HIDE AND SEEK: HIDE CONCEPTS TO SEEK EXPLANATIONS FOR BLACK BOX MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpreting machine learning model predictions is a central challenge, particularly for black-box vision models. Concept-based attribution methods interpret models using human-understandable concepts. Existing concept-based methods can only interpret classification and regression models. They require access to model weights or large concept-annotated datasets, and most methods cannot provide image-level attribution. We propose Hide and Seek (*HnS*), a new method for concept-level attribution that does not require access to model weights or labeled concept datasets. *HnS* can interpret a wide range of vision models. Given a concept and image set, *HnS* retrieves concept-aligned images using a vision-language model, localizes the concept within each image, masks the localized region, and measures attribution from the change in the black-box model's output between the original and masked images. This enables faithful attribution by directly measuring the model's behavior while also enabling local attributions. We validated *HnS* on CelebA and COCO-Stuff. Compared to existing methods, *HnS* showed better agreement with ground truth attributions. We demonstrated the applicability of *HnS* on diverse tasks, including image captioning and satellite image segmentation, where no existing concept attribution method can be used. We provided our anonymous code in the supplementary material.

1 INTRODUCTION

Interpreting why a model made a certain prediction is important for deploying it responsibly, especially in high-stakes settings such as medicine or hiring. Concept attribution is a popular approach for interpreting the behavior of vision models (Kim et al., 2018; Yeh et al., 2020; Wang et al., 2024; Chen et al., 2020; Yuksekgonul et al., 2023). It measures how much a human understandable concept (e.g., smiling, long hair) contributed to a model's prediction. Suppose we have a classification model M that predicts whether an image contains a male or a female, and M showed a high attribution score for the concept "smiling" for male images. This means that M relies on smiling-related features to predict males. This insight is valuable for model refinement, since we know M has learned to rely on a spurious feature in the dataset to classify males.

To be practically and widely useful, a concept attribution method should be applicable to black-box models (no access to model weights), not only for white-box models (models with publicly accessible weights). Second, it should measure both global attribution (the overall attribution of a concept across all images in a dataset) and local attribution (the attribution of a concept for a single image). Third, it should not require a labeled concept dataset, since curating such datasets is expensive and impractical to do for every application. Fourth, it should be able to interpret a wide range of vision models. No existing method meets all of these criteria.

TCAV (Kim et al., 2018) was one of the first methods to measure concept attribution. Several subsequent works developed extended versions of TCAV (Yeh et al., 2020; Wang et al., 2024; Chen et al., 2020). TCAV and its derivatives only work for white-box models, as they require access to model weights. More recent attribution methods, such as Post-hoc Concept Bottleneck Models (PCBMs) (Yuksekgonul et al., 2023), work for both white-box and black-box models. However, PCBMs require labeled concept datasets, measure only global attribution, and can only interpret classification and regression models. Motivated by these gaps in prior work, we present a new concept attribution method called Hide and Seek or *HnS* (Figure 1).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To understand how Hide and Seek (*HnS*) works, say we want to compute the attribution of the concept "smiling" for male predictions, for model M (discussed in our example earlier). We use a vision-language model (VLM) to retrieve images that best represent the concept (smiling faces). In each of these images, *HnS* automatically identifies the region where the target concept is present (mouth region). It creates a corresponding masked image where that concept is hidden. To compute attribution, *HnS* passes the actual image (a smiling face) and its masked version (mouth region hidden) to the model M. We define concept attribution as the difference in the predicted probability (of the male class) between the original and the masked image. A high attribution score indicates that M relies on smiling as a cue to predict males.

HnS does not need a labeled concept dataset, since it uses a VLM to identify concepts. Since *HnS* uses a model's predicted probability to measure attribution (only forward-pass outputs, not internal representations), it works with both white-box and black-box models. *HnS* measures both global (overall) and local (image-level) attribution. It can interpret a wide range of vision models (e.g., classification, regression, segmentation, captioning). It is also the fastest method to measure attributions (refer to Appendix A).

We validated *HnS* on two distinct types of datasets: CelebA (a dataset of celebrity faces) and COCO-Stuff (an object- and scene-based dataset). *HnS* achieved the highest correlation with ground-truth attributions compared to previous methods such as TCAV and PCBM. We also tested *HnS* in real-world use cases such as image captioning and field boundary segmentation, where existing methods cannot be applied directly. We found that several state-of-the-art captioning models had gender bias (e.g., models generated male-centric captions when images had a tie or a toaster, and female-centric captions when images had a cake or a hair dryer). Using *HnS*, we also investigated why PRUE is the current state-of-the-art model to segment crop fields (Muhawenayo et al., 2026). We found that PRUE has not only learned to identify concepts useful for segmenting cropland (e.g., arable land) but also concepts irrelevant to cropland (e.g., urban areas).

Our contributions are as follows: (1) We propose *HnS*, the first concept attribution method that can interpret any vision model, without a labeled concept dataset. (2) Compared to previous methods (TCAV, PCBM), we show that *HnS* has the best correlation with ground-truth attributions on facial datasets (CelebA) and object-based datasets (COCO-Stuff). (3) We applied *HnS*s on real-world vision tasks like image captioning and field boundary segmentation to show how our method can be used to interpret a wide variety of vision models.

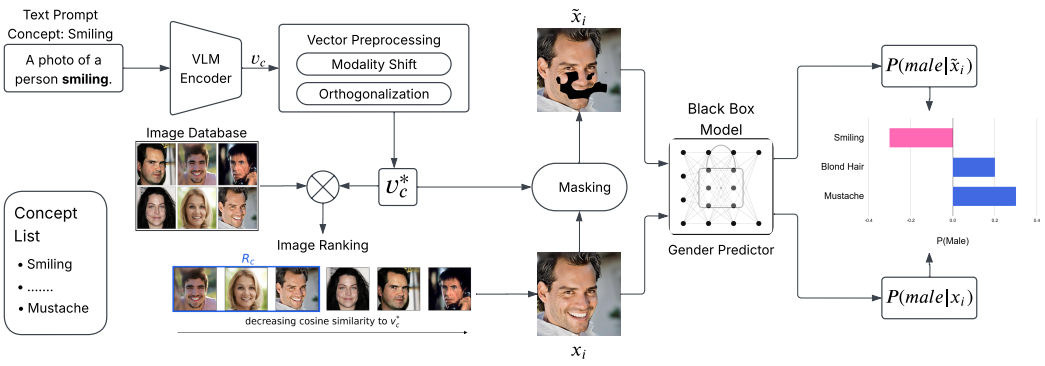


Figure 1: Overview of Hide and Seek (*HnS*). For a concept c (smiling), we encode its text prompt ("photo of a person smiling") into a text vector v_c using SigLIP (a vision-language model). We preprocess v_c (using modality correction and orthogonalization) to obtain v_c^*. From an image database, we select a set of images R_c that show the highest cosine similarity with v_c^*. For each image x_i in R_c, *HnS* identifies the region c in x_i and hides this region to create a masked image $\tilde{x}_i$. We pass x_i and $\tilde{x}_i$ to our model M to obtain predicted probabilities for a class y (male): $P(x_i)$ and $P(\tilde{x}_i)$, respectively. We measure attribution for the concept c (on class y) as $P(x_i)$ - $P(\tilde{x}_i)$.

2 Related Works

Concept-based interpretability methods aim to explain model predictions in terms of human-understandable concepts rather than low-level features like pixels. Previous works proposing con-

cept attribution techniques can be organized into methods for whitebox models (in which the model's internals are accessible), interpretable concept models, and methods for black-box models.

Concept attribution for whitebox models: Kim et al. (2018) introduced TCAV (Testing with Concept Activation Vectors), the first method for concept-level attributions. They start by finding vectors that represent a concept in a model's internal representations. These vectors are known as Concept Activation Vectors (CAV). The method works by finding the sensitivity of the output with respect to the CAV. This sensitivity quantifies the influence of the concept on the model's prediction.

Further work by Wang et al. (2024) and Bai et al. (2023) improved the original TCAV method, making it more robust and better suited for local attribution. Works like Yeh et al. (2020) and Li et al. (2021) combine the ideas of concept-level attribution with Shapley values. These methods have two major practical limitations: they do not work for black-box models, since they require access to the model's internal representation, and they require labeled concept datasets.

Interpretable concept models: Building on TCAV, Chen et al. (2020) introduced "Concept Whitening", a method for training interpretable models by aligning internal representations with disentangled individual concepts. Koh et al. (2020) later proposed Concept Bottleneck Models (CBMs), which decompose prediction into concept prediction followed by target prediction as a linear function of the predicted concepts. Subsequent work, including Kim et al. (2023); Vandenhirtz et al. (2024), improved the reliability and robustness of CBMs. However, these methods train interpretable models rather than explain existing ones.

Concept attribution for black box models: PCBM (Yuksekgonul et al., 2023) works by training a surrogate CBM to mimic the black box predictions. The final-layer weights of the surrogate CBM are considered concept attributions. It leverages CLIP (Radford et al., 2021) to construct concept classifiers from text, avoiding concept annotations. However, CBMs may not faithfully approximate the black-box model and cannot provide local attributions (Tapli et al., 2026). Also, CBMs are designed for classification and regression. They cannot be used to explain models for tasks like segmentation or captioning.

In contrast to prior work, we propose Hide and Seek (*HnS*), a concept attribution method that directly attributes predictions of black-box models without surrogate approximation, does not require concept-labeled datasets, captures local attributions, and supports a broad class of tasks. Table 1 shows a comparison of existing concept attribution techniques with *HnS*.

Table 1: Comparison of Concept Attribution Techniques

Method	Black Box	Concept Label-Free	Surrogate Free	Local Attribution	Wide Task Support
TCAV (Kim et al., 2018)	✗	✗	✓	✓	✗
ConceptSHAP (Yeh et al., 2020)	✗	✗	✓	✓	✗
PACE (Wang et al., 2024)	✗	✗	✓	✓	✗
PCBM (Yuksekgonul et al., 2023)	✓	✓	✗	✗	✗
Hide and Seek (ours)	✓	✓	✓	✓	✓

3 HIDE AND SEEK

3.1 PROBLEM SETUP

Consider a classifier M, which accepts an input image and outputs a predicted probability for a class y (e.g., male). Consider a concept c described using natural-language text (e.g., smiling). We measure concept attribution or $\mathrm{Attr}(c)$: how much does the model M rely on concept c to predict class y? We measure attribution at two granularities: global and local $\mathrm{Attr}(c)$. Global $\mathrm{Attr}(c)$ is the overall influence of concept c on y across all images in a dataset. Local $\mathrm{Attr}(c)$ is the influence of c on y for a particular image.

3.2 HIDE AND SEEK METHODOLOGY

To measure attribution for concept c, we extract its text representation vector $\boldsymbol{v}_c$ using the vision-language model SigLIP. To ensure that $\boldsymbol{v}_c$ accurately represents concept c, we perform additional pre-processing steps on $\boldsymbol{v}_c$ (modality correction and orthogonalization) to get a processed vector $\boldsymbol{v}_c^*$. We discussed these pre-processing steps in Section 3.3. Hide and Seek (*HnS*) works with any VLM.

