

1	Supplementary Material	
2	A Method Details: ConceptScope	2
3	A.1 SAE Training	2
4	A.2 Concept Dictionary	2
5	A.3 Concept Categorization: Target, Bias, and Context	3
6	B Experiment Details: Concept Prediction	6
7	B.1 Datasets	6
8	B.2 Baselines	6
9	B.3 Additional Results: Concept prediction	7
10	B.4 Measuring correlation between SAE activation with CLIP similarity	8
11	B.5 Additional Results: Correlation with CLIP embedding	8
12	C Experiment Details: Bias Discovery	9
13	C.1 Setup	10
14	C.2 Datasets	10
15	C.3 Baselines	10
16	C.4 Qualitative results: examples of discovered known biases	11
17	C.5 Additional Results: Examples of discovered novel biases	11
18	D Experiment Details: Model Robustness Analysis	12
19	D.1 Full list of models evaluated on ImageNet	12

20 A Method Details: ConceptScope

21 In this section, we provide additional technical details and illustrative examples for our proposed
22 framework ConceptScope, supplementing §3 of the main paper.

23 A.1 SAE Training

24 We include details regarding hyperparameters and computational resources for training the SAE,
25 supplementing the explanations provided in §3.2 of the main paper.

26 **Hyperparameter settings.** To train SAE, we utilize OpenAI CLIP ViT-L/14¹ as base vision encoder,
27 extracting embeddings from its penultimate (23rd) layer with an embedding dimensionality d of
28 1024. The vision encoder resizes the input images to 224×224 . Given a patch size of 14, this resize
29 results in 257 tokens consisting of one class token ([CLS]) and 16×16 spatial tokens. We applied an
30 expansion factor of 32, resulting in a total SAE latent dimensionality d' of 32,768. The L_1 sparsity
31 loss weight λ was set to $8e - 5$, and the learning rate was set to $4e - 4$ with a constant warmup
32 scheduling over 500 steps. We initialized decoder bias with geometric median [23] of training data
33 (we use the first training batch) and applied ghost gradient [2], a neuron resample technique designed
34 to mitigate the issue of dead neurons due to sparsity regularization. We set the batch size as 64. For
35 the training dataset, we use ImageNet-1K² training split, comprising approximately 1.28 million
36 images, and trained for a total of 200,000 steps.

37 **Computation resources.** We used a single NVIDIA RTX A6000 GPU, requiring approximately
38 0.79 seconds per iteration with a batch size of 64, consuming 24 GB of GPU memory and performing
39 2,208 GFLOPs per iteration.

40 A.2 Concept Dictionary

41 We provide implementation details for constructing the concept dictionary introduced in §3.2 of the
42 main paper, including methods for filtering meaningful latent dimensions and naming concepts using
43 LLMs.

44 **Latent filtering.** As discussed in Lim et al. [15], not all SAE latent dimensions are interpretable and
45 monosemantic, i.e., we observe that multiple unrelated concepts activate the same latent. For enhanced
46 interpretability, we filter out non-informative or uninterpretable latents and use only interpretable ones
47 for the analysis. First, we compute image-level activation values $f(\mathbf{z})_c$ across all latent dimensions
48 for the entire ImageNet training dataset. We then filter out latent dimensions that never activate
49 beyond an activation value of 0.5 for any image, as well as those whose average concept strength
50 exceeds 0.1. The former are non-informative, while the latter activate frequently across many images
51 and thus lack interpretability. After applying these filters, we retain 3,103 latent dimensions.

52 **Naming latents.** To assign semantic labels to latent dimensions, we retrieve the top five images per
53 latent and generate their segmentation masks. We then query the GPT-4o API³ using these images
54 alongside the following prompt:

Prompt

Identify the shared concept among images, focusing solely on non-blocked
areas and excluding dark silhouette areas, using 2-3 clear and specific words.
Answer with concepts only.

55
56 Annotating all 3,103 latent dimensions incurs a total cost of approximately \$7 USD.

57 **Examples of discovered concepts** In Fig. 1, we visualize examples of the discovered concepts along
58 with segmentation masks for the five most highly activated images from ImageNet. These latents

