CLASSIFIER-TO-BIAS: Toward Unsupervised Automatic Bias Detection for Visual Classifiers

Quentin Guimard[1] Moreno D'Incà[1] Massimiliano Mancini[1] Elisa Ricci[1,2]
[1]University of Trento [2]Fondazione Bruno Kessler
https://github.com/mardgui/C2B

Abstract

A person downloading a pre-trained model from the web should be aware of its biases. Existing approaches for bias identification rely on datasets containing labels for the task of interest, something that a non-expert may not have access to, or may not have the necessary resources to collect: this greatly limits the number of tasks where model biases can be identified. In this work, we present CLASSIFIER-TO-BIAS *(*C2B*), the first bias discovery framework that works without access to any labeled data: it only relies on a textual description of the classification task to identify biases in the target classification model. This description is fed to a large language model to generate bias proposals and corresponding captions depicting biases together with task-specific target labels. A retrieval model collects images for those captions, which are then used to assess the accuracy of the model w.r.t. the given biases.* C2B *is training-free, does not require any annotations, has no constraints on the list of biases, and can be applied to any pre-trained model on any classification task. Experiments on two publicly available datasets show that* C2B *discovers biases beyond those of the original datasets and outperforms a recent state-of-the-art bias detection baseline that relies on task-specific annotations, being a promising first step toward addressing task-agnostic unsupervised bias detection.*

1. Introduction

The wide availability of pre-trained machine learning models allows anyone to access them. A user may search in online libraries (*e.g.*, HuggingFace [27]) for models addressing their task of interest, download the most suitable one, and use it for their specific use case. As the user did not develop the model, they may not be aware of potential failure modes [28, 60, 63] or harmful biases [9, 10, 24, 49, 69, 70]. These are usually reported on model cards [47], describing how the model has been developed, in which cases it has been tested, its recommended use, and its potential biases.

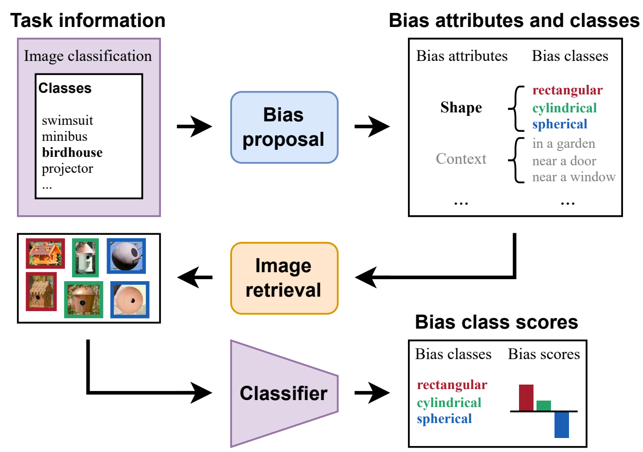

Figure 1. We explore the novel task of bias discovery when we are only given a specific classification task and a pre-trained model. We propose **CLASSIFIER-TO-BIAS** (**C2B**), which automatically detects potential biases in the model, categorizes these biases, and assigns scores to each category.

While model cards are a useful tool to describe a model and its shortcomings, (i) a lot of models still come without them; (ii) even if they are present, they may not describe specific cases or biases of interest to the users. This means that a user would need to assess the weaknesses of the downloaded model independently.

Various tools have already been proposed to measure and mitigate biases in datasets [8, 64] and models [2, 8, 11]. While these tools rely on bias labels, recent work has made progress toward open-set failure mode extraction [16, 19, 22, 30, 53] and bias detection [6, 17, 33, 34, 68] without bias labels, even removing the need for a predefined list of biases. However, in the context of discriminative models such as visual classifiers, all existing methods rely on task-specific labeled data to be able to identify failure modes and measure the biases, limiting the number of tasks where model biases can be identified, and preventing the automation of bias detection.

To address this problem, we present CLASSIFIER-TO-

BIAS (C2B), the first framework that, given a pre-trained visual classifier and its task specification, automatically identifies its biases (Fig. 1) in an unsupervised manner, removing the need for task-specific labeled data. We build C2B on two components: a large language model (LLM) and a text-to-image retrieval engine. First, the task specifications are fed to the LLM to produce a series of candidate *bias attributes* for the task (*e.g.*, gender, ethnicity), and their corresponding *bias classes* (*e.g.*, male, female). The list of candidate bias classes is fed again to the LLM to generate a set of captions describing both the biases and the task-specific target classes they are associated to. These captions are used by the retrieval module to get images from a large database: the images will constitute our dataset, where, for each image, both the target and bias labels are given by the constructed caption itself. By testing the pre-trained model on this dataset, we can compute the standard bias metrics, verifying which bias class the model is most biased against.

Importantly, C2B is fully unsupervised and does not suffer from the need of costly annotations, thus it can be applied to automatically detect biases for virtually any classification task. Moreover, the list of biases is class-specific, as different classification tasks and categories (*e.g.*, birds, deers) may exhibit different biases (*e.g.*, tail color, horns size). Finally, the model is modular and can readily incorporate more powerful LLMs, retrieval approaches, or databases as they become available. We test our framework on two datasets, *i.e.*, CelebA [41], and ImageNet-X [28]. Results indicate that C2B can uncover biases beyond those annotated in the original datasets, outperforming a recent supervised open-set bias detector [33].

Contributions. To summarize, our contribution is three-fold: (i) We propose the new task of unsupervised bias detection in a pre-trained model given *only* a textual description of the classification task *without* requiring any annotated dataset or manual intervention, and define a protocol to evaluate methods on this setting; (ii) We present C2B, the first framework for this task that uses an LLM to propose a list of biases and generate captions, and multimodal retrieval to produce a dataset from which biases can be estimated; (iii) We show that C2B can discover biases beyond those annotated in the original datasets, being the first step toward addressing this challenging problem and democratizing bias discovery.

