

CelebA [17] is a large-scale collection of celebrity images (over 200000), widely used in computer vision research. The setup for using it in a generalization context [18] consists of using the *Blond_Hair* attribute as the class label and the *Male* attribute as the spurious variable.

Nico++ [35] image dataset has annotations for a main object and its context (*e.g.* dog on the beach). Unlike other datasets, NICO++ includes over 50 classes and 6 contexts, providing a richer context for evaluating model generalization performance across diverse scenarios. For this work, we build a setup with spurious correlations between 4 classes and 3 contexts (more details in Appx. A.2).

CivilComments [5] is a large collection (1.8 millions) online user comments, used also for researching bias and fairness in NLP, across different social and identity groups.

3.2 Quantitative analysis through zero-shot prompting

In this experiment, we validate the ability of our method to identify biases. We follow B2T [13] setup and choose the zero-shot prediction task. We augment the initial, class-only related prompt, with the

Table 3: Identified global biases. For a qualitative comparison, we show the biases extracted by multiple methods on Waterbirds and CelebA datasets. See in red biases that are off-topic, person names, or too related to the semantic content of the class, in green new biases, that were not identified before, and in blue words that come from expressions like 'body of water', which are quite difficult to filter. Notice how our ConceptDrift method proposes lots of new biases, that might be correct, since they are obtained by analysing the model weights drift while iterating through each dataset.

	Waterbirds (highest rank first)		**CelebA** (highest rank first)	
	landbird	*waterbird*	*blonde hair*	*non-blonde hair*
B2T [13]	forest, woods, tree, branch	ocean, beach, surfer, boat, dock, water, lake	model, favorite, outfit, hair, love, style	man, player, person, artist
SpLiCE [4]	-	-	hairstyles, dolly, turban, actress, tennis, beard	hairstyles, visor, amy, kate, fielder, cuff, rapper, cyclist
Lg [37]	forest, woods, rainforest, tree branch, tree	beach, lake, water, seagull, pond	woman blonde hair, blonde hair, actress, model, woman long hair	man, man wearing sunglasses, young man, black hair, actor
ConceptDrift (ours)	bamboo, log, tree, surrounded, floor, field, snowy,ground, forest	boat, lake, flying, ocean, pond, body, swimming	smiles, woman, long, girl, beautiful	man, dark, brown, eye, made, hat

bias, through a minimal intervention (*e.g.* 'a photo of a {cls} in the {bias}' (see Appx. A.1). For each class, we test one prompt for each bias identified in the dataset, taking into account the score for the best one (zero-shot with max over templates). The results in Tab. 1 show how the biases, automatically selected by our method, improve the worst group accuracy, over the initial zero-shot baseline and other bias-extracting solutions, in all four tested datasets. This emphasises on the quality of the biases automatically extracted by ConceptDrift. The better they are, the more capable the zero-shot prompt approach is to generalize, by adapting the prompt better to the new dataset context.

Ablations We validate key decisions in our algorithm in Tab. 2. We changed the ranking score (*score*) in Eq. 3 to the difference in cosine similarity of a concept with the final weights and the initial ones for each class. This highlights the concepts that the weights of a class have become more similar to, but does not take their similarity to other class weights into account. We also notice that the number of chosen concepts (*top-q concepts*) is important, as taking too many adds noise to the prompts and lowers the performance. We leave finding a good cut-off strategy for future work.

3.3 Extracting qualitative biases

In Fig. 2, we analyse the scores for the n-gram concepts on Waterbirds, for both classes, extracted as explained in Sec. 2). Notice how the score variation for each class is steep at the margins, becoming almost flat as soon as similarity decreases, showing that there are only a few candidates with high similarity scores, worth to be taken into account next for extracting the biases.

Qualitative examples We present in Tab. 3 the identified biases. Notice how our method comes with lots of new proposals for biases (in green). This might be case because our approach is fundamentally different, when compared with others [13, 37, 4], relying on the decision-making process of the model being investigated. See Nico++ and CivilComments in Appx. A.2.