¹<https://huggingface.co/openai/clip-vit-large-patch14>

²https://huggingface.co/datasets/evanarlian/imagenet_1k_resized_256

³<https://platform.openai.com/docs/models/gpt-4o>

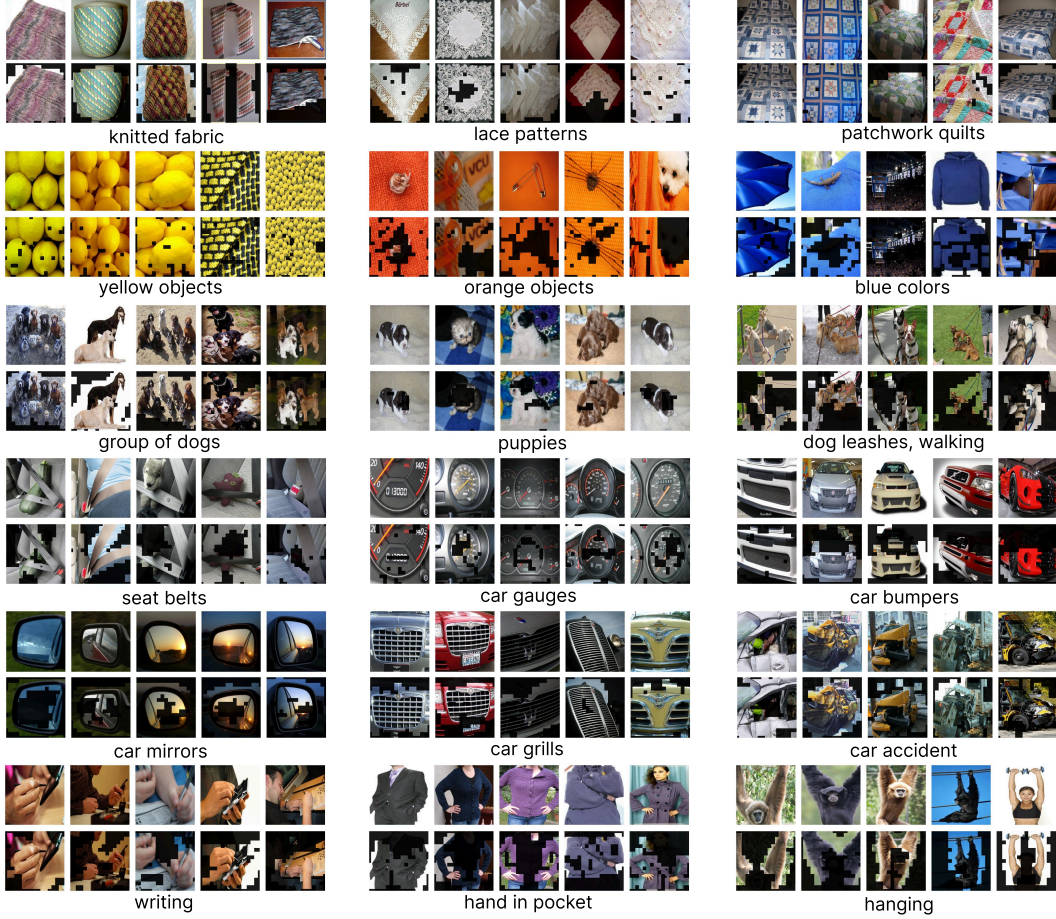


Figure 1: **Examples of discovered concepts.** The top row shows the five images from ImageNet with the highest activations for each latent dimension, while the bottom row shows the corresponding segmentation-mask-applied images. The concepts identified include various colors, textures, objects, and actions. The assigned semantic labels are generated by LLMs.

represent diverse visual concepts, encompassing colors, textures, objects, and actions. Additionally, the discovered concepts capture fine-grained distinctions within broader categories; for instance, car-related concepts include detailed components such as seat belts, gauges, numbers, rear-view mirrors, grills, and scenes depicting car accidents. Similarly, dog-related concepts identify specific scenarios, including groups of dogs, puppies, and dog leashes. Human actions such as writing, placing hands in pockets, and hanging objects are also among the concepts captured by these latents.

A.3 Concept Categorization: Target, Bias, and Context

In this section, we provide supplementary implementation details for computing alignment scores, along with illustrative examples of concept categorization from the ImageNet dataset, complementing §3.3 of the main paper.

Implementation details for computing alignment scores As explained in §3.3, we categorize each class’s concepts into *target* and *context* concepts by measuring the differences in cosine similarity between class labels and concept-ablated images. To compute text embeddings, we use only the class name without any additional textual prompts. For each class, we randomly select 100 images from the training dataset and calculate sufficiency and necessity scores (Eq. 3 in the main paper). Empirically, we observe that the top 20 concepts per class exhibit significantly higher mean activations, while the remaining concepts show negligible activations. We therefore compute alignment scores exclusively for these top 20 latents.

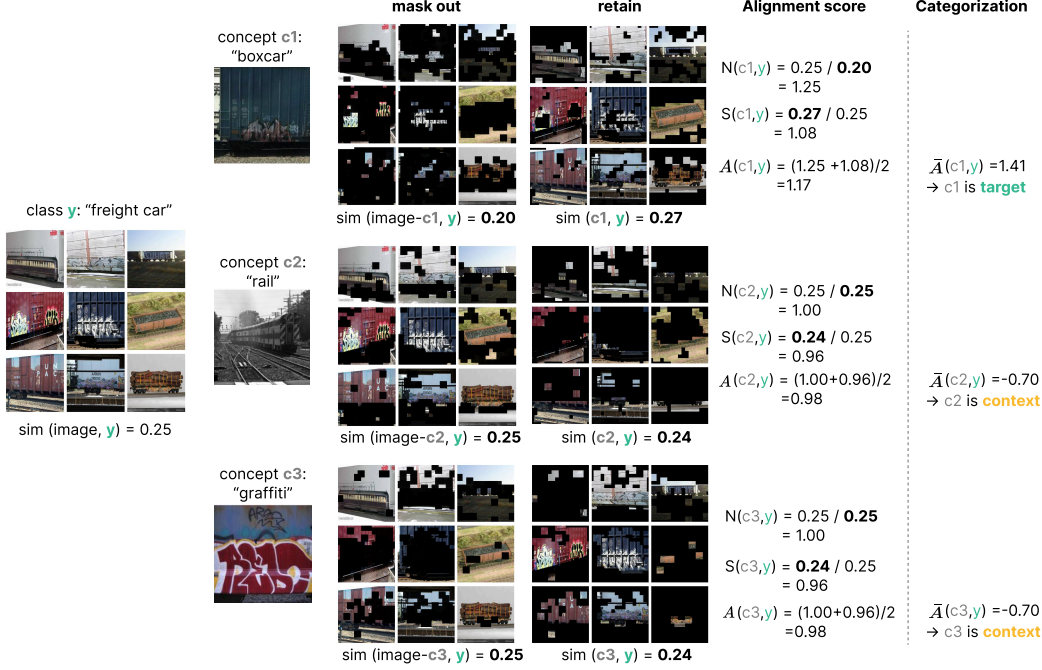


Figure 2: **Examples illustrating the computation of alignment scores.** We present the detailed process for calculating alignment scores of concepts associated with the class `freight car` from ImageNet dataset, including their actual cosine similarity values, as supplementary to Fig. 2 (c) in the main paper.

Example of computing alignment scores. An illustrative example from the `freight car` class in ImageNet is presented in Fig. 2. As shown, masking out regions activated by the concept `container` reduces the cosine similarity relative to the original image (from 0.25 to 0.20). In contrast, retaining only these activated regions increases the similarity (from 0.25 to 0.27). On the other hand, masking regions activated by concepts such as `rail` or `graffiti` causes negligible changes in cosine similarity, while retaining only these regions decreases the similarity. After normalization, the `container` concept yields a positive alignment score, indicating a target concept, whereas `rail` and `graffiti` yield negative scores, indicating context concepts. Since the alignment score reflects relative changes rather than absolute similarities, our method remains robust across classes that vary widely in their baseline similarities.