2. Related Work

Bias Detection and Mitigation. Bias detection and mitigation are essential to ensure fairness and robustness in machine learning models. Several approaches tackle hidden subgroup imbalances without relying on explicit group annotations [35, 37, 59, 61], *e.g.*, by analyzing the mistakes on annotated task data and re-weighting misclassified samples [37]. Other methods focus on open-set or annotation-free bias discovery [34, 36, 68]. UDIS [34] uses hierarchical clustering to surface systematic biases without predefined subgroup definitions. DebiAN [36] alternates training between a bias discoverer and a classifier to identify and mitigate multiple biases. DIM [68] decomposes feature spaces using Partial Least Squares to uncover multiple unknown biased subgroups and describes them using vision-language models. More recently, language-driven methods have also gained prominence, using captions and keyword extraction to detect and name potential biases learned by models [14, 33, 71]. B2T [33] and Zhao et al. [71] relies on captions derived from annotated datasets or misclassified samples with known labels, validating these biases using, *e.g.* vision-language similarity. SaMyNa [14] emphasizes interpretability by extracting semantic bias descriptors from large-scale captioning, supporting bias analysis and debiasing. However, all these methods require access to annotated, task-specific datasets.

In contrast, our approach, C2B, operates without any kind of annotated data. It relies solely on a natural language description of the classification task and a pre-trained classifier. Using LLM-generated, class-specific bias proposals combined with retrieval-based evaluation, C2B enables fully unsupervised, task-agnostic bias discovery.

Other related methods focus on bias detection or fairness in generative models [17, 32, 57]. For instance, Open-Bias [17] uses LLMs and VQA models to detect and quantify open-set biases in text-to-image generative models, while GELDA [32] reveals visual biases in image generators by generating domain-specific attributes with LLMs and annotating them using selected VLMs.

Inspired by these works, we choose to leverage an LLM as well as a text-to-image retrieval engine to tackle the new task of unsupervised bias detection, where only a pre-trained model and a textual description of the classification task are available, thus eliminating the need for annotated datasets or manual intervention.

Explainability and Failure Mode Detection. A parallel research direction focuses on systematic failure mode discovery and interpretability [3, 4, 16, 25, 29]. VoG [3] highlights difficult examples by analyzing gradient variance, and Spotlight [16] identifies contiguous underperforming regions in representation space. Saliency-based methods [29] provide local visual explanations, while counterfactual explanations are generated in either natural language [25] or visual form using diffusion models [4].

Recent work has scaled these efforts using embeddings and generative models. Wiles et al. [65] leverages large-scale generative models to automatically find and describe failure clusters. Domino [19] uses cross-modal embeddings and mixture models to identify and describe coherent data slices where models underperform. Jain et al. [30] extracts latent directions corresponding to consistent model errors and generates challenging counterfactual examples for data augmentation. AdaVision [22] enables interactive discovery of failure modes by combining CLIP-based retrieval with GPT-generated prompts. PRIME [53] prioritizes interpretability by starting from human-defined tags and identifying minimal descriptive combinations that explain failures. MAIA [56] automates interpretability workflows using vision-language models for feature analysis and failure mode discovery. Csurka et al. [15] formalizes language-based error explainability and proposes evaluation metrics to assess the quality of generated explanations.

While these methods analyze model behavior and failure patterns through model outputs or datasets, our work introduces a complementary approach: starting from a high-level task description, C2B uses LLM-based bias proposals and retrieval-based validation to automatically identify, validate, and quantify biases in pre-trained classifiers, enabling large-scale, unsupervised bias discovery across arbitrary classification tasks.

3. Method

In this section, we first define bias attributes and bias classes (Sec. 3.1), and formulate the problem of identifying biases of pre-trained models from their task specifications (Sec. 3.2). We then describe C2B, that proposes a list of biases (Sec. 3.3) and collects data for testing them (Sec. 3.4). Finally, we describe how we measure model's biases from the collected data (Sec. 3.5).

3.1. Bias definition

Formally, we denote the target classification model as $f : \mathcal{X} \to \mathcal{Y}$, mapping an image in the space $\mathcal{X}$ to labels in the space $\mathcal{Y}$. We define as **bias attributes** $\mathcal{B} = \{\mathbf{B}_1, \cdots, \mathbf{B}_N\}$ the different categories of bias, *i.e.*, factors that can impact its performance [44]. For each **bias attribute**, we define a list of **bias classes** $\mathbf{B}_i = \{b_{i,1}, \cdots, b_{i,n_i}\}$, the different values that each bias attribute can take. For a given target class, we define the model to be **biased** or unfair w.r.t specific bias classes depending on the per-class accuracy difference of the model. Let us denote the per-class accuracy of a model f on a target class y on data that matches a bias class $b_{i,j}$ as $A_y(f, b_{i,j})$. We define the bias score $\phi_{y,i,j}$ for each bias class $b_{i,j}$ as

$$\phi_{y,i,j} = A_y(f, b_{i,j}) - \frac{1}{n_i - 1} \sum_{b_{i,k} \in \mathbf{B}_i, k \neq j} A_y(f, b_{i,k}). \quad (1)$$

3.2. Problem formulation

Given a pre-trained model f, our goal is to identify its biases. We assume f to be a generic classification model that we did not develop and that we aim to use (*e.g.*, downloaded from the web). Thus, we also assume to have no access to additional data f had access to (*e.g.*, the dataset set used for training). As we want to avoid costly annotation procedures, we also assume to have no access to any dataset containing labeled images of the classes in $\mathcal{Y}$.

To address this problem, we can rely only on what we know about the model: *i.e.*, the task it has been designed for, the input domain it expects, and the output it predicts. This information can all be expressed using simple, natural language. Specifically, we assume to have a general description of the model's task T, the input the model expects to process, and its output (*i.e.*, classes in $\mathcal{Y}$).

This task is challenging as (i) detecting biases requires measuring the performance of the model on a labeled dataset, potentially containing annotations for the factors of interest and (ii) we have no pre-defined list of biases to start with. This implies that we have to both generate a list of potential biases and to collect a dataset to test them. In the following, we will describe our framework, C2B, that exploits LLMs for proposing biases, and a combination of LLMs and multimodal retrieval for scoring them. An overview of C2B is illustrated in Fig. 2.