With SigLIP, *HnS* showed the best correlation with ground-truth attributions. Refer to our ablation with different VLMs in Appendix B.2.

We identify a set of images R_c that best represent c. To extract R_c, we use an image database (e.g., a database of human faces). We compute cosine similarity between $\boldsymbol{v}_c^*$ and each image from the database. We define R_c as the set of the top K images with the highest cosine similarity to $\boldsymbol{v}_c^*$. For our experiments, we set $K = 50$. Refer to our ablation with different K values in Appendix B.3.

To compute the influence (or attribution) of concept c on class y, we assess how model M predicts y when c is visible in an image vs. when c is hidden. Suppose the predicted probability of y changes when c is present compared to being hidden in the image. This means that concept c has a strong influence on class y: the attribution of c is high for y. If the predicted probability of y changes only minimally, c does not influence y: the attribution of c to y is low.

For each image in R_c, we create two versions: the original image x and its masked version $\tilde{x}$, in which we hide concept c in x. To create $\tilde{x}$, we first identify the region in x where c is present. We split the image x into L square patches using SigLIP's encoder. We compute an embedding vector for each patch. We compute cosine similarity between $\boldsymbol{v}_c^*$ and the embedding vector of each of the L patches. Concept c is present in the patches that show high cosine similarity with $\boldsymbol{v}_c^*$. To identify these patches, we use a cosine similarity threshold τ_c.

$$\tau_c = \mu_c + \alpha\,\sigma_c \tag{1}$$

We assume that L cosine similarity scores (for L patches) follow a Gaussian distribution, where μ_c and σ_c are the mean and standard deviation, respectively, of these scores. We set $\alpha = 1.5$ (refer to our ablation with different α values in Appendix B.4). To create $\tilde{x}$, we select patches in x that show a cosine similarity $> \tau_c$. At $\alpha = 1.5$, around 7% of the patches in L show a cosine similarity $> \tau_c$.

To create the masked image $\tilde{x}$, we hide the selected patches using a patch filling strategy. We have 7 filling strategies: blur, zero-fill, white-fill, mean-fill, hue-shift, zero-fill-noise, and noise-then-blur (refer to Appendix C for a description of each strategy). We use only the strategy that best hides c. We offer a variety of strategies, as different strategies are useful for different types of images.

To select the best filling strategy for $\tilde{x}$, we use vector differences. Assume that a filling strategy produces a masked image that effectively hides c. If we subtract the image vector of x (where c is present) from the image vector of this masked image (where c is absent), the resultant vector should predominantly have c's information. The resultant vector should be closely aligned with $\boldsymbol{v}_c^*$. We therefore select a strategy that maximizes the cosine similarity between the resultant vector and $\boldsymbol{v}_c^*$.

$$\tilde{x} = \arg\max_{f \in \mathcal{F}} \left\langle \frac{\phi(x) - \phi(\tilde{x}_f)}{\|\phi(x) - \phi(\tilde{x}_f)\|},\ \boldsymbol{v}_c^* \right\rangle \tag{2}$$

$\langle\rangle$ denotes the cosine similarity operation. $\phi(x)$ is the embedding vector of the original image, and $\phi(\tilde{x}_f)$ is the embedding vector for its corresponding masked image (f is a specific filling strategy). $\frac{\phi(x)-\phi(\tilde{x}_f)}{\|\phi(x)-\phi(\tilde{x}_f)\|}$ is the resultant vector for strategy f.

We pass images x and $\tilde{x}$ to model M. We compute the predicted probability of class y for images x and $\tilde{x}$. We call these predicted probability values $M(x)$ and $M(\tilde{x})$, respectively. We define local attribution of c on y as the difference between these probabilities.

$$\text{Local Attr}(c) = M(x) - M(\tilde{x}) \tag{3}$$

To compute global attribution, we average local attribution values across all K images in our set R_C. We exclude images for which the corresponding masked image is poor (e.g., the entire image is hidden), as they can affect global attribution scores.

$$\text{Global Attr}(c) = \frac{1}{K} \sum_{i}^{R_c} \Big[M(x_i) - M(\tilde{x}_i) \Big] \tag{4}$$

In rare cases, a concept can have noisy local attribution scores (either the concept vector created by SigLIP is poor, or the concept is absent in the image database). We must report a low global attribution score for such concepts, as we cannot reliably measure their influence on our model. To penalize noisy concepts, we perform a Cohen's d normalization step across all concepts. We compute s_c, which is the standard deviation across local attribution scores of c. We normalize the global attribution score with s_c.

$$\text{Global Attr}(c) = \frac{\text{Global Attr}(c)}{s_c} \tag{5}$$

If local attribution scores of c are noisy, s_c will be large, making the normalized global attribution score small for that noisy concept.

To interpret a model with ***HnS***, users should compare the relative rankings of concepts rather than individual attribution values. For example, if a gender classifier assigns a higher global attribution to *smiling* than to *baldness* when predicting *male*, it relies more strongly on smiling. For an image-level explanation, users should rank concepts by their local attribution score for that image.

TCAV and PCBM define concept attribution as $\Delta y/\Delta c$: where Δy is the change in the model output induced by a change Δc in the concept. ***HnS*** instead uses only Δy, as Δc is hard to define in practical use cases. The way existing methods compute Δc for a concept (e.g., cat) is unreliable. They use the c's concept vector: one end of this vector points toward a positive set (cat-like images) and the other end toward a negative set (non-cat images). This makes Δc unstable: if we change the positive or negative set, c's concept vector shifts, and Δc shifts along with it, even though Δy (the model's actual output) has not changed. For stable attribution scores, measuring Δy is sufficient.

3.3 Pre-processing: Modality Correction and Orthogonalization

Modality-shift correction. We would expect an image vector of a concept and its corresponding text vector to be closely aligned. However, in VLMs, these vectors occupy different regions in the vector space (Levi & Gilboa, 2025). Due to this text-image modality gap, we cannot directly compare the text vector of c ($\boldsymbol{v}_c$) to its corresponding image vector. To account for this modality gap, we perform a modality shift correction using the following equation:

$$\boldsymbol{v}'_c = \boldsymbol{v}_c - \mu_{\text{text}}, \qquad \mu_{\text{text}} = \frac{1}{|\mathcal{W}|} \sum_{c' \in \mathcal{R}} \boldsymbol{v}_{c'}, \tag{6}$$

μ_{text} is the mean text embedding computed across a large, diverse word set $\mathcal{W}$ (this word set is independent of the concept set). We give intuition for this correction in Appendix D.

Löwdin orthogonalization. In VLMs, concepts can often be entangled. For instance, VLMs can give similar concept vectors for two different concepts like "baldness" and "male" (due to some inherent bias). To ensure we get a unique concept vector for each concept, we performed orthogonalization using the following equation:

$$V^* = VS^{-1/2}, \tag{7}$$

V is a matrix of all $\boldsymbol{v}'_c$ stacked together. S is the Gram Matrix, given by $S = V^T V$. S expresses the overlap between the vectors in V. The matrix V^* represents the orthogonalized vectors across all concepts. A single row $\boldsymbol{v}^*_c$ represents the orthogonalized vector for the concept c.

Depending on the VLM, these pre-processing steps may be unnecessary or may even introduce noise. In Appendix B.1, we performed ablations on these steps across multiple VLMs.

4 Experiments and Results

4.1 Global Attributions compared to Ground-Truth

4.1.1 Experiment Setup

We compute concept attribution on two datasets: CelebA (a facial dataset) and COCO-stuff (an object-and scene dataset). We compared three attribution techniques: TCAV, PCBM, and Hide and Seek or ***HnS*** (ours). We used three ImageNet-pretrained classification models: ViT-B-16 (Dosovitskiy, 2020), ConvNeXt-Tiny (Liu et al., 2022) and ResNet-18 (He et al., 2015).

We fine-tuned these models on both CelebA and COCO-Stuff. CelebA provides annotations for 40 binary facial attributes (e.g., bald, blond hair). For CelebA, we used 2 of these attributes (Attractive and Male) as classes and fine-tuned the models. From the 38 remaining attributes, we used 26 attributes as our concepts. For COCO-stuff, we fine-tuned models on a subset of 20 classes Singh et al. (2020). For COCO-stuff, we used 103 concepts from Broden (Abid et al., 2022).

To compute ground truth attributions for a concept c (against which each attribution method is compared), we follow the masking strategy of Meister et al. (2023). For each ground-truth image, we created two versions: the original image and a version in which concept c is hidden from the image. We identified the region where c is present in the ground truth and masked that region using a zero-fill strategy. To mask c in the image, we used the segment annotations for our 26 CelebA concepts, from the CelebA-HQ dataset (Lee et al., 2020). Because COCO-Stuff lacks annotations for Broden concepts, we generated segments for Broden concepts (on COCO-stuff images) using Grounded-SAM (Ren et al., 2024). We used these generated segments as the ground truth segments. More details on ground truth generation are available in Appendices E and F.

We passed the original ground-truth image and its masked version to the model. For a target class, we defined the ground-truth local attribution of c as the difference in the predicted probability (of that class) between the ground-truth image and its masked version. To compute ground-truth global attribution of c on a class, we averaged the local attributions across 100 images in which c is present. For each class-concept pair, we computed the Spearman rank correlation (ρ) between ground-truth global attributions and the global attributions provided by each of the three attribution methods. We computed ρ across 52 class-concept pairs in CelebA (26 concepts $\times$ 2 classes) and 2060 pairs in COCO-stuff (103 concepts $\times$ 20 classes).

Our approach to creating ground-truth attributions is similar to *HnS*. One might argue that this similarity can inflate the correlation scores for *HnS*. We addressed this concern with an additional experiment on global attribution in Appendix G.

To create the representative set R_c (for *HnS*), we used CelebA's validation set as the image database for CelebA. We used COCO-Stuff's validation set as the image database for COCO-Stuff. To create the concept dataset for CelebA (for PCBM and TCAV), we used CelebA's training set. For COCO-Stuff, we used the Broden dataset as the concept dataset.

Table 2: For three models, we reported the correlation (ρ) between ground-truth global attribution scores and the global attribution scores from *HnS*, TCAV, and PCBM. A higher ρ value for a method indicates that it shows better agreement with ground-truth attributions.

	CelebA			COCO-Stuff		
Method	ResNet18	ViT	ConvNeXt	ResNet18	ViT	ConvNeXt
TCAV	0.319	0.213	0.192	0.030	0.085	0.117
PCBM (ResNet-18)	0.180	0.164	0.258	0.058	0.018	0.059
PCBM (SigLIP)	0.208	0.197	0.291	0.163	0.211	0.214
PCBM (CLIP-RN50)	0.149	0.142	0.201	0.168	0.199	0.214
HnS (ours)	**0.615**	**0.507**	**0.421**	**0.315**	**0.342**	**0.257**

4.1.2 Global Attribution Results

***HnS* shows the best agreement with global ground truth attributions**. In Table 2, we reported the Spearman rank correlation for all attribution methods (across three models). *HnS* showed the highest correlation across all models for both CelebA and COCO-Stuff.