4 Related Work

To enable a more meaningful comparison, we have distilled in Tab. 4 existing methods down to the aspects we consider fundamental to bias detection.

Biases and generalization Machine learning methods easily capture relevant factors to solve a task. Nevertheless, many times, models capture shortcuts [10], that are helpful in solving a task, but are not fundamental or essential for it. These shortcuts represent spurious correlations or biases, that don't always hold, and should not be used for reliable generalization outside of training distribution, often

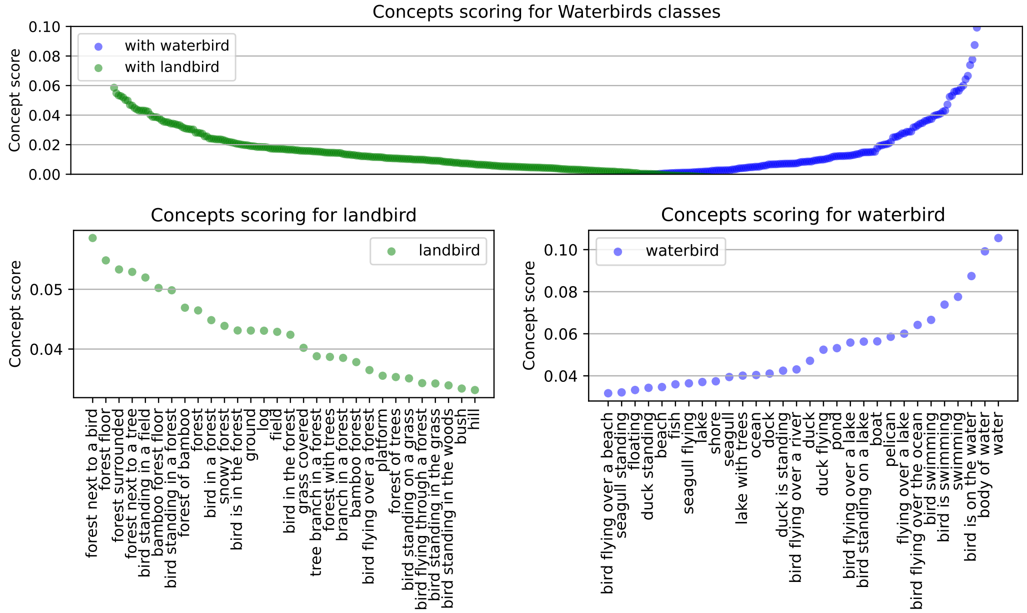

Figure 2: Top concept scores for Waterbirds (before Step 5: Filtering concepts). Notice that the curve has a steep descent on both ends, showing that there are just a few top candidates (with high scores), for each class. The first plot shows the similarities scores for all the concepts, while second and third plot are the detailed high score areas, one per Waterbirds class.

Table 4: Bias-extraction approaches comparison, based on fundamental differences in methods.

	Bias definition key focus	Source for bias candidates	Principle for scoring candidates
B2T [13]	mistakes driven	valid-set	common keywords in mistakes
SpLiCE [4]	dictionary learning	full dataset	Lasso solver
Lg [37]	class-specificity score	full dataset	embedding-space arithmetic
ConceptDrift (ours)	weights drift towards biases	full dataset	embedding-space arithmetic

leading to degraded performance [24, 3, 11]. Prior works [13, 37, 4] have thus focused on identifying dataset biases, through data analysis procedures.

Debiasing Debiasing and bias extraction techniques have become crucial in ensuring the fairness and accuracy of machine learning models [28], with extensive research dedicated to removing harmful biases across various domains. Some existing methods use bias annotations to train unbiased model, by means of group balanced subsampling [12], reweighting [27] or data augmentations [34]. In the absence of these annotations, other works [21, 16, 23] have proposed to first learn a biased model and then focus on its mistakes to train an unbiased one.