3.3. Generating bias proposals

The first step of C2B is to generate a list of bias attributes and classes that we can explicitly test for. To generate the list, we cannot rely on attribute annotations, as we have no access to labeled data. Moreover, we cannot use lists available online, as biases depend on the task of interest.

To address this problem, we exploit the reasoning capabilities of LLMs. Specifically, let us denote as Φ_{LLM} a generic LLM, mapping textual inputs to textual outputs, *i.e.*, $\Phi_{\text{LLM}} : \mathcal{T} \to \mathcal{T}$, with $\mathcal{T}$ being the space of natural language. To generate the candidate set of biases, we prompt the LLM to produce a list of bias attributes and classes for the given model and task description. Because different classes may have different biases, we query the LLM to give us a list of potential biases $\mathcal{B}^y$ for each class $y \in \mathcal{Y}$. Formally, we have:

$$\mathcal{B}^y = \Phi_{\text{LLM}}(\pi_{\text{bias}} \circ T \circ y) \quad (2)$$

where $\circ$ is the string concatenation operation, π_{bias} is the system prompt that instructs the LLM about the task it should perform (*Supp. Mat.*, Appx. I), and T is the model's task description, including input/output specification.

The output of Eq. (2), $\mathcal{B}^y$, is a list[1] of bias attributes $\mathcal{B}^y = \{\mathbf{B}_1^y, \cdots, \mathbf{B}_{N_y}^y\}$, of size N_y. We prompt the LLM so that each bias attribute is associated to a list of $n_{y,i}$ bias classes $\mathbf{B}_i^y = \{b_{i,1}^y, \cdots, b_{i,n_{y,i}}^y\}$.

[1]For simplicity, we assume that the natural language output of the LLM is already formatted as a list.

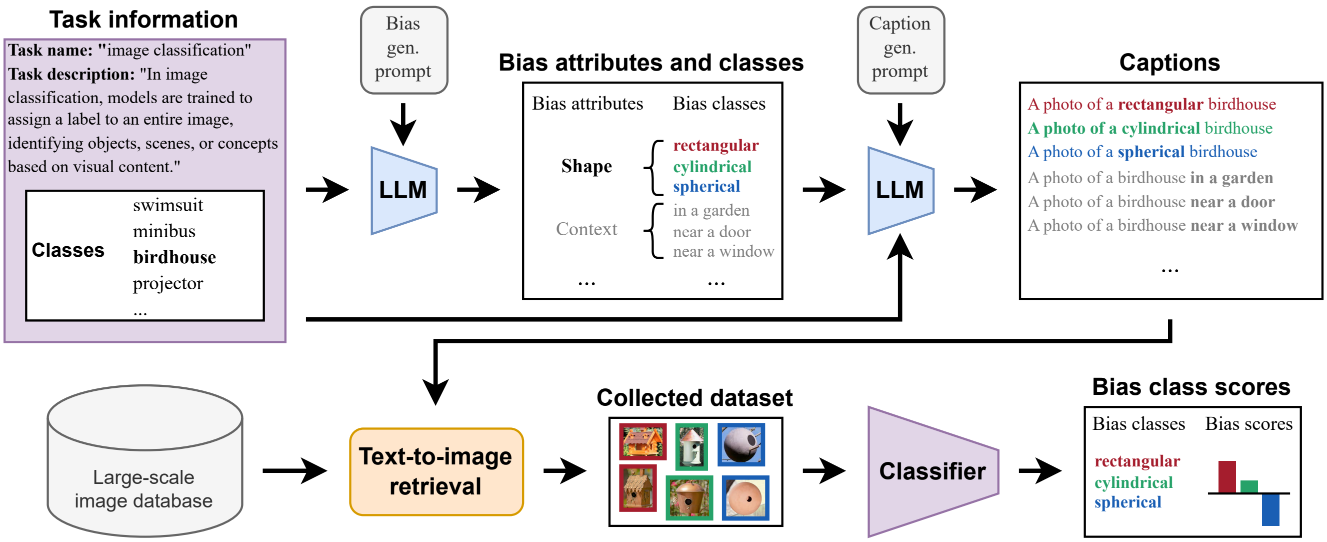

Figure 2. **Overview of C2B**. Given a specific task with an associated description and a pre-trained classification model, our approach leverages an LLM to identify potential candidate bias attributes and corresponding classes. These candidate biases are then used to prompt the LLM to generate captions, which are subsequently utilized by a retrieval module to collect a dataset of images for bias testing. This dataset enables evaluation of the target classification model, producing scores for each identified bias category.

3.4. Collecting data

Given the list of candidate bias attributes and classes, we still need a dataset to score them. Ideally, for each target class $y \in \mathcal{Y}$ and bias class $b^y \in \mathbf{B}^y$, with $\mathbf{B}^y \in \mathcal{B}^y$, we would like to have a dataset $D(b^y, y)$ depicting images of class y exhibiting the bias class b^y, *i.e.*, $D(b^y, y) = \{(x_i, y_i, b_i^y)\}_{i=1}^M$ of size M. Unfortunately, we cannot rely on labeled data for this task as (i) we do not have access to it; (ii) biases change w.r.t. the particular application, thus it is very hard to collect beforehand a dataset containing all possible target and bias classes of interest.

To address this problem, we resort to external sources. In particular, we assume to have access to a large-scale database $\mathcal{D}$ containing images. Note that $\mathcal{D}$ has no annotations: we just assume it to be large-scale (*i.e.*, potentially web-scale). From this database, we can collect images for each target-bias combination of interest using a text-to-image retrieval model. This is achieved in two steps: captions generation and cross-modal retrieval.

Caption generation. A straightforward solution to create captions is to use a fixed template (*e.g.*, `an image of [target class] with [bias class]`). However, this would not consider the specificity of the considered task and thus may lead to retrieving erroneous images. For instance, the biases and captions associated to a facial attribute classification task should be different from those in a natural scene classification task. To avoid this issue, we resort to using the LLM, querying it in two steps: first to create a template for the specific task, and then to generate captions specific for target and bias class. Formally, we obtain the initial template as:

$$\texttt{temp} = \Phi_{\text{LLM}}(\pi_{\text{template}} \circ T) \tag{3}$$

where π_{template} is the system prompt that instructs the LLM about the task it should perform (*Supp. Mat.*, Appx. I). Then, we ask the LLM to refine this template for each specific target class y and bias class b, as follows:

$$c_b^y = \Phi_{\text{LLM}}(\pi_{\text{caption}} \circ \texttt{temp} \circ b \circ y). \tag{4}$$

where c_b^y is the resulting caption. Note that, while with this strategy we obtain one caption per target-bias class pair, in principle we can also generate multiple captions per pair to have more variability, beneficial for more complex tasks.