For *HnS*, we used an image database of in-distribution images (CelebA's validation set for CelebA). In Appendix H, we showed that *HnS* provides accurate attributions for CelebA, even if we use an image database of out-of-distribution images (e.g., UTKFace (Zhang et al., 2017).

4.2 Local Attributions

We visually examine whether concept attribution methods can retrieve sensible images for a class-concept pair. For image retrieval, we only used the two methods that provide local (image-level) attributions: *HnS* and TCAV. We did not use PCBM as it only provides global attributions.

4.2.1 Experiment Setup

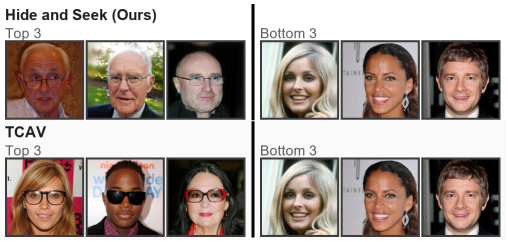
For CelebA, we selected the following class-concept pairs: {class: attractive, concept: eyeglasses}, and {class: attractive, concept: blond hair}. For COCO-stuff: {class: apple, concept: black}, and {class: skateboard, concept: blurry}. For each pair, we computed local attribution scores (using

both *HnS* and TCAV) for all images in which the class was present. For each attribution method, we sorted images (in both CelebA and COCO-stuff) from top to bottom by the magnitude of the local attribution scores. For each class-concept pair, we visually examined the top 3 and bottom 3 images retrieved by *HnS* and TCAV.

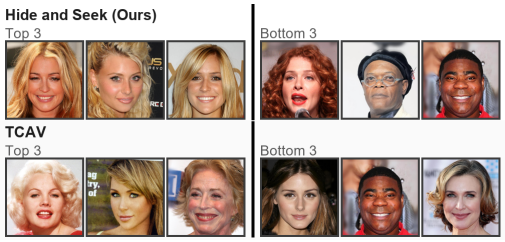
4.2.2 Local Attribution Results

In Figure 2, the top row shows the images retrieved by *HnS*, while the bottom row shows images by TCAV. For both methods, the top 3 images (for a class-concept pair) are in the left grid, while the bottom 3 images are in the right grid.

***HnS* is better than TCAV at image retrieval, even though *HnS* has no access to model weights**. In CelebA, for {class: attractive, concept: eyeglasses} (Figure 2a), the top 3 images of both methods had eyeglasses while the bottom 3 images did not. For {class: attractive, concept: blond hair} (Figure 2b), the top 3 images of both methods had blond hair, while the bottom 3 images did not. On CelebA, both methods retrieved sensible images. In COCO-Stuff, for {class: apple, concept: black} (Figure 2c), all the top 3 images in *HnS* had an apple and some black-colored elements. None of the top 3 images in TCAV had any black elements. For {class: skateboard, concept: blurry} (Figure 2d), all top 3 images in *HnS* had a skateboard and visible motion blur. None of the top 3 images in TCAV had a blurring effect. Only *HnS* retrieved sensible images in COCO-Stuff.



(a) Class: Attractive, Concept: Eyeglasses



(b) Class: Attractive, Concept: Blond Hair



(c) Class: Apple, Concept: Black



(d) Class: Skateboard, Concept: Blurry

Figure 2: For each class-concept pair, we ranked images by magnitude of local attribution scores of Hide and Seek (*HnS*) and TCAV. For each method, we visualized the top 3 and bottom 3 images. (a)-(b) are for CelebA, (c)-(d) are for COCO-Stuff. We show more examples in Appendix I.

5 Hide and Seek in Real World Applications

5.1 Image Captioning

Previous studies (Nair et al., 2026; Tang et al., 2021) have shown that image captioning models exhibit gender bias. To identify gender bias, these studies required annotated datasets (in which gender and objects are annotated in the images). We evaluated whether *HnS* can identify bias trends similar to those reported in prior work, without these annotations.

We computed concept attribution on 10 captioning models: ViT-GPT2 (Mishra et al., 2024), BLIP (Li et al., 2022), Florence-2 (Xiao et al., 2023), LLaVA (Liu et al., 2023b), BakLLaVA (Liu et al., 2023a), GPT-4o (Achiam et al., 2023), FC (Rennie et al., 2017), Att2In (Rennie et al., 2017), Transformer (Vaswani et al., 2017), and UpDown (Anderson et al., 2018). We evaluated these models on the COCO Captions dataset, which contains $10,780$ images of people (male, female) co-occurring with objects (e.g., tie, ball).

Given an input image, each model generates a caption and the predicted probability of each word in the caption. In each caption, we identified male-centric words (e.g., he, him) and female-centric words (e.g., she, her). We used a pre-defined list of these words (see Appendix J for the word list). We summed the predicted probabilities for the male-centric words ($\sum male$) and then for the female-centric words ($\sum female$). We define a gender score: $\sum male - \sum female$. For ***HnS***, we defined the attribution of a concept c as the difference in the gender score when c is present in the image versus when c is hidden. We ignored captions that did not contain any male- or female-centric words. We used the 79 object names in COCO captions as our concept list.

Table 3: Across all models, we identified concepts that frequently occurred in the list of top 20 concepts with the highest positive attribution (male-centric concepts), and in the list of top 20 concepts with the highest negative attribution (female-centric concepts). For instance, tie was a male-centric concept in 9 out of 10 models.

Male-centric concepts	**Number of models**	**Female-centric concepts**	**Number of models**
tie	9/10	cake	7/10
toaster	8/10	hair dryer	6/10
backpack	7/10	teddy bear	6/10
sandwich	7/10	carrot	6/10
bottle	6/10	umbrella	6/10

We identified the concepts that most commonly influenced models to generate male- and female-centric captions. For each model, we ranked 79 concepts (highest to lowest) using ***HnS***'s global attribution scores. We created two lists: (1) male-centric concepts: top 20 concepts in each model with the highest positive attribution score, and (2) female-centric concepts: top 20 concepts with the highest negative attribution score. In Table 3, across the 10 models, we identified the concepts that frequently appeared in the list of male-centric and female-centric concepts.

***HnS* can identify gender biases in captioning models, without using gender or object annotations**. In Table 3, we see several bias trends. For instance, for the tie concept, $9/10$ models showed high positive attribution scores. This indicates that when images had a tie, $9/10$ models generated male-centric captions. For the cake concept, $7/10$ models showed high negative attribution scores. When images had a cake, $7/10$ models generated female-centric captions. Similar gender-bias trends have been shown in prior work (Nair et al., 2026; Tang et al., 2021).

5.2 Field Segmentation

We used ***HnS*** to understand why PRUE is the current state-of-the-art model for segmenting crop fields (Muhawenayo et al., 2026). PRUE accepts two images as input: window A and window B. Window A is the image of a field area (e.g., a field in Kenya) during the planting season (when crops are sown). Window B is the image of the same field area during the harvest season (when crops have fully grown). Using these two images, PRUE produces a semantic segmentation of the field area, in which each pixel is classified as either "field" or "not field". To interpret PRUE, we used crop field images from the Fields of the World Benchmark Kerner et al. (2025).

We used the 19 classes in BigEarthNet (Sumbul et al., 2019) as our concept list (e.g., arable land, marine waters). Since PRUE operates on two images at a time, we hide concept c in three different ways: (1) c is hidden in window A (visible in window B), (2) c is hidden in window B (visible in window A), and (3) c is hidden in both windows A and B. To compute concept attribution, we passed the original image (c is present in both windows A and B) and its three masked versions (c is hidden in either window A, window B, or both) to PRUE. We computed three attribution scores per concept (one for each masked version).

For each version, we computed the attribution score using the difference between the average predicted probability (across all pixels) for the field class when the concept is present and when it is hidden. In Figure 3, for each concept, we reported these three attribution scores: mask window A (blue bar), mask window B (orange bar), and mask both (green bar).

PRUE can not only identify concepts that are useful to segment croplands, but also the concepts that are irrelevant to cropland. PRUE showed high positive attribution scores for concepts such as arable (farmland) and permanent crops, which are important concepts for croplands. PRUE showed

high negative attribution scores for irrelevant concepts like inland waters and urban. We show that these insights are replicated at the country level as well in Appendix K.

PRUE understands that complex cultivation patterns look different in the planting season (window A) and the harvest season (window B). In Figure 3, for most concepts, PRUE showed consistent attribution scores across the three masks: either all attribution scores were non-zero (e.g., inland waters), or all scores were near zero (e.g., marine water). The complex cultivation concept exhibits a different behavior. For the partial masking cases (mask window A and mask window B), the attribution was near zero (blue and orange bars). However, for the full masking case (mask both), the attribution was high (green bar). PRUE has learned that most concepts are static: they should look the same in both windows (e.g., inland waters). If static concepts are masked even in one of the windows, PRUE's outputs are affected. PRUE has also learned that complex cultivation patterns are dynamic: they look different in both windows. If we mask this concept in one of the windows, PRUE's outputs are unaffected.

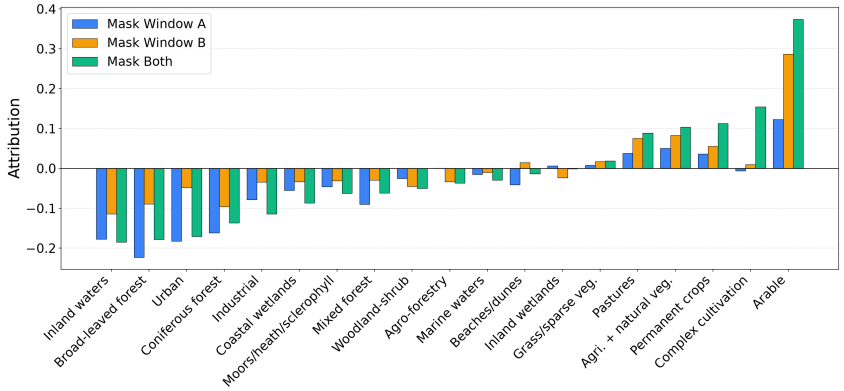


Figure 3: *HnS* attributions for PRUE, a model used to segment crop fields. For each concept c (e.g., inland waters), we reported three attribution scores: (1) mask window A, with a blue bar (c is only hidden in window A), (2) mask window B, with a orange bar (c is only hidden in window B), and (3) mask both, with an green bar (c is hidden in both windows).

6 Limitations

***HnS* cannot always guarantee concept isolation**. To ensure optimal attribution scores from *HnS*, concept vectors should be sufficiently distinct from one another (concept isolation). While concept isolation is largely ensured by our Lödwin orthogonalization step, prior work has shown that VLMs like SigLIP can produce similar vectors for two concepts (Rawlekar et al., 2025; Materzyńska et al., 2022). In such cases, even orthogonalization cannot guarantee concept isolation. Users should be mindful of potential entanglement and avoid supplying similar concepts (e.g., dog and canine).