Example of concept categorization. Fig. 3 illustrates the concept activation and categorization of concepts identified using ConceptScope. The y-axis denotes the average image-level concept activation value $f(z)$ computed over all images belonging to class y , i.e., $\tilde{f}_{c,y} = \text{avg}_{z_y \in Z_y} (f(z_y)_c)$, where Z_y represents the set of image embeddings labeled as class y . The x-axis displays concepts sorted according to their alignment scores. Positive alignment scores indicate target concepts, with higher positive values (encoded in darker green) corresponding to more frequently captured and representative concepts. For the `freight car` class, the primary target concept is a zoomed-in view of a `boxcar`, while secondary target concepts include side or front views of `trains`, as illustrated on the right side of the figure. Negative alignment scores indicate context concepts, with more negative values (encoded in darker yellow) representing concepts that are irrelevant to the target and less frequently captured. For instance, concepts such as `small stones`, `outdoors`, or `rusty objects` are infrequently captured and generally irrelevant. Conversely, `graffiti` and `rails` appear more commonly but lack sufficient necessity to independently identify an image as a `freight car`.

Concepts discovered by ConceptScope provide valuable insights into target classes by effectively identifying both their diverse visual states and stereotypical examples. Fig. 4 illustrates additional examples of categorized concepts from various ImageNet classes, particularly focusing on classes where typical visual representations and their diversity may not be immediately intuitive.

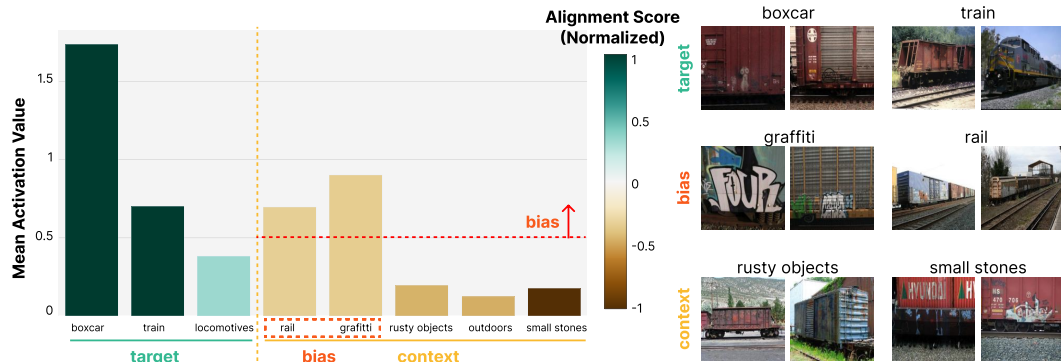


Figure 3: **Examples of concept categorization.** As a supplement to Fig. 2 (d) in the main paper, we show the average concept activation values for each identified concept within the class freight car from the ImageNet dataset. Colors represent the alignment score values. Example images corresponding to each concept are provided on the right.

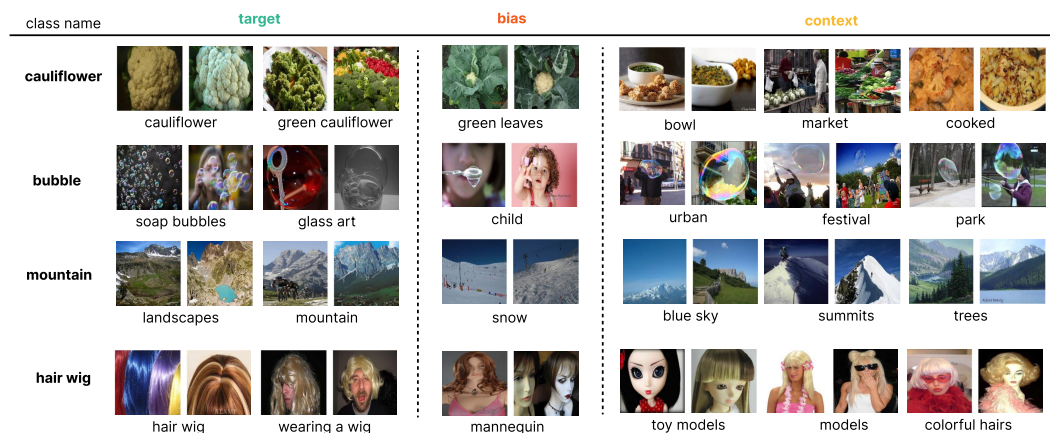


Figure 4: **Additional examples of categorized concepts from ImageNet classes.** To further illustrate the outputs of concept categorization, we provide additional examples of target, context, and bias concepts. ConceptScope effectively identifies prototypical images for target classes, as well as the contexts in which these targets typically appear (context concepts), and highlights the most frequent contexts that may introduce bias (bias concepts).

104 ConceptScope clearly identifies prototypical examples through target concepts, reveals variations
 105 using context concepts, and highlights dominant contexts as bias concepts.

B Experiment Details: Concept Prediction

In this section, we provide additional experimental details to supplement Section 4 of the main paper, including descriptions of datasets, baselines, and further results.

B.1 Datasets

Caltech-101 [8] The Caltech-101 dataset is a widely used benchmark for image classification tasks. It contains images from 102 object categories, totaling 9,146 images. In our experiments, we use the training split provided by HuggingFace⁴, which includes 3,060 images.

DTD [5] The Describable Textures Dataset (DTD) is specifically designed for texture recognition tasks. It consists of 5,650 images in total, from which we use the test split containing 1,880 images⁵.

Waterbirds [24] The Waterbirds dataset consists of bird images across four different background types: bamboo forest, forest, ocean, and lake. We use the test split, which contains 5,794 images.