Image retrieval. The obtained captions can be used to retrieve images from a large scale image database, and we implement the retrieval by using a contrastive vision-language model (VLM) that can score the similarity between a text and image. Formally, we can decompose the VLM in three elements: a visual encoder $\varphi_{\texttt{vis}} : \mathcal{X} \rightarrow \mathbb{R}^d$, a text encoder $\varphi_{\texttt{txt}} : \mathcal{T} \rightarrow \mathbb{R}^d$, and a similarity function $\texttt{sim} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ (*e.g.*, cosine similarity for CLIP [51]). Given a caption, we can retrieve a set of images $\mathbf{X}_b^y$ for each class label y and bias b as follows:

$$\mathbf{X}_b^y = \underset{x \in \mathcal{D}}{\text{top-k}} \ \texttt{sim}\left(\varphi_{\texttt{vis}}(x), \varphi_{\texttt{txt}}(c_b^y)\right), \tag{5}$$

where we retrieve the k images with the highest matching score. While these images do not have a ground-truth, they can still help to estimate the biases of the model, with the fidelity of the latter depending on the retrieval accuracy (Sec. 4.4). We also experiment with a different type of retrieval system, a Web search engine (details in Sec. 4.1).

3.5. Measuring biases

In the previous section, we have collected a dataset with pseudo-annotations for all bias values and classes of interest. We can now test the target model on this dataset to compute its performance, using the variation across bias classes to score each bias. We use the formula defined in Eq. (1) to score each target-bias pair relative to other target-bias pairs. For each bias class, we obtain a bias score between -1 and 1. If the per-class accuracy is higher for a given bias class, the classifier is biased *toward* the bias class (positive bias score). Reciprocally, if the per-class accuracy if lower for a given bias class, the classifier is biased *against* the bias class (negative bias score). Intuitively, larger absolute bias score values also correspond to stronger biases.

To quantitatively evaluate the quality of the biases detected by C2B, we propose two novel evaluation settings that complement each other. First, in Sec. 4.2.1, we propose to match the detected biases with ground-truth annotations using SBERT embeddings for tasks where datasets with additional bias annotations exist. Second, in Sec. 4.2.2, we propose to assign bias labels (based on the initial proposed biases) to ground-truth task-specific data with a VQA model, to check the agreement between the bias detection method and an expert VQA model.

Collecting scores for all bias attributes and classes enables various types of analyses to explore the biases. Biases can be ranked by score, to assess which attributes are more critical than others. Bias class scores can also be aggregated across bias attributes, to measure which bias attributes exhibit a larger bias. We provide examples of qualitative analyses in Sec. 4.3, and more in the *Supp. Mat.*, Appx. D.

4. Experiments

In this section, we present the results of our experiments, where we apply C2B to detect biases for different models on two different image classification tasks. We first describe our experimental protocol (Sec. 4.1), we then quantitatively compare the biases detected by C2B with a state-of-the-art competitor (Sec. 4.2), and we provide qualitative examples of detected biases (Sec. 4.3). Finally, we analyze the core component of our approach (Sec. 4.4).

4.1. Experimental protocol

Tasks and models. We test C2B on two different tasks: facial attribute classification and image classification.

Facial attribute classification. We consider the CelebA dataset [41], which is composed of 200k images of celebrity faces annotated with 40 facial attributes, such as *male*, *blond hair*, *young*, or *high cheekbones*. To evaluate C2B, we choose to detect the biases of a recent state-of-the-art model, FaceXFormer [48]. This model relies on a pre-trained Swin/B [42] backbone, coupled with an MLP-based fusion module and an attention-based decoder, which are co-trained on several face datasets.

Image classification. We focus on the widely used ImageNet-1K [54] dataset, where each image is associated to one of the 1000 ImageNet classes. We choose to detect the biases of four widely used models, easily accessible through TorchVision [43], comprising both convolutional networks and transformers: ResNet50_V2 [23], ResNet101_V2 [23], ResNet152_V2 [23], ViT_B_16_SWAG_E2E_V1 [58].

Competitors. As we introduce the new problem of unsupervised bias detection relying solely on a textual description of the classification task, there are no existing baseline methods designed for this specific setting. Nevertheless, we still compare C2B to a recent supervised open-set bias detection method, Bias-to-Text (B2T) [33]. However, unlike C2B which does not requires task-specific labeled data to identify biases in an image classifier, B2T depends on ground-truth task labels to identify model failures. It then employs a pre-trained captioning model to generate descriptions of misclassified images, extracts common keywords, and computes a bias score based on the difference between (i) the similarity (CLIP score) between the discovered bias keyword and misclassified images, and (ii) the similarity between the discovered bias keyword and well-classified images. Thus, we test B2T using the labeled validation sets of the two datasets (*i.e.*, CelebA and ImageNet).

Implementation details. C2B comprises two components: the LLM that proposes biases and captions, and the retrieval module. For the former, we use Llama 3.1 8B [18] (see *Supp. Mat.*, Appx. J for qualitative LLM comparisons). For the retrieval, we explore two strategies: (i) CC12M: we consider the large Conceptual Captions 12M (CC12M) dataset [12] and retrieve images based on CLIP embeddings [7], and (ii) BING: instead of using a VLM to retrieve images from an existing database where some image domains may not be well-represented, we propose to retrieve images from the Web using a publicly available search engine, Bing. For both datasets, we retrieve 20 images per caption, and we measure biases as in Eq. (1), comparing the difference in the model's per-class accuracy between images that match different candidate bias classes.