***HnS* only works for concepts that have spatial meaning.** We cannot compute attribution for abstract concepts (e.g., lighting, happiness) since they cannot be masked.

7 Conclusion

We introduced Hide and Seek (*HnS*), a technique for measuring concept attribution in models. Our results show that *HnS* provides the most reliable concept attribution among existing techniques. *HnS* offers a combination of benefits that no existing technique offers: (1) it does not require a curated concept dataset, (2) it can measure attribution for both white and black-box models, (3) it provides both global and local (image-level) attributions, and (4) it can be used to interpret specialized models such as captioning and segmentation models. Although *HnS* provides more accurate concept attribution scores than previous methods, these scores can be further improved with human input. Users can manually inspect the representative set R_c to remove images that do not represent a concept. Users can also avoid using concept names with similar meanings.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, or interpret results.

Additionally, we used generative AI tools to create or edit software code, create artifacts, summarize or analyze existing literature, brainstorming, sourcing/searching for information, edit a research paper to improve readability, and propose a title or keywords for a research paper.

We have reviewed all AI-assisted work. LLM-generated code and text were verified and tested for correctness by the authors. Furthermore, we verified all existing method details from the actual paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have provided the code for our method and experiments in supplementary materials. We provide parameter details and ablations in Section 3 and Appendix B.

REFERENCES

Abubakar Abid, Mert Yuksekgonul, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 66–88. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/abid22a.html.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6077–6086, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00636. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00636.

Andrew Bai, Chih-Kuan Yeh, Pradeep Ravikumar, YC Lin, and Cho-Jui Hsieh. Concept gradient: Concept-based interpretation without linear assumption. In *International Conference on Learning Representation*. ICLR, 2023.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8730–8738, 2018.

Kaiming He, Xiangyu Zhang, Shaoquing Ren, and Jian Sun. Deep residual learning for image recognition. 2016 ieee conf. In *Comput. Vis. Pattern Recognit*, pp. 770–778, 2015.

Yusuke Hirota, Yuta Nakashima, and Noa Garcia. Quantifying societal bias amplification in image captioning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13440–13449. IEEE, 2022.

Hannah Kerner, Snehal Chaudhari, Aninda Ghosh, Caleb Robinson, Adeel Ahmad, Eddie Choi, Nathan Jacobs, Chris Holmes, Matthias Mohr, Rahul Dodhia, et al. Fields of the world: A machine learning benchmark dataset for global agricultural field boundary segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28151–28159, 2025.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In *International Conference on Machine Learning*, pp. 16521–16540. PMLR, 2023.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5548–5557. IEEE, 2020.

Meir Yossef Levi and Guy Gilboa. The double-ellipsoid geometry of clip. In *International Conference on Machine Learning*, pp. 33999–34019. PMLR, 2025.

Jiahui Li, Kun Kuang, Lin Li, Long Chen, Songyang Zhang, Jian Shao, and Jun Xiao. Instance-wise or class-wise? a tale of neighbor shapley for concept-based explanation. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pp. 3664–3672, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517. doi: 10.1145/3474085.3475337. URL https://doi.org/10.1145/3474085.3475337.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023b.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11966–11976. IEEE, 2022.

Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16389–16398. IEEE, 2022.

Nicole Meister, Dora Zhao, Angelina Wang, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Gender artifacts in visual datasets. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4814–4825. IEEE, 2023.

Swapneel Mishra, Saumya Seth, Shrishti Jain, Vasudev Pant, Jolly Parikh, Rachna Jain, and Sardar M.N. Islam. Image caption generation using vision transformer and gpt architecture. In *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pp. 1–6, 2024. doi: 10.1109/InCACCT61598.2024.10551257.

Gedeon Muhawenayo, Caleb Robinson, Subash Khanal, Zhanpei Fang, Isaac Corley, Alexander Wollam, Tianyi Gao, Leonard Strnad, Ryan Avery, Lyndon Estes, Ana M. Tárano, Nathan Jacobs, and Hannah Kerner. Prue: A practical recipe for field boundary segmentation at scale, 2026. URL https://arxiv.org/abs/2603.27101.

Rahul Nair, Bhanu Tokas, and Hannah Kerner. A woman with a knife or a knife with a woman? measuring directional bias amplification in image captions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 255–264, March 2026.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

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 *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Samyak Rawlekar, Yujun Cai, Yiwei Wang, Ming-Hsuan Yang, and Narendra Ahuja. Efficiently disentangling clip for multi-object perception. *arXiv preprint arXiv:2502.02977*, 2025.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7008–7024, 2017.

Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11067–11075. IEEE, 2020.

Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. *arXiv preprint arXiv:1902.06148*, 2019.

Ruixiang Tang, Mengnan Du, Yuening Li, Zirui Liu, Na Zou, and Xia Hu. Mitigating gender bias in captioning systems. In *Proceedings of the Web Conference 2021*, pp. 633–645, 2021.

Merve Tapli, Quentin Bouniot, Wolfgang Stammer, Zeynep Akata, and Emre Akbas. Rethinking concept bottleneck models: From pitfalls to solutions, 2026. URL https://arxiv.org/abs/2603.05629.

Bhanu Tokas, Rahul Nair, and Hannah Kerner. DPA: A one-stop metric to measure bias amplification in classification datasets. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=efuY2yYC1U.

Moritz Vandenhirtz, Sonia Laguna, Ričards Marcinkevičs, and Julia E Vogt. Stochastic concept bottleneck models. *Advances in Neural Information Processing Systems*, 37:51787–51810, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Hengyi Wang, Shiwei Tan, and Hao Wang. Probabilistic conceptual explainers: Trustworthy conceptual explanations for vision foundation models. In *International Conference on Machine Learning*, pp. 51502–51522. PMLR, 2024.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. *arXiv preprint arXiv:2311.06242*, 2023.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33:20554–20565, 2020.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

A Computational Cost Comparison

Hide and Seek (*HnS*) estimates attribution by directly comparing the black-box model output on original and masked images, so unlike a method that fits a single reusable probe, its cost scales with the number of images scored per concept rather than with the size of any training set. We report wall-clock time at the actual scale used throughout this paper: *HnS* retrieves from the full official-val pool (16,874 images) with $K{=}50$; PCBM's CLIP-concepts variant is trained on the full official CelebA train partitions (162,770 images); TCAV uses the same exemplar, random, and control set sizes as our main results. Each method's procedure is run twice to reflect both tasks used in our experiments (Attractive, Male).

Tables 5 and 4 split each method's cost along a different axis: how much of it is spent purely on a black-box-independent encoder, versus how much requires querying the specific black-box model being explained. This split matters because only the second kind of cost must be paid again if the same images and concepts are used to explain a different black-box model: the encoder-side work carries over unchanged, since it never depends on the model being audited.

By this measure, TCAV is the only method with no reusable component at all: a CAV is fit directly on a specific model's internal activations at a specific layer, so its entire cost is black-box-dependent and must be repeated in full for every new model (405.8s on an RTX-4090, 349.5s on an A100). The other four methods all rely on a backbone that is decoupled from the model being explained, so most of their cost carries over, and this pattern holds consistently across both GPUs: black-box-dependent cost accounts for 3.0% of *HnS*'s total on an RTX-4090 (10.6s of 348.7s) and 2.5% on an A100 (6.0s of 238.0s); 28.0% of PCBM (SigLIP)'s on an RTX-4090 (461.5s of 1648.5s) and 31.7% on an A100 (391.7s of 1236.4s); 38.8% of PCBM (CLIP-RN50)'s on an RTX-4090 (429.6s of 1108.1s) and 40.3% on an A100 (348.1s of 864.7s); and 42.0% of PCBM (ResNet-18)'s on an RTX-4090 (481.6s of 1145.7s) and 36.1% on an A100 (347.3s of 962.5s). Concretely, after explaining one black-box model, explaining a second model on the same images and concepts would cost *HnS* only another 6-11s, depending on hardware, versus roughly 350-480s for the three PCBM variants, and the full cost again from scratch for TCAV.

Table 4: Wall-clock cost with an Nvidia A-100 for different techniques, split into encoder-side cost that is reusable across black-box models and cost that depends on the specific black-box model being explained.

Method	**Reusable (encoder-side)**	**Black-Box-Dependent**	**Total**
Hide and Seek (ours)	**232.0s**	**6.0s**	**238.0s**
TCAV	-	349.5s	349.5s
PCBM (SigLIP)	844.7s	391.7s	1236.4s
PCBM (CLIP)	516.6s	348.1s	864.7s
PCBM (ResNet-18)	615.2s	347.3s	962.5s

Table 5: Wall-clock cost with an Nvidia RTX-4090 for different techniques, split into encoder-side cost that is reusable across black-box models and cost that depends on the specific black-box model being explained.

Method	**Reusable (encoder-side)**	**Black-Box-Dependent**	**Total**
Hide and Seek (ours)	**338.1s**	**10.6s**	**348.7s**
TCAV	-	405.8s	405.8s
PCBM (SigLIP)	1187.0s	461.5s	1648.5s
PCBM (CLIP)	678.5s	429.6s	1108.1s
PCBM (ResNet-18)	664.1s	481.6s	1145.7s

B Component Ablation

B.1 Orthogonalization and Modality Shift

To explore the impact of our vector preprocessing steps (orthogonalization and modality shift correction) across encoders, we ablate them across four encoders (SigLIP, CLIP, OpenCLIP-H and Perception) across two datasets (CelebA and COCO-Stuff) with a Resnet-18 based black box model.

Table 6: Joint ablation of encoder, orthogonalization, and Modality Shift for CelebA dataset with ResNet18 black box classifier.

(a) SigLIP

	Orthogonalize	
Modality Shift	On	Off
On	**0.615**	0.526
Off	0.613	0.525

(b) CLIP

	Orthogonalize	
Modality Shift	On	Off
On	0.237	0.371
Off	0.257	**0.409**

(c) OpenCLIP-H

	Orthogonalize	
Modality Shift	On	Off
On	**0.354**	0.349
Off	0.322	0.186

(d) Perception

	Orthogonalize	
Modality Shift	On	Off
On	0.498	**0.573**
Off	0.544	0.544

Table 7: Joint ablation of encoder, orthogonalization, and Modality Shift for COCO-Stuff dataset with ResNet18 black box classifier.

(a) SigLIP

	Orthogonalize	
Modality Shift	On	Off
On	0.315	0.300
Off	**0.318**	0.274

(b) CLIP

	Orthogonalize	
Modality Shift	On	Off
On	0.322	**0.331**
Off	0.309	0.272

(c) OpenCLIP-H

	Orthogonalize	
Modality Shift	On	Off
On	0.180	**0.197**
Off	0.172	0.156

(d) Perception

	Orthogonalize	
Modality Shift	On	Off
On	0.223	**0.245**
Off	0.226	0.192

Table 6 and Table 7 report, for each encoder, the correlation between our attribution scores and ground truth under all four combinations of orthogonalization and modality shift correction switched on or off.