CelebA [19] The CelebA dataset consists of cropped celebrity images annotated with 40 binary facial attributes. We utilize the test split, which contains 19,962 images. Although this dataset is widely employed, many attributes are subjective (e.g., attractive, big lips, or pointy nose) or overly fine-grained (e.g., facial hair divided into 5 o'clock shadow, mustache, sideburns, goatee). To maintain clarity and relevance, we select a subset of attributes primarily related to hairstyles and accessories. The selected 15 attributes are: bald, bangs, black hair, blond hair, brown hair, eyeglasses, gray hair, male, smiling, wavy hair, straight hair, wearing earrings, wearing hat, wearing necklace, wearing necktie, and young.

RAF-DB [14] The RAF-DB (Real-world Affective Faces Database) is a widely used dataset for emotion recognition research, comprising seven emotional categories: happy, sad, surprise, fear, disgust, anger, and neutral. The full dataset contains approximately 30,000 images. For our experiments, we randomly sampled 100 images per class, resulting in a total of 700 images.

Stanford40 Actions [29] The Stanford40 Action Dataset is an image classification dataset commonly used in action recognition research. It comprises images of humans performing 40 distinct actions, including playing guitar, riding a bike, cooking, and reading. The dataset contains approximately 9,500 images in total. For our experiments, we sampled 100 images from each class, resulting in a subset of 4,000 images.

B.2 Baselines

The primary objective of this binary attribute classification experiment in § 4.1 is to verify whether individual latents consistently and selectively correspond to distinct visual concepts. Specifically, a latent dimension representing a given concept (e.g., “chair”) should reliably activate across diverse instances of that concept while remaining inactive for unrelated concepts (e.g., “table”). Given the absence of directly comparable baselines, we instead compare our approach against VLM-based methods as suitable alternatives.

Setup. For the VLM-based baseline methods used in §4.1 (Table 1) in the main paper, we followed SSD-LLM [20] but with simplification. SSD-LLM is designed to identify distribution of attributes such as backgrounds, actions, or co-occurring objects related to a target class through four steps: (1) caption extraction using VLMs, (2) identification of major attributes from generated captions, (3) iterative refinement of the attribute list, and (4) final assignment of attributes to captions. Although the original method heavily relies on large language models (LLMs) to carry out steps 2 to 4, we simplify the attribute proposal (steps 2 and 3) and assignment stage (step 4) by employing straightforward keyword matching. Keyword matching is similarly used in ConBias [3], which identifies attributes

⁴<https://huggingface.co/datasets/dpdl-benchmark/caltech101>

⁵<https://huggingface.co/datasets/tanganke/dtd>

by matching manually curated keyword sets against VLM-generated captions. Unlike ConBias, we incorporate synonym matching rather than exact matching to enhance flexibility and coverage.

Implementation details. Specifically, we first construct a keyword mapping dictionary containing synonyms and phrases semantically related to each target class label. To generate these synonym lists, we use GPT-4o with the following prompt:

Prompt
"For the given class label, provide synonyms or phrases that are semantically equivalent. Return the output in JSON array format."

An example of such a dictionary entry is:

```
"airplane": [
  "airplane", "airplanes", "plane", "jet", "aircraft",
  "airliner", "passenger plane", "commercial aircraft",
  "propeller plane", "boeing", "airbus", ...
]
```

We then consider an attribute prediction correct if the generated caption contains any of these keywords.

VLM details. We employ the BLIP-2 [13] model architecture with a 2.7B parameter model⁶. Input images are resized to 256×256 , and captions are generated using the prompt: "Describe this image in detail." For LLaVA-NeXT [12], we utilize the 8B parameter model⁷. The same prompt used for BLIP-2 is applied, with a maximum token limit set to 256.

B.3 Additional Results: Concept prediction

Table 1: **Performance comparison for concept prediction.** We compare our SAE-based method (ConceptScope) against caption-based baselines (BLIP-2, LLaVA-NeXT), reporting average *precision* and *recall* scores across classes.

Method	Metric	Caltech101 [8] (Objects)	DTD [5] (Textures)	Waterbird [24] (Backgrounds)	CelebA [18] (Facial Attr.)	RAF-DB [14] (Emotions)	Stanford40 [29] (Actions)	Average
BLIP-2 [13]	<i>precision</i>	0.83±0.29	0.59±0.43	0.81±0.18	0.80±0.33	0.43±0.38	0.89±0.29	0.72
	<i>recall</i>	0.75±0.34	0.29±0.30	0.49±0.09	0.34±0.29	0.20±0.22	0.37±0.30	0.41
LLaVA-NeXT [12]	<i>precision</i>	0.72±0.33	0.56±0.31	0.75±0.19	0.72±0.15	0.61±0.15	0.84±0.27	0.70
	<i>recall</i>	0.69±0.35	0.44±0.27	0.53±0.07	0.57±0.27	0.47±0.25	0.37±0.35	0.51
ConceptScope (Ours)	<i>precision</i>	0.80±0.21	0.53±0.28	0.77±0.16	0.62±0.20	0.50±0.25	0.72±0.20	0.66
	<i>recall</i>	0.88±0.14	0.56±0.22	0.81±0.06	0.74±0.13	0.50±0.25	0.81±0.15	0.76

SAE activations achieve higher recall than VLM-based baselines despite slightly lower precision, highlighting limitations of VLMs in capturing specific visual concepts. In Table 1, we report precision and recall scores for concept prediction tasks as complementary results to Table 1 of the main paper. Additionally, Fig. 5 shows precision-recall curves for SAE at varying activation thresholds (blue), with baseline performances indicated by red "✗" marks for BLIP-2 captions and green "✕" marks for LLaVA-NeXT captions. As shown in both the table and the figure, the VLM-based baselines achieve slightly higher precision compared to SAE activations; however, their average recall is significantly lower. This reduction in recall primarily arises from the linguistic diversity inherent in VLM-generated captions. Specifically, as demonstrated in Fig. 6, VLM-generated captions tend to describe general visual characteristics but often fail to capture the specificity needed for identifying target attributes. For example, they describe a bubbly pattern merely as irregular shape, blonde hair as light-colored hair, or simply standing rather than specifically fishing. These observations suggest that while VLMs excel at capturing broad visual descriptions, they are less effective in consistently identifying and quantifying discriminative visual concepts within image datasets.