Fairness Fairness in machine learning has been extensively studied, with numerous approaches [8, 30] proposed to facilitate ethical research and ensure equitable outcomes across different subpopulations. Most of those methods overlap with domain generalization and worst-group performance improvements. This is also a field where model interpretability plays a crucial role [26], as understanding how decisions are made can help in identifying and mitigating biases.

Invariant Learning Robustness to out-of-distribution changes can be obtained by enforcing that the learning model is invariant to different environments or domains [2, 14, 34]. But there are many cases where we don't have access to such environments and we must discover them. Approaches like [9] partition the data into subsets that maximally contradict an invariant constraint, and apply algorithms for distributional robustness, like groupDRO [27], on those subsets, called environments.

5 Conclusions

We introduce **ConceptDrift**, the first method to identify biases using a weight-space approach, moving beyond traditional data-restricted protocols. Our novel embedding-space scoring method highlights concepts that significantly influence class predictions. We empirically demonstrate its effectiveness in bias investigation across four datasets: Waterbirds, CelebA, Nico++, and CivilComments, revealing previously undetected biases and achieving notable improvements in zero-shot bias prevention over current state-of-the-art methods. Validated on image and text datasets, with a foundation model also endowed with text processing capabilities, ConceptDrift can accommodate any other modality.

6 Acknowledgments

This work was funded by EU Horizon project ELIAS (No. 101120237).

References

[1] Angwin, Julia and Larson, Jeff and Mattu, Surya and Kirchner, Lauren. Machine Bias. There's software used across the country to predict future criminals. And it's biased against blacks. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2016. Accessed on: 2024-08-31.

[2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[3] S. Beery, G. Van Horn, and P. Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[4] U. Bhalla, A. Oesterling, S. Srinivas, F. P. Calmon, and H. Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice), 2024. URL `https://arxiv.org/abs/2402.10376`.

[5] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561, 2019. URL `http://arxiv.org/abs/1903.04561`.

[6] A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017. doi: 10.1126/science.aal4230. URL `https://www.science.org/doi/abs/10.1126/science.aal4230`.

[7] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2019.09.013. URL `https://www.sciencedirect.com/science/article/pii/S0020025519308588`.

[8] S. Caton and C. Haas. Fairness in machine learning: A survey. *ACM Comput. Surv.*, 56(7), apr 2024. ISSN 0360-0300. doi: 10.1145/3616865. URL `https://doi.org/10.1145/3616865`.

[9] E. Creager, J.-H. Jacobsen, and R. Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[10] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[11] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.

[12] P. Izmailov, P. Kirichenko, N. Gruver, and A. G. Wilson. On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*, 35:38516–38532, 2022.

[13] Y. Kim, S. Mo, M. Kim, K. Lee, J. Lee, and J. Shin. Discovering and mitigating visual biases through keyword explanation. In *CVPR*, 2024.

[14] D. Krueger, E. Caballero, J. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. L. Priol, and A. C. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *Proceedings of the 38th International Conference on Machine Learning, ICML*, 2021.

[15] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[16] E. Z. Liu, B. Haghgoo, A. S. Chen, A. Raghunathan, P. W. Koh, S. Sagawa, P. Liang, and C. Finn. Just train twice: Improving group robustness without training group information. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/liu21f.html`.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015. doi: 10.1109/ICCV.2015.425.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[19] A. Lynch, G. J.-S. Dovonon, J. Kaddour, and R. Silva. Spawrious: A benchmark for fine control of spurious correlation biases, 2023. URL `https://arxiv.org/abs/2303.05470`.

[20] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL `https://doi.org/10.1145/219717.219748`.

[21] J. Nam, H. Cha, S. Ahn, J. Lee, and J. Shin. Learning from failure: De-biasing classifier from biased classifier. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20673–20684. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/eddc3427c5d77843c2253f1e799fe933-Paper.pdf`.

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[23] M. Pezeshki, D. Bouchacourt, M. Ibrahim, N. Ballas, P. Vincent, and D. Lopez-Paz. Discovering environments with xrm. In *Forty-first International Conference on Machine Learning*, 2024.