Neither preprocessing step is uniformly helpful: the best configuration is encoder-specific, and in some cases, dataset-specific as well. On CelebA, SigLIP and OpenCLIP-H reach their best correlation with both orthogonalization and modality shift enabled (0.615 and 0.354, respectively), while CLIP is best with both disabled (0.409) and Perception is best with only modality shift enabled (0.573). On COCO-Stuff, three of the four encoders (CLIP, OpenCLIP-H, and Perception) reach their best correlation with modality shift enabled and orthogonalization disabled (0.331, 0.197, and 0.245), while SigLIP is the outlier, preferring the opposite combination, modality shift disabled and orthogonalization enabled (0.318).

B.2 Encoder Ablation

We summarize the results for cross-encoder ablations by taking the best-performing preprocessing combination for each encoder.

Table 8: Each encoder's own best setting (over the orthogonalize x modality-correction grid)

Dataset	SigLIP	CLIP	OpenCLIP-H	Perception
CelebA	**0.633**	0.409	0.354	0.573
COCO-Stuff	0.318	**0.331**	0.197	0.245

Table 8 shows that the SigLIP encoder clearly outperforms other encoders for the CelebA datasets. In fact, the worst-case SigLIP performance ($\rho = 0.525$ on CelebA) is still competitive with the best-case performance of the next-best encoder, the Perception encoder (0.573). For the COCO-Stuff dataset, CLIP appears to be the best-performing model ($\rho = 0.331$), with SigLIP closely following behind ($\rho = 0.318$).

Given that the trade-offs are minimal in the case of COCO-Stuff, we recommend using SigLIP with orthogonality and demeaning, as it significantly outperforms other methods on CelebA.

B.3 K-Value

The present set size K determines how many images from the retrieval pool are pulled into a concept's positive set via top-K cosine similarity to the concept vector. This choice trades off two competing risks: too small a K gives the concept vector too few supporting images to reliably estimate the concept's attribution, while too large a K pulls in images that are only weakly related to the concept, diluting the current set with noise.

To choose a default value, we sweep $K \in \{15, 30, 50, 75, 100\}$ and measure its effect on attribution quality, reported as Spearman correlation with ground truth on CelebA and COCO-Stuff. Table 9 shows that correlation is highest at $K = 50$ on both datasets, and declines as K moves away from it in either direction (more steeply toward smaller K on COCO-Stuff than toward larger K). The two datasets agree on $K = 50$ as the optimal value, which we adopt as the default.

Table 9: K (retrieval pool size) ablation.

Dataset	K=15	K=30	K=50	K=75	K=100
CelebA	0.603	0.607	**0.615**	0.604	0.594
COCO-Stuff	0.269	0.297	**0.315**	0.311	0.305

B.4 Alpha Value

The concept localization step determines which patches of an image belong to a concept by thresholding each patch's similarity to the concept vector at $\tau_c = \mu_c + \alpha\sigma_c$, where μ_c and σ_c are the mean and standard deviation of similarity scores pooled across the concept's present set. This threshold controls a precision/recall trade-off in the resulting mask: a larger α produces a smaller, more selective mask that isolates only the most strongly concept-aligned patches, while a smaller α produces a larger, more inclusive mask that risks capturing background or context patches unrelated to the concept. Too small a mask under-perturbs the concept and understates its effect, while too large a mask perturbs unrelated content and adds noise to the attribution.

To choose a default value, we sweep α and measure its effect on attribution quality, reported as Spearman correlation with ground truth on CelebA and COCO-Stuff, using a ResNet-18 black-box model and the SigLIP encoder (Figure 4). In both datasets, the correlation is relatively stable across a wide range of α, then declines sharply once α exceeds 1.75, consistent with the threshold becoming so strict that masks shrink to the point where they no longer capture enough of the concept to produce a reliable perturbation. Based on this, we set $\alpha = 1.5$ as the default for real-world use, since it sits comfortably inside the stable region while leaving headroom before the sharp decline.

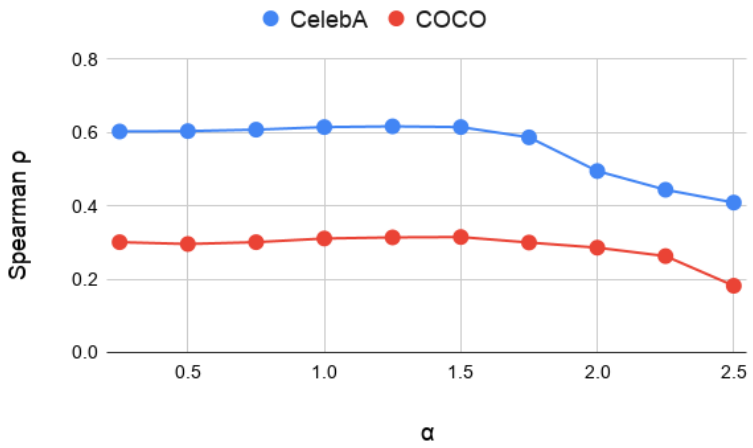


Figure 4: Spearman ρ reported on CelebA and COCO-stuff dataset with ResNet-18 black box model and SigLIP encoder. We observe that for both datasets, the curve remains relatively flat until $\alpha = 1.75$, after which it declines sharply. As such, we suggest keeping the default $\alpha = 1.5$ for real-world usage.

Table 10: Prompt ablation: 3 phrasing templates and ensemble, for *HnS* on CelebA classifier and COCO-Stuff dataset. Class-pooled Spearman ρ is reported against ground truth annotations.

Prompt	CelebA	COCO-Stuff
a photo of concept	0.621	0.308
a photo showing concept	**0.634**	0.311
an image of concept	0.605	0.306
Ensemble	0.615	**0.315**

B.5 Prompt Variation

For VLM-based models, prompt ensembling can help improve zero-shot accuracy (Radford et al., 2021). To test whether similar observations hold for our method, we evaluate 3 prompt variations and their ensemble. All other parameters are kept the same as the global attribution experiment in Section 4.1.

The three prompt templates that we use are: ”a photo of concept”, ”a photo showing concept”, and ”an image of concept”. For CelebA, we rephrase each ”concept” as ”a person with concept” for better clarity. I.e., the concept ”blond hair”, can be templated as ”a photo of a person with blond hair”. For ensembling, we take the average embedding from all three templates.

Table 10 shows that for both datasets, prompt variations do not cause a significant difference in the resulting ρ value compared to other ablated parameters. We also do not see any clear “best” prompting strategy. As such, we defer to existing literature and suggest using prompt ensembling.

C PERTURBATION METHODS

Given a concept's localized binary mask $M_i \in \{0,1\}^{H\times W}$, each perturbation strategy $f \in \mathcal{F}$ first computes a fully-filled version of the *entire* image, $\text{fill}_f(x)$, and then composites it with the original image using M_i as a per-pixel selector:

$$\tilde{x}_f = M_i \odot \text{fill}_f(x) + (1 - M_i) \odot x. \tag{8}$$

Only pixels inside M_i are ever replaced; the rest of the image is untouched. We use seven strategies, chosen to span a range of ways a region can be made concept-uninformative without introducing the same confound every time:

`blur` Gaussian-blurs the *whole* image (σ=20px), then composites the blurred version within M_i. Because the blur is computed over the full image before compositing, a pixel near M_i's boundary can still incorporate some low-frequency content bled in from just outside the mask through the blur kernel – blur destroys fine detail and structure but does not fully erase a region's own coarse color statistics (its mean color typically survives largely intact). This is a common choice in the deletion-metric literature (e.g. RISE (Petsiuk et al., 2018)) specifically because it avoids the out-of-distribution "hole" a constant fill can create, while still meaningfully degrading the region.

`zero_fill` Replaces the region with solid black $(0, 0, 0)$. Erases color and structure completely, at the cost of creating a flat, unnatural patch that can itself be salient to a model independent of the concept being removed.

`white_fill` The complementary constant fill, solid white $(255, 255, 255)$ – included specifically so that concepts whose real color is closest to black (e.g. dark hair, dark plumage) are not the only ones facing maximal fill-color contrast; `white_fill` gives the same maximal-contrast treatment to light-colored concepts.

`mean_fill` Fills with a single constant color corresponding to the ImageNet normalization mean, $(124, 116, 104)$ in 8-bit RGB. A neutral, dataset-standard color chosen to be information-free under the same normalization the encoder itself expects, rather than an arbitrary constant.

`hue_shift` Rotates only the region's hue by 180° in HSV space (each color's exact complementary hue) while leaving saturation and value – i.e. luminance-driven edges, texture, and shape – untouched. This is a surgical, color-only perturbation: unlike `blur` (which leaves much of the true color intact) or `zero_fill`/`white_fill` (which erases color and structure together), `hue_shift` isolates color specifically, letting the perturbation-selection step choose a color-only edit when that is what best matches a concept's own direction, and a structure-destroying edit otherwise. Has no analog for single-band (non-RGB) rasters, since a lone reflectance band has no hue. Therefore, omitted for the FTW/PRUE application.

`zero_fill_noise` Fills with i.i.d. Gaussian noise ($\mathcal{N}(0, 20)$ on the 0-255 scale, clipped to valid range). A flat, constant fill is unlike any real photographic region. Even a genuinely uniform surface has natural sensor and texture noise, so this targets the same confound as `zero_fill` while avoiding an artificially perfect-flat patch.

`noise_then_blur` The same Gaussian noise draw as `zero_fill_noise`, subsequently Gaussian-blurred with the same σ as `blur`. This combines the erasure completeness of noise with the smooth, photographically-plausible appearance of a blurred patch. Thus, it is content-free without either `zero_fill_noise`'s visible high-frequency static or `blur`'s own tendency to leak the original region's low-frequency content.

Multispectral (FTW) analog. For the FTW/PRUE application (Section 3), each of the four Sentinel-2 bands (R, G, B, NIR) is perturbed independently using the same six strategies (all but `hue_shift`). Because raw Sentinel-2 digital numbers do not share RGB's fixed 0-255 scale or ImageNet's fixed mean, the `zero_fill`/ `white_fill`/ `mean_fill` constants and the noise strategies' standard deviation are each derived from that specific band's own observed minimum, maximum, and mean, rather than a fixed constant. Filled values are clipped to the band's valid integer

range before the perturbed raster is written back to its original data type, to avoid integer wraparound (e.g. a negative noise value silently overflowing to a near-maximum value under an unsigned encoding).