4.2. Quantitative evaluation

In this section, we provide a quantitative evaluation to assess the quality of biases identified by C2B. First, we present results based on ground-truth bias annotations, *i.e.*, exploiting attributes annotated in the validation sets of the datasets. For biases not in the ground-truth annotations, we introduce an evaluation based on Visual Question Answering (VQA) to measure the relevance of the proposed biases.

4.2.1 Evaluation with ground-truth annotations

Setting. In this setting, we exploit the available attribute annotations, checking if models can detect and quantify those biases correctly. In the case of CelebA, when we focus on a specific attribute classification task (*e.g.*, *blonde hair*) we consider all other 39 attributes (*e.g.*, *young*) as biases measurable with the ground-truth. For ImageNet-1K, we use ImageNet-X [28], *i.e.*, the validation set of ImageNet-1K with 16 labeled "factors" distinguishing images from typical prototypes (*e.g.*, *orientation*), as ground-truth biases.

To obtain ground-truth biases, we test the model on each validation set and check the difference in per-class accuracy among images where the bias attribute is present and images where it is absent. We consider a ground-truth bias to be present if this difference is higher than a threshold (*i.e.*, $\tau = 0.05$) using the same threshold to assess bias identification on both C2B and B2T scores. We refer to the sign of the bias score as the *direction* of the bias. We provide more results with a varying threshold in the *Supp. Mat.*, Appx. C.

Due to the open-set nature of the detected biases for both C2B and B2T, there are almost no exact matches between them and ground-truth biases. To compare the two sets, we compute the cosine similarity between SBERT embeddings [52] of the ground-truth and detected biases. We empirically set the similarity threshold to 0.9 to avoid having too many false positives. We provide more details and qualitative examples of matches in the *Supp. Mat.*, Appx. K.

Metrics. We report results from two complementary perspectives: GT → Detected and Detected → GT. In GT → Detected, we start from each ground-truth bias and check whether it was correctly detected with the same bias direction (HIT), detected with the opposite direction (false hit, FH), or not detected (MISS). This perspective focuses on recall. However, methods that propose many biases may artificially reduce misses. Therefore, we also report Detected → GT, where we start from each detected bias and verify whether it matches a ground-truth bias with the correct direction. This second view measures precision and helps assess the quality of the proposed biases.

Results. In Tabs. 1 and 2, we report the results of our analysis. On CelebA (left side of Tab. 1, GT → Detected), C2B outperforms B2T in detecting ground-truth biases, regardless of the retrieval source. We obtain the best results with Bing, detecting between 10.8% and 12.3% of the ground-truth biases, compared to only 4.5% for B2T. However, B2T exhibits a lower proportion of false hits (3.1%), whereas C2B shows slightly higher false hit rates (6.9–7.8%). This is largely due to retrieval noise and the broader search space introduced by LLM-generated proposals, which can sometimes lead to less precise bias candidates. Nevertheless, C2B misses fewer ground-truth biases (80.8–81.5% vs. B2T's 92.3%). These trends remain consistent when considering the proportion of detected biases that correspond to ground-truth biases (Detected → GT), where C2B achieves 12.8–14.2%, compared to 6.6% for B2T.

Table 1. Proportion (%) of ground-truth biases detected on CelebA (GT → Detected) and of detected biases corresponding to ground-truth ones (Detected → GT). FH=False Hit.

METHOD	GT → DETECTED HIT(↑)	GT → DETECTED FH(↓)	GT → DETECTED MISS (↓)	DETECTED → GT HIT(↑)	DETECTED → GT FH (↓)	DETECTED → GT MISS(↓)
			FaceXFormer			
B2T [33]	4.53	**3.14**	92.32	6.59	**4.10**	89.29
C2B (BING)	**12.29**	6.88	**80.83**	**14.18**	7.34	**78.48**
C2B (CC12M)	10.76	7.75	81.49	12.75	8.15	79.09

Table 2. Proportion (%) of ground-truth biases detected on ImageNet-X (GT → Detected) and of detected biases corresponding to ground-truth ones (Detected → GT). FH=False Hit.

METHOD	GT → DETECTED HIT(↑)	GT → DETECTED FH(↓)	GT → DETECTED MISS (↓)	DETECTED → GT HIT(↑)	DETECTED → GT FH (↓)	DETECTED → GT MISS(↓)
			ResNet50_V2			
B2T [33]	2.40	**2.19**	95.41	0.80	**0.71**	98.49
C2B (BING)	7.80	8.53	83.66	2.60	2.98	94.42
C2B (CC12M)	**11.18**	11.30	**77.52**	**2.99**	2.87	**94.14**
			ResNet101_V2			
B2T [33]	2.55	**2.11**	95.33	0.84	**0.69**	98.47
C2B (BING)	7.91	8.41	83.68	2.56	2.77	94.67
C2B (CC12M)	**11.21**	11.68	**77.11**	**2.76**	2.96	**94.28**
			ResNet152_V2			
B2T [33]	2.53	**1.72**	95.75	0.83	**0.58**	98.59
C2B (BING)	7.53	7.93	84.54	2.75	2.79	94.46
C2B (CC12M)	**11.10**	12.01	**76.90**	**2.81**	2.95	**94.24**
			ViT_B_16_SWAG			
B2T [33]	2.11	**1.98**	95.90	0.72	**0.67**	98.61
C2B (BING)	7.72	7.18	85.10	2.48	2.48	95.05
C2B (CC12M)	**10.85**	11.07	**78.08**	**2.64**	2.70	**94.66**

On ImageNet-X (Tab. 1), across four classifiers, C2B again detects more ground-truth biases than B2T (7.5–11.2% vs. 2.1–2.6%), at the cost of higher false hit rates (7.2–12.0% vs. 1.7–2.2%). As before, these false hits are attributable to noisy retrieval and the open-ended nature of bias proposals. Still, C2B misses significantly fewer ground-truth biases (76.9–85.1% vs. 95.3–95.9%). Interestingly, on ImageNet-X, retrieving from CC12M provides better results than Bing. The Detected → GT evaluation confirms these findings, with a larger fraction of C2B-detected biases aligning with ground-truth ones (2.5–3.0% vs. 0.6–0.7% for B2T).