[24] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 12 2008. ISBN 9780262255103. doi: 10.7551/mitpress/9780262170055.001.0001. URL `https://doi.org/10.7551/mitpress/9780262170055.001.0001`.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[26] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16, 01 2022. doi: 10.1214/21-SS133.

[27] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL `https://api.semanticscholar.org/CorpusID:213662188`.

[28] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars. *A Deeper Look at Dataset Bias*, pages 37–55. Springer International Publishing, Cham, 2017. ISBN 978-3-319-58347-1. doi: 10.1007/978-3-319-58347-1_2. URL `https://doi.org/10.1007/978-3-319-58347-1_2`.

[29] V. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5): 988–999, 1999. doi: 10.1109/72.788640.

[30] S. Verma and J. Rubin. Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness*, FairWare '18, page 1–7, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357463. doi: 10.1145/3194770.3194776. URL `https://doi.org/10.1145/3194770.3194776`.

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. J. Belongie. The caltech-ucsd birds-200-2011 dataset. `https://api.semanticscholar.org/CorpusID:16119123`, 2011. URL `https://api.semanticscholar.org/CorpusID:16119123`.

[32] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[33] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. 09 2010.

[34] H. Yao, Y. Wang, S. Li, L. Zhang, W. Liang, J. Zou, and C. Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR, 2022.

[35] X. Zhang, Y. He, R. Xu, H. Yu, Z. Shen, and P. 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.

[36] X. Zhang, Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang, M. Zhang, W. Li, and M. Zhang. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *CoRR*, abs/2407.19669, 2024. doi: 10.48550/ARXIV.2407.19669. URL `https://doi.org/10.48550/arXiv.2407.19669`.

[37] Z. Zhao, S. Kumano, and T. Yamasaki. Language-guided detection and mitigation of unknown dataset bias, 2024. URL `https://arxiv.org/abs/2406.02889`.

A Appendix

Finding biases in models We discuss so far how our approach can be used to find biases in datasets, but it can also be used for finding biases of a model w.r.t. the foundation model. We apply the same procedure, such that the new ground-truth labels of the dataset entries are the predictions of our model of interest.

Broader impacts We emphasize that our method should not be used in a stand-alone fashion for automated discovery of biases in every field and that human assistance is needed in order to interpret the model output before any further actions of consequence. Our tool is meant to aid and assist humans in the process of bias identification, not to replace them.

Limitations An important limitation in our method is the captioning model used for image classifications task. Zhao et al. [37] acknowledged as well that these models usually do not extract all the details in the images, so methods relying on them are limited to discovering the biases that they can extract. Another limitation is the keyword extracting procedure - using a more sophisticated one could bring forth new biases (*e.g.* extracting topics or taking into account synonymy). The method also relies on known hierarchies of concepts to detect biases by filtering concepts related to the desired class. These hierarchies and the relations they provide thus limit the type of filtering that we can ensure.

Bound on cosine similarity of vectors Let $u, v, t \in \mathbf{R}^D$ be three vectors of unit length, with u and v being fixed. We are interesting in finding the vector t that maximizes the difference in cosine similarity with the two fixed vectors:

$$\underset{\|t\|_2=1}{\arg\max}\ (t \cdot u - t \cdot v)\,,$$

where $\cdot$ represents the standard dot product of vectors. This can be rewritten as:

$$\begin{aligned}\underset{\|t\|_2=1}{\arg\max}\ t \cdot (u - v) &= \underset{\|t\|_2=1}{\arg\max}\ cos(t, u - v)\,\|u - v\|_2 \\ &= \underset{\|t\|_2=1}{\arg\max}\ cos(t, u - v)\,,\end{aligned}$$

as $\|u - v\|_2$ is a constant. It is now easy to see that the solution to this problem is $t = \frac{1}{\|u-v\|_2}(u - v)$, the unit length vector with the same orientation as $u - v$. Using this we can place an upper bound on the initial difference:

$$t \cdot u - t \cdot v \leq \|u - v\|_2\ ,$$

which we then rearrange as

$$t \cdot v \geq t \cdot u - \|u - v\|_2\ .$$

The norm $\|u - v\|_2$ can be equivalently expressed as

$$\|u - v\|_2 = \sqrt{(u - v) \cdot (u - v)} = \sqrt{2 - 2u \cdot v} = \sqrt{2(1 - u \cdot v)}\,.$$