D MODALITY SHIFT INTUITION

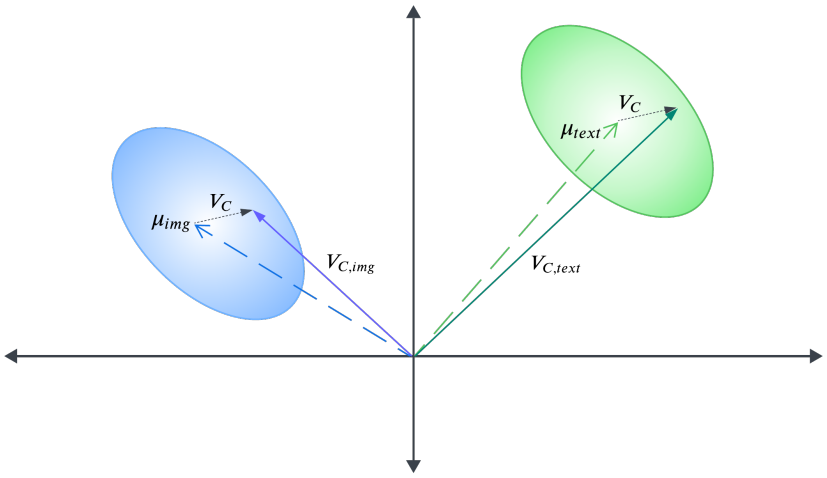


Figure 5: Modality Shift: image embeddings (blue) and text embeddings (green) occupy two separate regions, each offset from the origin by its own center, μ_{img} and μ_{text} respectively, rather than being centered at zero. For a given concept C, the full image embedding $V_{C,img}$ and text embedding $V_{C,text}$ (solid arrows from the origin) each decompose into their ellipsoid's center plus a concept-specific offset V_C (dotted) that is shared between the two modalities. Because $\mu_{img} \neq \mu_{text}$, directly comparing $V_{C,img}$ and $V_{C,text}$ conflates this modality offset with the concept-specific signal we actually want, motivating the modality shift correction.

Levi & Gilboa (2025) demonstrates that in the CLIP embedding space, prior to normalization, the image and text embedding exist in two different ellipsoids that are not centered at the origin, see Figure 5. Let's denote the centers of the image and text ellipsoids by μ_{img} and μ_{text}, respectively. As such, the CLIP embeddings for the text "an image of a dog" can be modeled as

$$V_{dog,text} = \mu_{text} + V_{dog}$$

Similarly, we can model the embeddings for an actual image of a dog as:

$$V_{dog,img} = \mu_{img} + V_{dog}$$

Let's say we have an image database of n images: $\{x_1, x_2, ..., x_n\}$. We can define their image embeddings as $\{V_{x_1,img}, V_{x_2,img}, ..., V_{x_n,img}\}$. This can be further simplified to $\{\mu_{img} + V_{x_1}, \mu_{img} + V_{x_2}, ..., \mu_{img} + V_{x_n}\}$.

To find the representative set R_c for the concept dog, we must compute the cosine similarity $(V_{dog,text} \cdot V_{x_i,img})$ between the image embeddings in the dataset and the text embedding for the phrase "an image of a dog". This dot product expands to four terms:

$$V_{dog,text} \cdot V_{x_i,img} = (\mu_{text} + V_{dog}) \cdot (\mu_{img} + V_{x_i})$$

$$V_{dog,text} \cdot V_{x_i,img} = \underbrace{V_{dog} \cdot V_{x_i}}_{\text{what we want}} + \underbrace{V_{dog} \cdot \mu_{img} + \mu_{text} \cdot \mu_{img} + \mu_{text} \cdot V_{x_i}}_{\text{contamination}}$$

Only the first term, $V_{dog} \cdot V_{x_i}$, reflects whether image x_i actually aligns with the dog-specific direction. The remaining three terms are contamination from the two ellipsoids' offsets.

Modality shift correction aims to removes this contamination. We first estimate μ_{text} empirically, as the average text embedding over a large, diverse set of generic words. We then subtract our estimated $\hat{\mu}_{text}$ from each concept's own text embedding:

$$V_{dog,text} - \hat{\mu}_{text} \approx V_{dog}$$

Using this new approximation of V_{dog}, we compute the similarity as:

$$V_{dog} \cdot V_{x_i,img} = V_{dog} \cdot (\mu_{img} + V_{x_i})$$

$$V_{dog} \cdot V_{x_i,img} = \underbrace{V_{dog} \cdot V_{x_i}}_{\text{what we want}} + \underbrace{V_{dog} \cdot \mu_{img}}_{\text{contamination}}$$

Note that the new contamination term $V_{dog} \cdot \mu_{img}$ is actually a constant term independent of the image x_i. Therefore, it acts as a constant offset in the cosine similarity computed across all images in the image dataset. As such, the relative ranking remains unchanged. This gives us a reliable representative set R_c.

E COCO-STUFF GROUND TRUTH VIA GROUNDED DETECTION AND SEGMENTATION

Unlike CelebA, COCO-Stuff images have no pre-existing per-concept segmentation masks we can reuse to compute ground-truth attribution: knowing that an image contains a "dog" and a "bicycle" does not tell us *where* each one is. To get a real, causal ground-truth signal for this track, we detect and segment each Broden concept directly within COCO's own images, then measure the effect of physically removing it on a trained classifier's predictions.

Concept vocabulary. We use the same 143 active Broden concepts as elsewhere in this paper (the full 170-concept set minus the 27 concepts dropped in Appendix F), so the ground-truth vocabulary stays consistent across tracks.

Detector threshold calibration. Grounding DINO, our open-vocabulary detector, requires a box confidence threshold and a text-matching threshold. Rather than choosing these by inspection, we calibrate them against real ground truth: we take the subset of Broden concepts whose names exactly match one of COCO's 80 official class names, sample 1,000 COCO train-set images containing at least one instance of an overlapping class, and sweep all 16 combinations of box threshold $\in \{0.30, 0.35, 0.40, 0.45\}$ and text threshold $\in \{0.25, 0.30, 0.35, 0.40\}$. For each combination, we run Grounding DINO on all 1,000 images and compare its predicted boxes against COCO's real human-annotated instance boxes for the same concepts, computing mean IoU, recall at IoU ≥ 0.5, and precision at IoU ≥ 0.5. The best combination, box threshold $= 0.30$ and text threshold $= 0.30$ (mean IoU $= 0.6708$), is used for all ground-truth generation.

Detection and deduplication. For each COCO train-set image containing at least one of the 20 target classes, we query Grounding DINO once with all 143 concepts batched into a single caption, returning a bounding box and matched concept name for every detection. Because a single object instance can trigger multiple overlapping boxes, we apply non-maximum suppression (IoU $= 0.5$) separately for each concept, which collapses duplicate detections of the same instance while keeping genuinely distinct instances of that concept in the image.

Segmentation. Each surviving box is passed to the Segment Anything Model (SAM) as a box prompt, producing one instance mask per detection. When a concept appears multiple times in an image, we take the union of moving that concept, meaning we remove every instance of it simultaneously.

Masking and scoring. We compute the original image's class probabilities once per image with our trained ResNet-18 COCO-Stuff classifier, then compare against a masked variant per concept: the unioned SAM instance mask filled black. This gives us Δp and Δlogit (original minus masked) for all 20 target classes.

F ARE BRODEN CONCEPTS RELIABLE

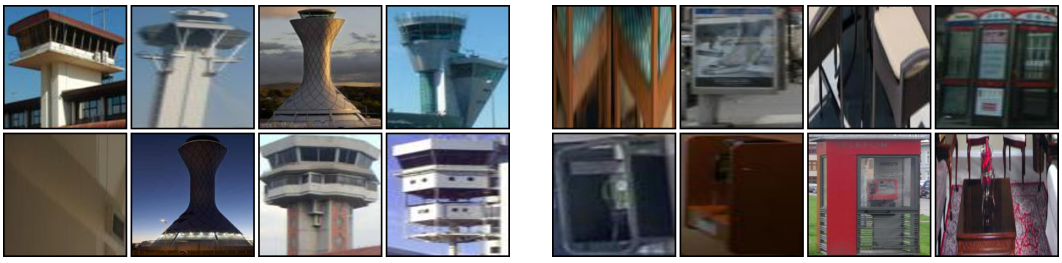
The Broden concept bank we use for COCO-Stuff (Section 4.1) is a repackaged, image-crop version of the original Broden dataset (Fong & Vedaldi, 2018). The concept bank was introduced in (Abid et al., 2022) but is distributed alongside PCBM's own codebase (Yuksekgonul et al., 2023).

Due to inconsistent results in our initial experiments, we wanted to run a sanity check on the Broden dataset. We aimed to identify the Broden concepts that our SigLIP encoder struggled to recognize. For this purpose, we aimed to classify the positive and negative sets for each concept by projecting the images on the concept vector $\boldsymbol{v}_c^*$ in SigLIP embedding space. The intuition being, if SigLIP cannot express the concept, the image representation along the vector $\boldsymbol{v}_c^*$ will not be a good feature to classify the positive and negative set.

But a manual inspection of the worst-performing concepts revealed a different pattern. Instead of SigLIP failing to express the concept, we found that the concept annotations were incorrect. Below, we list the wrongly annotated concepts, along with some example mislabeled positive images for the concept.

Mislabled concepts: *air_conditioner*, *apron*, *awning*, *bathroom_s*, *beak*, *bedroom_s*, *blotchy*, *bumper*, *can*, *candlestick*, *cap*, *clock*, *dining_room_s*, *drinking_glass*, *eye*, *eyebrow*, *faucet*, *figurine*, *handle*, *inside_arm*, *jar*, *knob*, *loudspeaker*, *mouse*, *mouth*, *outside_arm*, and *street_s*.

We exclude these 27 concepts entirely rather than attempting to repair them; the remaining 143 concepts are the ones used throughout Section 4.1.



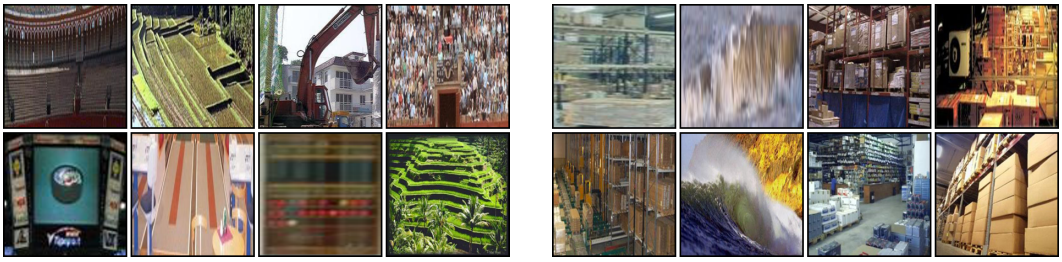
(a) *air conditioner* (b) *apron*

Figure 6: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *awning* (b) *bathroom (scene)*

Figure 7: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *beak* (b) *bedroom (scene)*

Figure 8: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *blotchy* (b) *bumper*

Figure 9: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *can* (b) *candlestick*

Figure 10: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *cap* (b) *clock*

Figure 11: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *dining room (scene)*

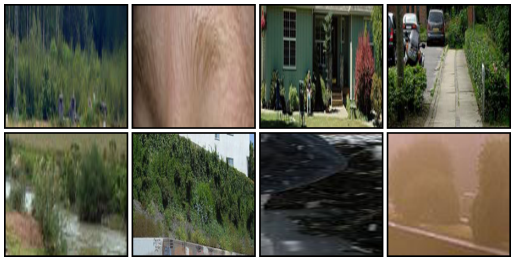


(b) *drinking glass*

Figure 12: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *eye*

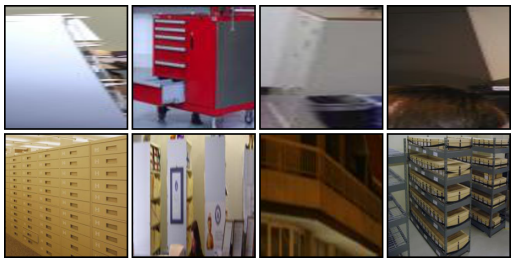


(b) *eyebrow*

Figure 13: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.