Discussion. Due to the limited human annotations available in CelebA and ImageNet-X, the proportion of matching biases is low across all methods, particularly on ImageNet-X. Furthermore, since we are using text embeddings to

Table 3. Detected biases and VQA agreement on CelebA.

MODEL	B2T [33]	C2B (BING)	C2B (CC12M)
FaceXFormer	0.10	0.22	**0.25**

Table 4. Detected biases and VQA agreement on ImageNet-X.

MODEL	B2T [33]	C2B (BING)	C2B (CC12M)
ResNet50_V2	0.20	**0.26**	0.20
ResNet101_V2	0.19	**0.28**	0.20
ResNet152_V2	0.20	**0.28**	0.20
ViT_B_16_SWAG	0.20	**0.32**	0.21

match the biases, the numbers are also influenced by the semantic proximity between LLM-proposed biases, caption-extracted keywords, and ground-truth annotations. Moreover, a high number of MISS in the Detected $\rightarrow$ GT case is not necessarily a bad thing, as C2B and B2T may be detecting accurate biases that are not present in the annotation. For these reasons, in the next section we introduce a different evaluation based on visual question answering.

4.2.2 VQA-based evaluation

As ground-truth bias annotations may not always be available, we also propose an alternative evaluation scheme based on VQA, measuring the quality of the proposed biases that are not present in ground-truth annotations.

Metric. We use a VQA model (LLaVA-1.5-13B [38, 39]) to pseudo-label each image of the validation sets with each of the proposed bias by C2B or B2T. Additional details can be found in the *Supp. Mat.*, Appx. G. From these pseudo-labels, we can use the ground-truth task labels (*i.e.*, class annotations) to check how the per-class accuracy varies over different bias classes, using the same threshold $\tau = 0.05$ of Sec. 4.2.1. For each bias class b proposed by the model, there are three cases: a positive bias is detected ($\phi_b \geq \tau$), a negative bias is detected ($\phi_b \leq -\tau$), or the bias is not detected ($|\phi_b| < \tau$). For each bias class, we define the agreement score to be 1 if both scores fall in the same case, -1 if they detect opposite biases, and 0 otherwise.

Results. Tabs. 3 and 4 present the results of our VQA-based evaluation on CelebA and ImageNet-X. On CelebA, C2B achieves higher agreement with VQA pseudo-labels (0.22 for Bing and 0.25 for CC12M) than B2T (0.10), indicating that the biases discovered by C2B are more aligned with visually detectable and semantically coherent bias attributes. On ImageNet-X, while C2B often achieves higher agreement scores than B2T with Bing (*i.e.*, +0.09 on average), results with CC12M are either the same or comparable to those of B2T (*i.e.*, +0.005 on average). We attribute this to the challenge of relying on retrieval-based data collection, without access to annotated failure cases or task-specific captions. In contrast, B2T benefits from captions derived from known misclassifications. Despite this more challenging and fully unsupervised setting, C2B still achieves agreements that are competitive with or superior to B2T. This confirms that C2B's proposals are not only meaningful and visually grounded but also relevant for bias detection in settings where no annotations are available.

Discussion. The results presented in this section are influenced by both the types of questions posed to the VQA model and the model's own inherent biases. Given these limitations, evaluations based on ground-truth annotations and those derived from VQA outputs serve as complementary approaches. Overall, biases proposed by the LLM tend to offer more relevant bias indicators than keywords extracted from captions as in B2T [33], allowing C2B to detect a greater proportion of ground-truth biases and achieve higher agreement with the VQA model than B2T.

4.3. Qualitative evaluation

In this section, we show examples of biases discovered by C2B for different models on different tasks.

In Fig. 3, we show biases that C2B detects for the *high cheekbones* attribute on face attribute classification, with bias scores from different retrieval methods. We can see that a strong bias was detected on the *expression* attribute: the classifier appears to perform better when people are smiling and worse when they appear angry. Other smaller biases were detected, such as *young*, and *male*.

Fig. 4 shows a similar analysis for biases detected by C2B for the *minibus* class on image classification for the ResNet50_V2 model. Strong biases were detected on the *lighting*, *orientation*, and *presence of reflection* bias attribute: the classifier appears stronger on daytime images, portrait-oriented images, and images without reflections, while performing worse on nighttime images, landscape-oriented images, and images with reflections. Finally, Fig. 5 depicts biases for the *birdhouse* class again for image classification, but for the ViT_B_16_SWAG model. A strong bias is found on the *shape* bias attribute: the classifier seems to be performing better on images of rectangular birdhouses, while it underperforms on images of spherical birdhouses.

4.4. Evaluation of VLM-based retrieval

Retrieving images matching the target classes and biases present in the captions is essential for the detected biases to be trustworthy. Therefore, in this section, we analyze the CLIP-based retrieval approach used by C2B, by trying to retrieve images for which we have annotations. For a VQA-based analysis of both the Bing-based retrieval and CLIP-based retrieval on CC12M, please refer to *Supp. Mat.*, Appx. F. To perform our analysis, we use CLIP to retrieve images from the validation sets of CelebA and ImageNet-

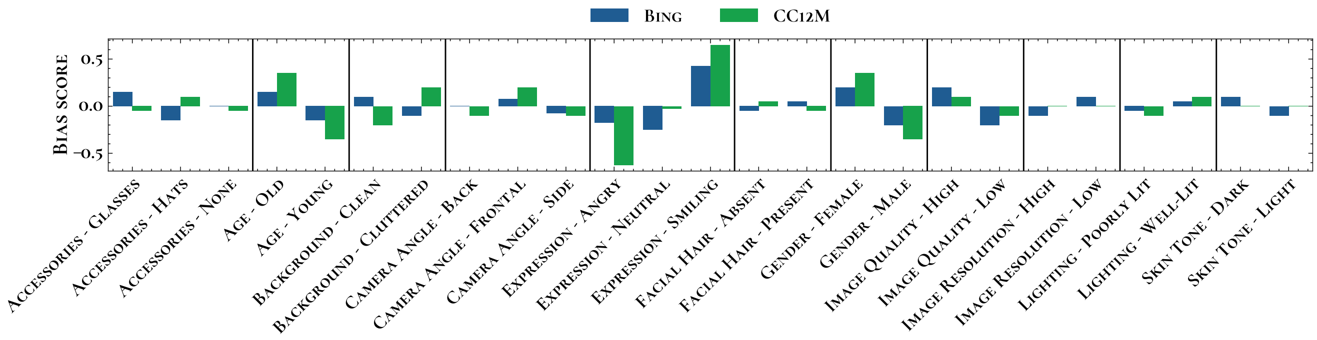

Figure 3. Examples of biases found by C2B for the *high cheekbones* target attribute on CelebA.