Introducing this in the previous inequality we obtain

$$t \cdot v \geq t \cdot u - \sqrt{2(1 - u \cdot v)}\,.$$

Since u, v and t are vectors of unit length we can replace the dot products with the cosine similarity. By then setting $u = M(c_i)$, $v = w_k$ and $t = M(x_j)$ we finally obtain the inequality:

$$cos(M(x_j), w_k) \geq cos(M(x_j), M(c_i)) - \sqrt{2(1 - cos(M(c_i), w_k))}$$

A.1 Zero-Shot Prompts

The basic prompts we used for each dataset are the following:

- Waterbirds: 'a photo of a {class name}'
- CelebA: 'a photo of a person with {class name}',
- CivilComments: '{class name}'
- Nico++: 'a photo of a {class name}'.

Next, we change them to accomodate the biases wildcard:

- Waterbirds: 'a photo of a {class name} in the {bias}'

Table 5: Identified global biases - Nico++
We find words related to the environments that we associated to each class, but also some attributes more specific to the class itself than the other ones (*e.g*. 'wooden' for chair).

	Nico++ (highest rank first)			
Model	car	flower	chair	truck
Ground Truth Biases	outdoor	grass	water	water
ConceptDrift (ours)	beach, parking, standing, driving, parked, blue, road, pool, lot, group	red, close, yellow, wild, field, floating, water, white	sitting, red, pool, beach, wooden, floating, near	driving, road, large, lake, black, beach, spraying, field, standing, water

Table 6: Identified global biases - CivilComments++
Notice how references to religion and ethnicity are common in the class of offensive comments, while in the opposite part we have words that are more common in formal contexts.

	CivilComments (highest rank first)	
Model	non-offensive	offensive
ConceptDrift (ours)	experienced, completely, responsible, barrier, coverage, attempt, Engineer, total, Notice, shared, primarily, regard, helping, accepting, paycheck, wrote, petition, case, always, aspects, rest, noticed, name, hours, analysis, Extension, personal, blog, based, relative, important, new, mentioned	losers, acting, bigotry, misogynist, mental, racist, Muslim, jesus, Christian, driving, Sexuality, White, supremacist, Trump, someone, rid, repub, president, white, Mental, lesbian, like, people, Jihadist, intellectuals, state, God, dangerous, black, mans, killing, ultimate

- CelebA: 'a photo of a {bias} with {class name}'
- CivilComments: 'a/an {class name} comment about {bias}'
- Nico++: 'a photo of a {class name} in the {bias}'.

The class names used in the templates and for the initialization of the linear layer weights are:

- Waterbirds: 'landbird', 'waterbird'
- CelebA: 'non-blonde hair', 'blonde hair'
- CivilComments: 'non-offensive', 'offensive'
- Nico++: 'car', 'flower', 'chair', 'truck'

A.2 Nico++ and CivilComments Biases

See Tab. 5 and Tab. 6 for the biases extracted with our method for Nico++ and CivilComments datasets.

Custom Nico++ subset For the experiments on Nico++ we selected only the first four classes and paired them with the environments that they had the most samples in, resulting in the following associations: (car, outdoor), (flower, grass), (chair, water), (truck, water). Notice how the classes chair and truck shared the same bias, in contrast to most popular subpopulation shift datasets that only have one-to-one associations of classes and biases. For the training set we keep for each class 300 samples from its associated environment and only 25 from the other ones, while for validation we keep 50 from the associated one and 25 from the others. The test set is made up of all the remaining samples.