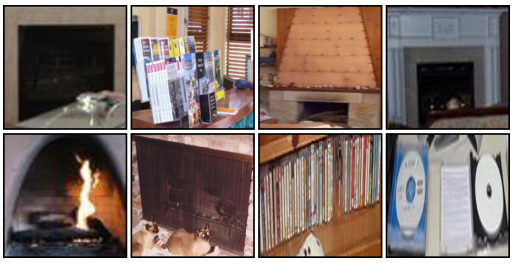


(a) *faucet*



(b) *figurine*

Figure 14: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *handle*



(b) *inside arm*

Figure 15: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *jar* (b) *knob*

Figure 16: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



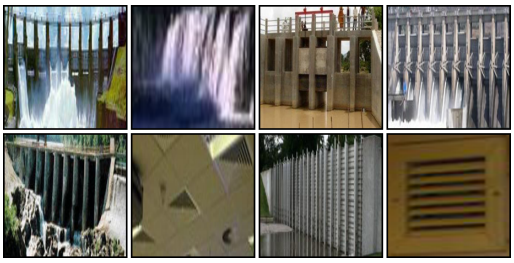
(a) *loudspeaker* (b) *mouse*

Figure 17: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *mouth* (b) *outside arm*

Figure 18: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.



(a) *street (scene)*

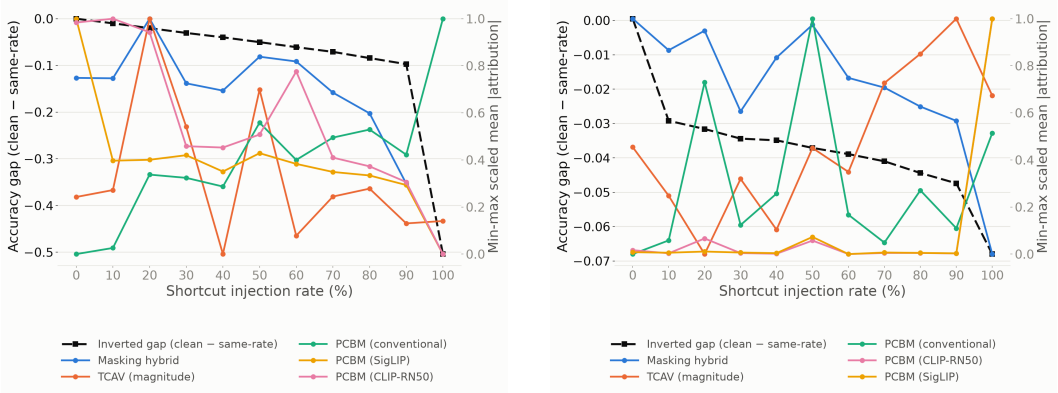
Figure 19: A random sample of images labeled as each concept, shown to illustrate that the labels are unreliable.

G SHORTCUT EXPERIMENT

To test whether an attribution method's output reflects a model's actual reliance on real semantic concepts, rather than being an artifact of the attribution mechanism itself, we construct a controlled shortcut-learning setup similar to the shortcut experiment in Tokas et al. (2025). We inject a synthetic, class-correlated visual cue (a fixed-position solid-color block, one color per class) into a fraction $s \in \{0\%, 10\%, 20\%, ..., 100\%\}$ of each class's training images, and train a separate classifier b_s at each shortcut rate s. As s increases, b_s can increasingly solve the task by relying solely on the injected shortcut rather than on actual image concepts.

We quantify this growing reliance with the accuracy gap $\Delta(s) = \mathrm{acc}_{\mathrm{clean}}(b_s) - \mathrm{acc}_s(b_s)$, the difference between b_s's accuracy on a clean, cue-free validation set and its accuracy on a validation set carrying the *same* injection rate s it was trained on. $\Delta(s)$ starts near zero at s=0% (no shortcut available, so the two accuracies coincide) and grows increasingly negative as s rises, reflecting the classifier's growing preference for the shortcut over actual concepts (black dotted line in Figure 20).

For each b_s, we compute every method's attribution score for the same fixed bank of 26 concepts for CelebA and 103 concepts for COCO-Stuff, and average their absolute magnitudes into a single per-rate summary, $\overline{|\mathrm{Attr}|}(s)$ (min-max normalized across rates in Figure 20 for visual comparability only. The correlations in Table 11 are computed on the raw, unnormalized values, which Pearson correlation is invariant to, regardless. Table 11 reports the Pearson correlation between $\overline{|\mathrm{Attr}|}(s)$ and b_s's own clean accuracy across the 4 shortcut rates: a method that correctly detects declining reliance on real concepts should show a strong *positive* correlation (attribution magnitude falls as clean accuracy falls).



(a) CelebA (b) COCO-Stuff

Figure 20: The accuracy gap $\Delta(s)$ is shown in the black dotted line (left axis, real scale). Average absolute concept attributions for each method are shown on the right-hand axis, min-max-normalized for visual comparability.

Table 11: Pearson correlation between the accuracy gap $\Delta(s)$ and mean absolute concept attributions across the 11 shortcut rates ($s \in \{0\%, 10\%, \ldots, 100\%\}$).

Metric	COCO-Stuff	CelebA
TCAV	-0.42	0.27
PCBM (Resnet-18)	-0.28	-0.84
PCBM (CLIP-RN50)	-0.62	0.71
PCBM (SigLIP)	-0.63	0.66
Hide and Seek (ours)	**0.82**	**0.85**

Results: Hide and Seek (*HnS*) shows a strong, statistically significant positive correlation on both datasets (CelebA: r=0.85, p<0.001; COCO-Stuff: r=0.82, p<0.01), correctly detecting the model's growing reliance on the injected shortcut as a substitute for real concepts in both settings.

TCAV, by contrast, shows no significant relationship on either dataset (r=0.27, p=0.42 on CelebA; $r = -0.42$, p=0.20 on COCO-Stuff) and even flips the sign between them.

For PCBM, the CAV-fitted, whole-image ResNet-18 variant shows a strong *negative* correlation on CelebA ($r = -0.84$, p<0.01) and a weaker, non-significant negative correlation on COCO-Stuff ($r = -0.28$, p=0.40): its mean attribution magnitude spikes at s=100% on CelebA, where the classifier's own clean-set accuracy has collapsed and PCBM's surrogate, produces arbitrarily large weights rather than decline. The two VLM PCBM variants disagree with *themselves* across datasets: CLIP-RN50 is positive on CelebA (r=0.71, p=0.01) but negative on COCO-Stuff ($r = -0.62$, p=0.04); SigLIP shows the same reversal (r=0.66, p=0.03 on CelebA vs. $r = -0.63$, p=0.04 on COCO-Stuff), despite identical construction on a different backbone. Taken together, none of PCBM's three variants reliably tracks a model's actual reliance on real concepts across datasets, unlike *HnS*, which is the only method in this comparison to show a consistent, significant, correctly-signed correlation on both.

H Dataset Ablation

A question users might ask is: Does the retrieval pool need to be drawn from the same dataset, or can we extend it with other datasets? To test this, we re-run *HnS*'s global attribution using two out-of-distribution face datasets, LFW and UTKFace, as the retrieval pool in place of CelebA, keeping the ground truth, the classifier, and every other component of the pipeline unchanged.

Table 12 reports the resulting agreement with ground truth, averaged across the Attractive and Male tasks.

Table 12: Concept attribution agreement with the official-train classifier's masking ground truth, averaged across the Attractive and Male tasks (previously_used ground truth).

Method	Attractive	Male	Average
TCAV	0.478	0.160	0.319
PCBM (ResNet-18)	0.131	0.228	0.180
PCBM (CLIP-RN50)	0.267	0.032	0.149
PCBM (SigLIP)	0.320	0.096	0.208
Hide and Seek (UTKFace)	0.501	0.426	0.463
Hide and Seek (LFW)	0.641	0.502	0.571
Hide and Seek (CelebA)	**0.702**	**0.528**	**0.615**



(a) UTKFace (b) CelebA

Figure 21: Example images from UTKFace (left) and CelebA (right). UTKFace's crops are tight to the face, leaving little to no visible neck or hairline, so concepts like *Bald*, *Necklace*, and *Receding Hairline* are frequently absent from the frame entirely. CelebA's native crops are looser, consistently including hair, neck, and shoulders, so these same concepts remain visible.

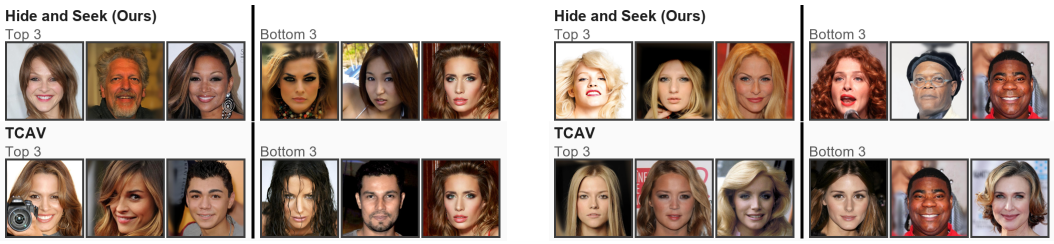
HnS's agreement with the ground truth declines when retrieving from an out-of-distribution pool: the average correlation drops from 0.615 on CelebA to 0.571 on LFW and 0.463 on UTKFace. Even so, both out-of-distribution variants remain clearly ahead of every baseline; specifically, *HnS* (UTKFace) still comfortably outperforms TCAV, the strongest baseline (0.463 vs. 0.319). *HnS*'s

own retrieval pool matters for its absolute performance, but its advantage over existing methods is not an artifact of sharing a dataset with the ground truth.

One thing worth noting is that for UTK Face, the images are closely cropped to the face. The crops are so close in fact that concepts like *Bald*, *Necklace*, *Wearing Hat*, etc., are often not present in the images themselves. As such, when attributing to such concepts, UTK-Face is at a natural disadvantage. Below are some example images from UTK-Face.

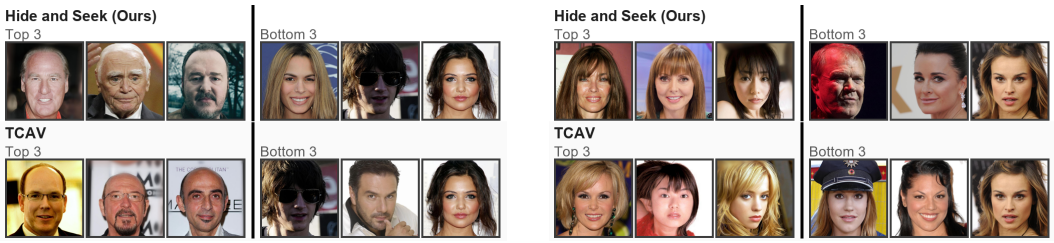
I Local Attribution across Concepts

For a representative sample of concepts, we rank images by |score| and show the three highest-ranked (**Top-3 Present**) alongside the three lowest-ranked (**Top-3 Absent**), for both *HnS* and TCAV, following the format introduced in Section 4.2. We show 10 concepts for CelebA's Male task, and 10 (task class, concept) pairs for COCO-Stuff.