⁶<https://huggingface.co/Salesforce/blip2-opt-2.7b>

⁷<https://huggingface.co/llava-hf/llama3-llava-next-8b-hf>

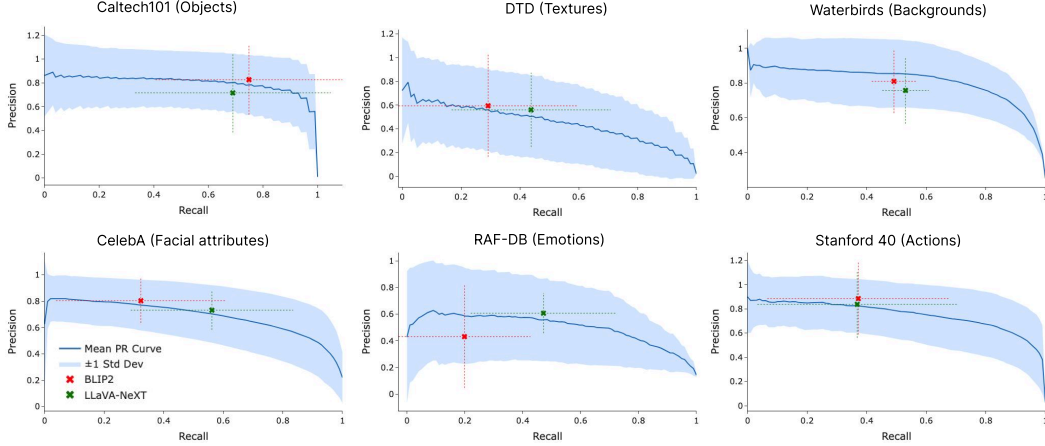


Figure 5: **Average precision-recall curve for attribute prediction using SAE activations.** The shaded blue region represents ± 1 standard deviation. Red and green markers indicate the mean precision and recall scores of BLIP-2 [13] and LLaVA-NeXT [12], respectively, with dashed lines showing ± 1 standard deviation.

184 B.4 Measuring correlation between SAE activation with CLIP similarity

185 In § 4.1 of the main paper, we report the average correlation between SAE activations and CLIP
 186 similarity scores across six datasets. Here, we provide supplementary implementation details and
 187 comprehensive results for each individual dataset.

188 **Implementation details.** For each target class, we first select images where the SAE activation
 189 for the target concept latent is greater than zero. Next, we compute the cosine similarity between
 190 the image embeddings and the text embedding derived from the class label. We sort the selected
 191 images based on SAE activation values and partition them into 100 percentile-based groups. For each
 192 group, we calculate the average SAE activation and average CLIP similarity. Finally, we evaluate the
 193 correlation between SAE activation and CLIP similarity using Pearson and Spearman correlation
 194 coefficients.

195 B.5 Additional Results: Correlation with CLIP embedding

Table 2: **Correlation of SAE activation (Ours) & CLIP similarity.** High Pearson and Spearman scores confirm that SAE activation strength reliably represents the degree of visual concept presence.

Metric	Caltech101 [8] (Objects)	DTD [5] (Textures)	Waterbird [24] (Backgrounds)	CelebA [18] (Facial Attr.)	RAF-DB [14] (Emotions)	Stanford40 [29] (Actions)	Average
Pearson	0.68 \pm 0.27	0.77 \pm 0.14	0.77 \pm 0.15	0.79 \pm 0.30	0.70 \pm 0.25	0.92 \pm 0.05	0.77
Spearman	0.52 \pm 0.29	0.78 \pm 0.16	0.94 \pm 0.03	0.77 \pm 0.32	0.58 \pm 0.31	0.81 \pm 0.13	0.73

196 **SAE activations for target concepts strongly correlate with CLIP similarity scores.** Table 2
 197 presents the Pearson and Spearman correlations between SAE activations and the CLIP text-image
 198 cosine similarity scores across various attributes. The consistently high correlations indicate that
 199 SAE activations reliably align with the CLIP embedding similarities, highlighting the reliability of
 200 SAE-derived visual concepts.

Dataset & Class	Input image	SAE activation	BLIP2	LLaVA-NeXT
Caltech101 "ant"			"a diagram of a fly and a bee "	"The image is a black and white illustration of a grasshopper . The grasshopper is depicted in a side view, with its body oriented horizontally across the image. The body of the grasshopper is elongated and segmented, with each segment clearly defined."
DTD "bubbly"			"abstract blue glitter texture background"	"The image displays a close-up of a textured surface with a blue hue. The texture appears to be a pattern of irregular shapes and lines, giving the impression of a cracked or shattered surface."
Waterbirds "bamboo"			"a young girl is holding a bird in the air"	"In the image, a young girl is standing in a field of tall, green grass . She is dressed in a blue shirt and a white hat, providing a contrast to the greenery around her. In her hands, she holds a black bird, which is perched on her arm."
CelebA "blond hair"			"a man with a beard and a black jacket "	"The image is a close-up portrait of a person with a neutral background that appears to be a wall with a patterned texture. The individual has short, light-colored hair that is styled to the side."
RAF-DB "angry"			"a boy is peeking out of a cardboard box "	"The image shows a close-up of a person's face peeking through a hole in a cardboard box. The person appears to be a young child with fair skin and dark hair. The child's eyes are wide open, and their mouth is slightly open, revealing teeth. The child's expression is one of surprise or shock ."
Stanford-action "fishing"			"a man standing in the grass near a river"	"In the image, a man is standing on a grassy bank, his gaze directed towards a body of water. He is dressed in a blue baseball cap, a plaid shirt, and blue jeans. The water, a light green color, is bordered by a line of trees and shrubs, adding a touch of nature to the scene."