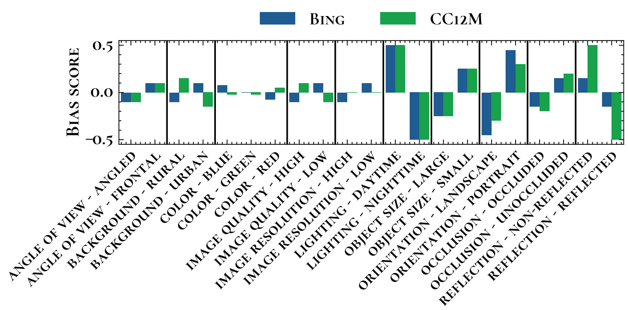

Figure 4. Examples of biases found by C2B for the *minibus* class on ImageNet-X (ResNet50_V2).

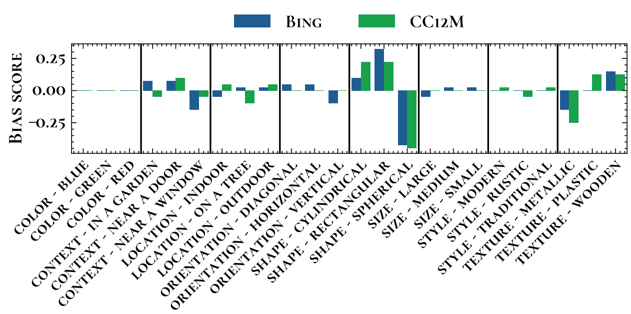

Figure 5. Examples of biases found by C2B for the *birdhouse* class on ImageNet-X (ViT_B_16_SWAG).

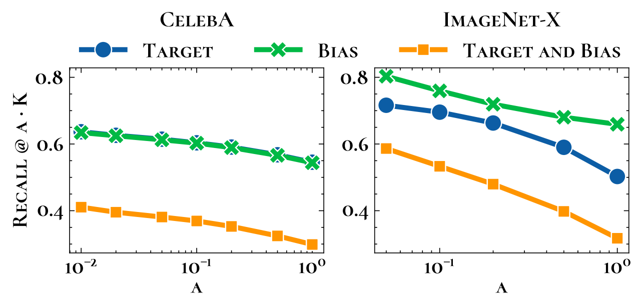

Figure 6. Accuracy of VLM-based retrieval.

X, using handcrafted captions as input: combinations of attributes on CelebA, and classes and factors on ImageNet-X.

Metric. We use recall@K, where K varies for each caption to match the number of corresponding ground-truth samples in the dataset. We show results using recall @ $A \cdot K$, where A is a coefficient that varies between 0.01 and 1.0.

Results. We present the results in Fig. 6. For CelebA (left), the recall is the same for both target and bias attributes, as expected, since both are drawn from the same set of attributes. On the same dataset, CLIP struggles to retrieve images with combinations of attributes, with a recall @ $0.01 \cdot K$ just above 40%. For ImageNet-X (right), CLIP is fairly accurate, especially for low values of A, with more than 80% and 70% recall at $0.05 \cdot K$ to retrieve images with specific biases and classes, respectively. Generally, the recall is higher when the number of retrieved images decreases relative to the number of potential matching images present in the dataset, which suggests that, the larger the dataset, the more accurate the retrieval is.

5. Conclusion

We introduced CLASSIFIER-TO-BIAS (C2B), a novel method for automatically detecting biases in visual classifiers entirely in an unsupervised manner, without the need for task-specific labeled data. Starting from a simple textual description of the classification task and a pre-trained model, C2B generates bias hypotheses, retrieves relevant images to test them, and produces a list of biases with their scores. Although C2B is the first approach to operate in such challenging setting, our results show that it surpasses a recent supervised open-set bias detector across two evaluation settings: one based on ground-truth annotations and another on VQA-based pseudo-labels. We believe our work marks an important step toward automated, open-set, and task-agnostic bias detection.

Acknowledgments

This work was sponsored by the MUR PNRR project FAIR - Future AI Research (PE00000013), funded by NextGeneration, and the Italian Ministerial grants PRIN 2022: "B-FAIR: Bias-Free Artificial Intelligence methods for automated visual Recognition" (CUP E53D23008010006), and the EU project ELIAS (No.01120237).

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 15

[2] Julius A Adebayo et al. *FairML: ToolBox for diagnosing bias in predictive modeling*. PhD thesis, Massachusetts Institute of Technology, 2016. 1

[3] Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *CVPR*, 2022. 2

[4] Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. Diffusion visual counterfactual explanations. *NeurIPS*, 2022. 2

[5] Razvan Azamfirei, Sapna R. Kudchadkar, and James Fackler. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):120, 2023. 2

[6] Yujia Bao and Regina Barzilay. Learning to split for automatic bias detection. *arXiv preprint arXiv:2204.13749*, 2022. 1

[7] Romain Beaumont. Clip retrieval: Easily compute clip embeddings and build a clip retrieval system with them. `https://github.com/rom1504/clip-retrieval`, 2022. 5

[8] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilović, S. Nagar, K. Natesan Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4:1–4:15, 2019. 1

[9] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NeurIPS*, 2016. 1

[10] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, 2018. 1

[11] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 46–56, 2019. 1

[12] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 5

[13] Davide Chicco, Niklas Tötsch, and Giuseppe Jurman. The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining*, 14(1):13, 2021. 11

[14] Massimiliano Ciranni, Luca Molinaro, Carlo Alberto Barbano, Attilio Fiandrotti, Vittorio Murino, Vito Paolo Pastore, and Enzo Tartaglione. Say my name: a model's bias discovery framework. *arXiv preprint arXiv:2408.09570*, 2024. 2