Figure 6: **Examples of failure cases from VLM-based baselines.** This figure provides illustrative examples explaining why the VLM-based baselines yield lower performance compared to SAE, as reported in Table 1 of the main paper. Although the generated captions accurately describe images at a general level, they frequently lack the specificity needed to effectively capture target attributes, leading to overly simplified or generalized descriptions.

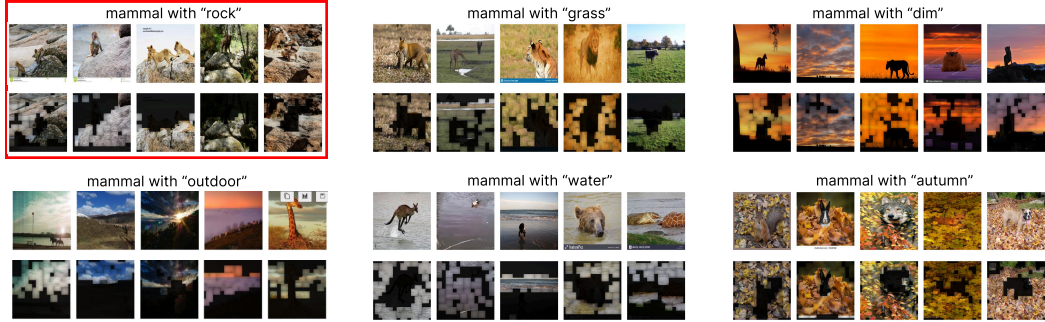


Figure 7: **The qualitative examples of discovered biases.** We illustrate the top 5 test samples of the Nico++ dataset, labeled as “mammals” from each attribute-specific subset. The subset highlighted with a red box corresponds to the attribute correlated with the “mammals” class during training (rock background), while the remaining subsets represent spurious attributes associated with other classes. As shown in the figure, each subset consistently shares the same background attributes, clearly visible in the corresponding segmentation masks shown in the bottom row.

201 C Experiment Details: Bias Discovery

202 This section provides additional details for the bias discovery experiments presented in § 5.1 and
203 § 5.2 of the main paper.

204 C.1 Setup

205 This subsection provides detailed explanations of the experimental setups, evaluation metrics, datasets,
206 and baselines, as well as additional qualitative results, for the bias discovery experiment described in
207 Section 5.1 of the main paper.

208 **Evaluation metric.** We measure the performance using Precision@10 [7], which is calculated
209 as follows: for each candidate spurious attribute, the ten test images predicted with the highest
210 confidence for containing that attribute are selected. Precision@10 is then computed as the fraction
211 of these selected images that truly contain the attribute. The final score is the average Precision@10
212 across all subset-attribute pairs.

213 C.2 Datasets

214 **Waterbirds [24].** The Waterbirds dataset contains two classes: waterbirds and landbirds. The
215 training set comprises 4,795 images. Of these, 95% of the waterbird images have ocean or lake
216 backgrounds, while the remaining 5% feature forest or bamboo forest backgrounds. Conversely, 95%
217 of landbird images have forest or bamboo forest backgrounds, with the remaining 5% featuring ocean
218 or lake backgrounds. The test set consists of 5,794 images, balanced equally across backgrounds
219 (50% water backgrounds, 50% land backgrounds) for both classes.

220 **CelebA [19].** Following prior work, we use CelebA for a binary classification task focused on the
221 attribute blond hair. In the training set, approximately 93% of individuals with blond hair are female.
222 The training dataset consists of 162,770 images, and the test dataset consists of 19,962 images.

223 **Nico++ [33]** Yenamandra et al. [33] modified the original dataset into a six-way classification task,
224 including the classes: mammals, birds, plants, waterways, landways, and airways. Each class is
225 strongly correlated with a particular background: rocks, grasses, dim lighting, water, outdoor, and
226 autumn, respectively. The severity of the bias in the training dataset is varied across three levels,
227 75%, 90%, and 95%, indicating the proportion of images in each class exhibiting the correlated
228 background attribute within the training set. In contrast, the test set was constructed to be balanced,
229 with backgrounds evenly distributed across each class. This resulted in six distinct subsets per class,
230 each has one of the 6 background attributes.

231 C.3 Baselines

232 Typical baseline methods for bias discovery involve two stages: first, training models on biased
233 datasets; and second, identifying biases based on model failures in test data through clustering
234 approaches that leverage various features such as model predictions, ground-truth labels, and feature
235 embeddings.

236 **DOMINO [7].** DOMINO fits Gaussian mixture models (GMMs) using model predictions, ground-
237 truth labels, and CLIP image embeddings for each input image. The GMM clusters data samples
238 based on error types, such as false positives or false negatives. Subsequently, DOMINO generates
239 natural language descriptions of biases by identifying the most similar keywords corresponding to
240 each cluster within the CLIP embedding space.

241 **FACTS [30].** FACTS extends DOMINO by adding a specialized training objective that amplifies
242 the model’s reliance on biased attributes. Specifically, this objective encourages higher confidence
243 scores for samples with biased attributes and lower scores for those without. FACTS then fits separate
244 GMMs within each class to identify distinct subsets associated with bias attributes. Finally, FACTS
245 generates image captions using pretrained captioning models [22] and identifies biases by extracting
246 the most frequently occurring keywords across these captions.

247 **ViG-Bias [21].** ViG-Bias further improves upon FACTS by incorporating visual explanations into
248 the bias discovery process. It first generates masks using Grad-CAM [25] heatmaps derived from
249 the trained model. In the second stage, ViG-Bias uses these masked image embeddings, rather than
250 original CLIP embeddings, to perform clustering.

Comparison with baselines. All baseline methods described above require model training and identify biases based on model errors observed in the test dataset. In contrast, ConceptScope does not require training any additional models and instead directly identifies biases from the dataset itself. Furthermore, baseline methods assume that each class contains only a single type of bias with high correlation (typically greater than 75% of samples per class). However, this assumption rarely holds true for real-world datasets such as ImageNet. Thus, these baseline methods have not been demonstrated to generalize effectively to real-world settings. In contrast, ConceptScope successfully identifies biases directly within real-world datasets, as demonstrated in Section C.5.

C.4 Qualitative results: examples of discovered known biases

As discussed in the last paragraph of § 5.1, one key advantage of ConceptScope in identifying biases is its ability to pinpoint specific regions of the input image responsible for introducing bias. Fig. 7 illustrates this capability by presenting examples of biases discovered in the Nico++ dataset. Specifically, the figure shows the top five test images labeled as mammals from each group, selected based on high activation of the latent dimension identified as a bias for each class. Each group consistently exhibits a particular background attribute, as visually highlighted by the segmentation masks provided in the bottom row. Since ConceptScope directly associates latents with clear semantic meanings, it eliminates the need to generate captions to determine the semantic labels of the discovered biases.

C.5 Additional Results: Examples of discovered novel biases

In this subsection, we provide additional examples of discovered biases from real-world datasets, supplementing the findings discussed in § 5.2 of the main paper.

ConceptScope identifies diverse biases, including backgrounds, co-occurring objects, and events, in real-world datasets. We illustrate examples of biases discovered by ConceptScope in real-world datasets such as ImageNet and Food101 [1]. Our analysis reveals multiple types of biases, including background biases, co-occurring object biases, and event-related biases. Animal-related classes, for example, often exhibit background biases: birds frequently appear near water, reptiles in terrariums, and insects on leaves. Food-related classes commonly display biases involving co-occurring objects, such as burgers paired with french fries. Additionally, we identify biases arising from specific events or contexts, such as balance beams appearing predominantly in gymnastics competitions and certain dog breeds frequently featured at dog shows.

Implementation details for measuring the impact of bias on model performance. To evaluate how bias concepts within training datasets impact model performance, we compare model accuracy between two groups of images: those with a high presence of bias concepts and those with a low presence. For the ImageNet dataset, we utilize a ResNet50⁸ model with pretrained weights provided by PyTorch. Because the ImageNet validation set contains only 50 images per class, we select the top 10 images per class exhibiting the highest presence of bias concepts and the bottom 10 images per class exhibiting the lowest presence. For the other datasets, we employ a ResNet50 model pretrained with MoCo-v2 [4], following recommendations by Yu et al. [31], who highlight potential test data leakage risks associated with supervised pretraining. For these datasets, we compute the accuracy difference using the top 10% of test samples with high bias presence and the bottom 10% with low bias presence. An example of a bias concept and its impact, measured as the accuracy difference between groups, is illustrated in Fig. 8 and Fig. 4 of the main paper.