[15] Gabriela Csurka, Tyler L Hayes, Diane Larlus, and Riccardo Volpi. What could go wrong? discovering and describing failure modes in computer vision. In *ECCV-WS*, 2024. 3

[16] Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1962–1981, 2022. 1, 2

[17] Moreno D'Incà, Elia Peruzzo, Massimiliano Mancini, Dejia Xu, Vidit Goel, Xingqian Xu, Zhangyang Wang, Humphrey Shi, and Nicu Sebe. Openbias: Open-set bias detection in text-to-image generative models. In *CVPR*, 2024. 1, 2

[18] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 5, 15

[19] Sabri Eyuboglu, Maya Varma, Khaled Kamal Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Re. Domino: Discovering systematic errors with cross-modal embeddings. In *ICLR*, 2022. 1, 2

[20] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei W Koh, Olga Saukh, Alexander J Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems*, pages 27092–27112, 2023. 2

[21] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50 (3):1097–1179, 2024. 2

[22] Irena Gao, Gabriel Ilharco, Scott Lundberg, and Marco Tulio Ribeiro. Adaptive testing of computer vision models. In *ICCV*, 2023. 1, 3

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[24] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, 2018. 1

[25] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language. In *ICML-WS*, 2018. 2

[26] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *NeurIPS*, 2024. 2

[27] https://huggingface.co/. 1

[28] Badr Youbi Idrissi, Diane Bouchacourt, Randall Balestriero, Ivan Evtimov, Caner Hazirbas, Nicolas Ballas, Pascal Vincent, Michal Drozdzal, David Lopez-Paz, and Mark Ibrahim. Imagenet-x: Understanding model mistakes with factor of variation annotations. In *ICLR*, 2023. 1, 2, 6

[29] Saachi Jain, Hadi Salman, Eric Wong, Pengchuan Zhang, Vibhav Vineet, Sai Vemprala, and Aleksander Madry. Missingness bias in model debugging. In *ICLR*, 2022. 2

[30] Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling model failures as directions in latent space. In *ICLR*, 2023. 1, 2

[31] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), 2023. 2

[32] Krish Kabra, Kathleen M. Lewis, and Guha Balakrishnan. Gelda: A generative language annotation framework to reveal visual biases in image generators. In *CVPRW*, 2024. 2

[33] Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. Discovering and mitigating visual biases through keyword explanation. In *CVPR*, 2024. 1, 2, 5, 6, 7, 4

[34] Arvindkumar Krishnakumar, Viraj Prabhu, Sruthi Sudhakar, and Judy Hoffman. Udis: Unsupervised discovery of bias in deep visual recognition models. In *BMVC*, 2021. 1, 2

[35] Zhiheng Li and Chenliang Xu. Discover the unknown biased attribute of an image classifier. In *ICCV*, 2021. 2

[36] Zhiheng Li, Anthony Hoogs, and Chenliang Xu. Discover and mitigate unknown biases with debiasing alternate networks. In *ECCV*, 2022. 2

[37] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *ICML*, 2021. 2

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 7, 11

[39] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 7, 11

[40] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 11

[41] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 2, 5

[42] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5

[43] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision`, 2016. 5

[44] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54 (6):1–35, 2021. 3

[45] Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 15

[46] Microsoft. Chat markup language chatml (preview), 2024. `https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/chat-markup-language`, Last accessed on 2024-11-21. 13

[47] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019. 1

[48] Kartik Narayan, Vibashan VS, Rama Chellappa, and Vishal M Patel. Facexformer: A unified transformer for facial analysis. *arXiv preprint arXiv:2403.12960*, 2024. 5

[49] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464): 447–453, 2019. 1

[50] Jesutofunmi A. Omiye, Jenna C. Lester, Simon Spichak, Veronica Rotemberg, and Roxana Daneshjou. Large language models propagate race-based medicine. *npj Digital Medicine*, 6(1):195, 2023. 2

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[52] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *ENMLP*, 2019. 6

[53] Keivan Rezaei, Mehrdad Saberi, Mazda Moayeri, and Soheil Feizi. PRIME: Prioritizing interpretability in failure mode extraction. In *ICLR*, 2024. 1, 3

[54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 5

[55] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine

Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, pages 25278–25294, 2022. 2

[56] Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *ICML*, 2024. 3

[57] Robik Shrestha, Yang Zou, Qiuyu Chen, Zhiheng Li, Yusheng Xie, and Siqi Deng. Fairrag: Fair human generation via fair retrieval augmentation. In *CVPR*, 2024. 2

[58] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. Revisiting weakly supervised pre-training of visual perception models. In *CVPR*, 2022. 5

[59] Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning? In *ICLR*, 2022. 2

[60] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz. Understanding failures of deep networks via robust feature extraction. In *CVPR*, 2021. 1

[61] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *NeurIPS*, 2020. 2

[62] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022. 2

[63] Vijay Vasudevan, Benjamin Caine, Raphael Gontijo Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. When does dough become a bagel? analyzing the remaining mistakes on imagenet. In *NeurIPS*, 2022. 1

[64] Angelina Wang, Alexander Liu, Ryan Zhang, Anat Kleiman, Leslie Kim, Dora Zhao, Iroha Shirai, Arvind Narayanan, and Olga Russakovsky. Revise: A tool for measuring and mitigating bias in visual datasets. *IJCV*, 130(7):1790–1810, 2022. 1

[65] Olivia Wiles, Isabela Albuquerque, and Sven Gowal. Discovering bugs in vision models using off-the-shelf image generation and captioning. In *NeurIPS-WS*, 2022. 2

[66] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023. 2

[67] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023. 2

[68] Zeliang Zhang, Mingqian Feng, Zhiheng Li, and Chenliang Xu. Discover and mitigate multiple biased subgroups in image classifiers. In *CVPR*, 2024. 1, 2

[69] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *ICCV*, 2021. 1

[70] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017. 1

[71] Zaiying Zhao, Soichiro Kumano, and Toshihiko Yamasaki. Language-guided detection and mitigation of unknown dataset bias. *arXiv preprint arXiv:2406.02889*, 2024. 2