⁸<https://docs.pytorch.org/vision/stable/models/resnet.html>

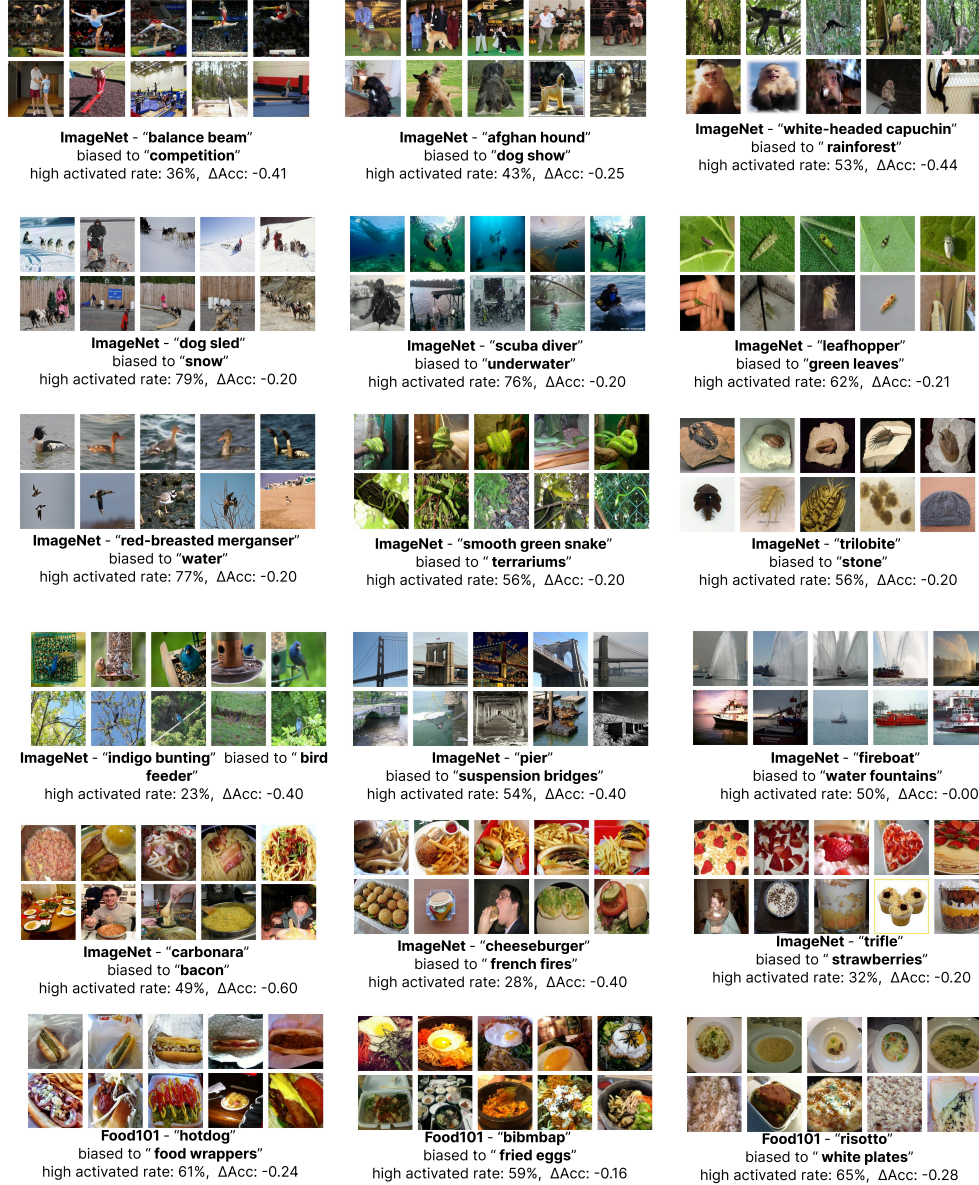


Figure 8: **ConceptScope discovers dataset biases in the wild.** We illustrate examples of biases identified by ConceptScope in the ImageNet and Food101 datasets. For each panel, the top row displays samples exhibiting the identified bias attribute, while the bottom row presents samples without the bias attribute. Additionally, we report the accuracy difference between the images with the highest presence of bias attributes and the images with the lowest presence. We also report the proportion of dataset samples with high activation (above a threshold of 0.5) for the corresponding bias latent.

293 D Experiment Details: Model Robustness Analysis

294 In this section, we provide additional experimental details to supplement §5.3 of the main paper.

295 D.1 Full list of models evaluated on ImageNet

296 The following list includes all models evaluated on the ImageNet and ImageNet-Sketch datasets. The
 297 pretrained weights used for evaluation are available at: <https://docs.pytorch.org/vision/>

298 [stable/models.html](#). For models offering multiple pretrained weight options, we consistently
299 select the ImageNet1K_V1 weights.

- 300 1. AlexNet [11]
- 301 2. VGG11 [26]
- 302 3. VGG16 [26]
- 303 4. VGG19 [26]
- 304 5. ResNet18 [9]
- 305 6. ResNet34 [9]
- 306 7. ResNet50 [9]
- 307 8. ResNet101 [9]
- 308 9. ResNet152 [9]
- 309 10. resnext50 [28]
- 310 11. resnext101 [28]
- 311 12. ViT-B-16 [6]
- 312 13. ViT-B-32 [6]
- 313 14. ViT-L-16 [6]
- 314 15. ViT-L-32 [6]
- 315 16. ConvNeXt-Tiny [17]
- 316 17. ConvNeXt-Base [17]
- 317 18. ConvNeXt-Small [17]
- 318 19. ConvNeXt-Large [17]
- 319 20. EfficientNet-B0 [27]
- 320 21. EfficientNet-B1 [27]
- 321 22. EfficientNet-B2 [27]
- 322 23. EfficientNet-B3 [27]
- 323 24. EfficientNet-B4 [27]
- 324 25. EfficientNet-B5 [27]
- 325 26. EfficientNet-B6 [27]
- 326 27. EfficientNet-B7 [27]
- 327 28. DenseNet121 [10]
- 328 29. DenseNet169 [10]
- 329 30. DenseNet201 [10]
- 330 31. WideResNet50 [32]
- 331 32. WideResNet101 [32]
- 332 33. SwinTransformer-Base [16]
- 333 34. SwinTransformer-Tiny [16]
- 334 35. SwinTransformer-Small [16]

References

- [1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *Computer vision—ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014.
- [2] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. <https://transformer-circuits.pub/2023/monosemantic-features>, 2023. Anthropic, Oct 4, 2023.
- [3] Riddhi Chakraborty, Yinong O Wang, Jialu Gao, Runkai Zheng, Cheng Zhang, and Fernando D De la Torre. Visual data diagnosis and debiasing with concept graphs. *Advances in Neural Information Processing Systems*, 37:106383–106410, 2024.
- [4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.
- [5] Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [7] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960*, 2022.
- [8] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [10] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.
- [11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [12] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.
- [13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.
- [14] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2584–2593. IEEE, 2017.
- [15] Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [16] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [17] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

- 386 [18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In
387 *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.
- 388 [19] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In
389 *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- 390 [20] Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as
391 dataset analyst: Subpopulation structure discovery with large language model. In *European Conference on*
392 *Computer Vision*, pages 235–252. Springer, 2024.
- 393 [21] Badr-Eddine Marani, Mohamed Hanini, Nihitha Malayarukil, Stergios Christodoulidis, Maria
394 Vakalopoulou, and Enzo Ferrante. Vig-bias: Visually grounded bias discovery and mitigation. In *European*
395 *Conference on Computer Vision*, pages 414–429. Springer, 2024.
- 396 [22] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint*
397 *arXiv:2111.09734*, 2021.
- 398 [23] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust Aggregation for Federated Learning.
399 *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. doi: 10.1109/TSP.2022.3153135.
- 400 [24] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural
401 networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint*
402 *arXiv:1911.08731*, 2019.
- 403 [25] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and
404 Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.
- 405 [26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recogni-
406 tion. *arXiv preprint arXiv:1409.1556*, 2014.
- 407 [27] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In
408 *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
- 409 [28] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transforma-
410 tions for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern*
411 *recognition*, pages 1492–1500, 2017.
- 412 [29] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human
413 action recognition by learning bases of action attributes and parts. In *2011 International conference on*
414 *computer vision*, pages 1331–1338. IEEE, 2011.
- 415 [30] Sriram Yenamandra, Pratik Ramesh, Viraj Prabhu, and Judy Hoffman. Facts: First amplify correlations
416 and then slice to discover bias. In *Proceedings of the IEEE/CVF International Conference on Computer*
417 *Vision*, pages 4794–4804, 2023.
- 418 [31] Han Yu, Xingxuan Zhang, Renzhe Xu, Jiashuo Liu, Yue He, and Peng Cui. Rethinking the evaluation
419 protocol of domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
420 *Pattern Recognition*, pages 21897–21908, 2024.
- 421 [32] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*,
422 2016.
- 423 [33] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyang Shen, and Peng Cui. Nico++: Towards better
424 benchmarking for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision*
425 *and pattern recognition*, pages 16036–16047, 2023.