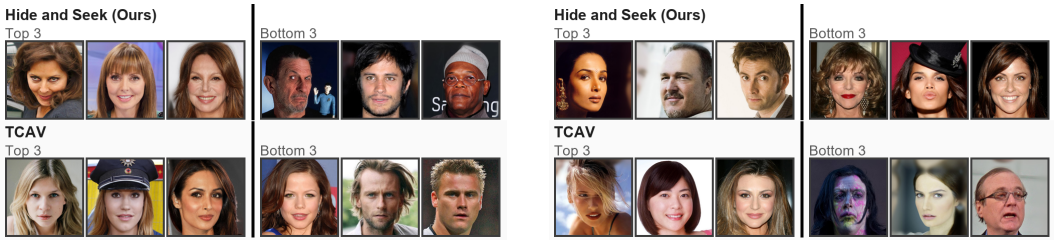
(a) Smiling (b) Blond Hair

Figure 22: Top-3 (highest |score|) and bottom-3 (lowest |score|) Male-task images for HnS and TCAV, for each concept shown.



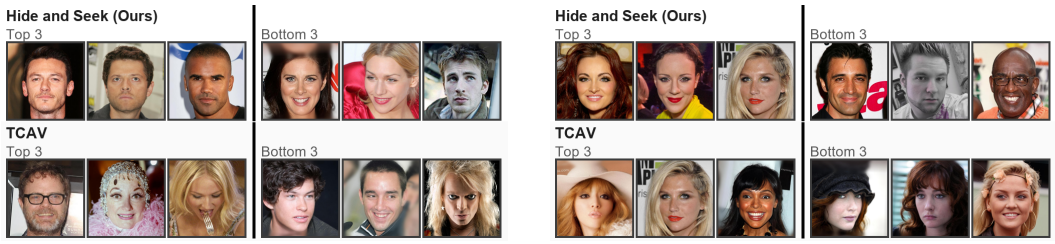
(a) Bald (b) Bangs

Figure 23: Top-3 (highest |score|) and bottom-3 (lowest |score|) Male-task images for HnS and TCAV, for each concept shown.



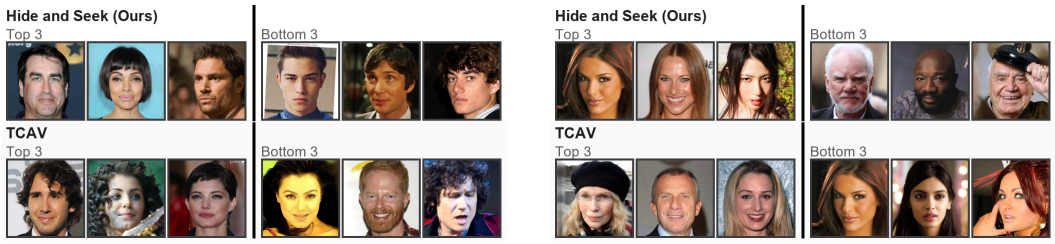
(a) Brown Hair (b) Receding Hairline

Figure 24: Top-3 (highest |score|) and bottom-3 (lowest |score|) Male-task images for *HnS* and TCAV, for each concept shown.



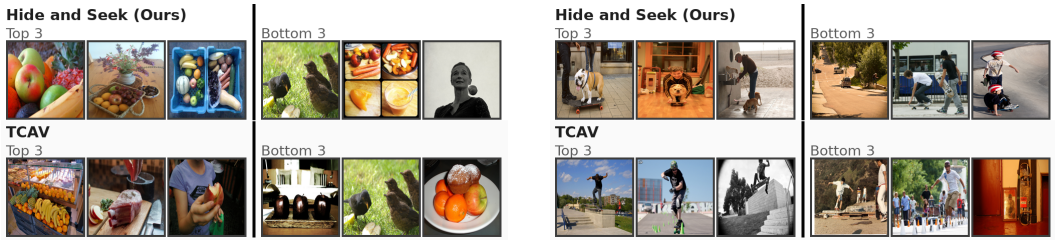
(a) Pointy Nose (b) Wearing Lipstick

Figure 25: Top-3 (highest |score|) and bottom-3 (lowest |score|) Male-task images for *HnS* and TCAV, for each concept shown.



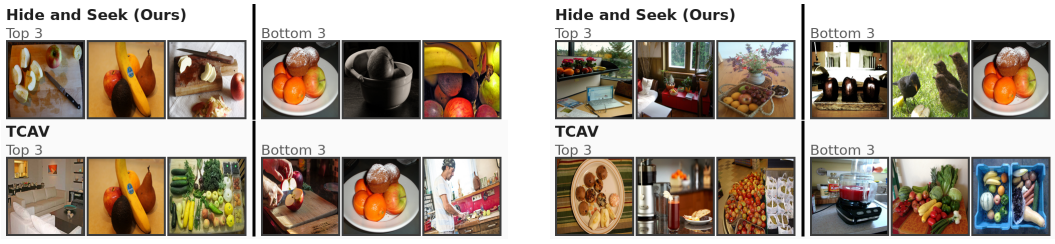
(a) Bushy Eyebrows (b) Straight Hair

Figure 26: Top-3 (highest |score|) and bottom-3 (lowest |score|) Male-task images for *HnS* and TCAV, for each concept shown.



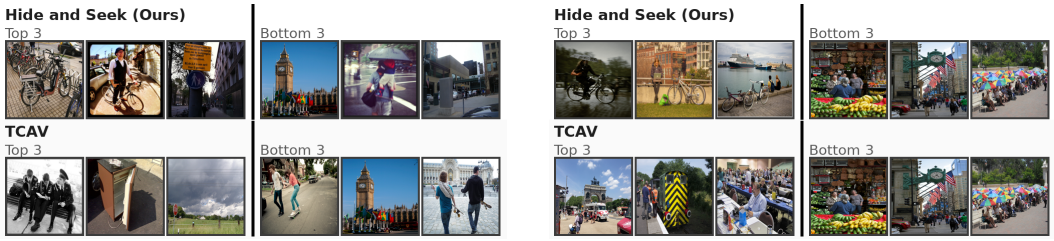
(a) apple / basket (b) skateboard / dog

Figure 27: Top-3 (highest |score|) and bottom-3 (lowest |score|) images for *HnS* and TCAV, for each (task class, concept) pair shown.



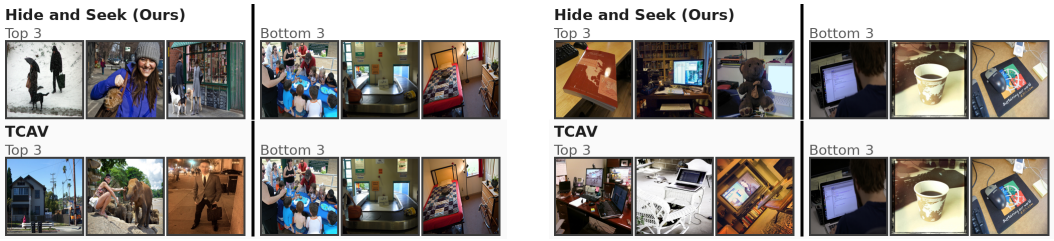
(a) apple / board (b) apple / flowerpot

Figure 28: Top-3 (highest |score|) and bottom-3 (lowest |score|) images for *HnS* and TCAV, for each (task class, concept) pair shown.



(a) car / bicycle (b) handbag / bicycle

Figure 29: Top-3 (highest |score|) and bottom-3 (lowest |score|) images for *HnS* and TCAV, for each (task class, concept) pair shown.



(a) handbag / dog (b) keyboard / book

Figure 30: Top-3 (highest |score|) and bottom-3 (lowest |score|) images for *HnS* and TCAV, for each (task class, concept) pair shown.



(a) skis / blue (b) tennis racket / field

Figure 31: Top-3 (highest |score|) and bottom-3 (lowest |score|) images for *HnS* and TCAV, for each (task class, concept) pair shown.

J Captioning Experiment Details

J.1 Gender Word Set

To score a caption's gender lean without any masking or intervention, we sum each caption model's token probabilities over a fixed set of masculine- and feminine-coded words (Section 5.1). We reuse the gender word lists from (Hirota et al., 2022) rather than defining a new list.

The masculine word list is {man, men, male, father, gentleman, boy, uncle, husband, actor, prince, waiter, he, his, him}, and the corresponding feminine word list is {woman, women, female, mother, lady, girl, aunt, wife, actress, princess, waitress, she, her, hers}.

J.2 Bias Breakdown

The pooled correlation results in Section 5.1 establish that our attribution scores relate to independent bias measures on average, across all 10 caption models and 79 concepts. To make this more concrete, we look at individual concepts: for each concept, we check whether models agree on its attribution sign (masculine- or feminine-leaning), then identify the masculine-leaning and feminine-leaning concepts with the strongest cross-model consensus.

Figure 32 shows the per-model attribution for these two concepts, *tie* (top consensus masculine-leaning) and *cake* (top consensus feminine-leaning). The consensus is strong but not universal: 9 of 10 models agree *tie* is masculine-leaning, with BakLLaVA the lone exception, and its attribution there is close to zero rather than reversed. For *cake*, 7 of 10 models agree on the feminine-leaning direction, while the 3 disagreeing models (LLaVA, BakLLaVA, GPT-4o) all attribute a positive (masculine-leaning) value instead. This pattern, most models agreeing on a concept's gender lean with a small minority disagreeing, rather than the vote being evenly split, is what we would expect if the attribution scores are picking up a real, shared bias across models.

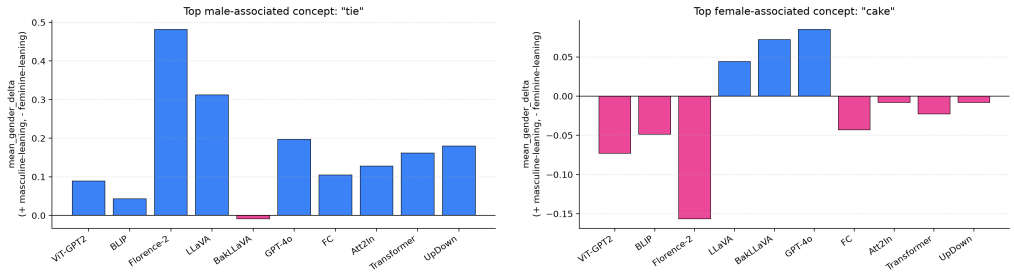


Figure 32: Per-model attribution for the top consensus masculine-leaning concept (*tie*, left) and top consensus feminine-leaning concept (*cake*, right). Bar color shows sign (blue = masculine-leaning, pink = feminine-leaning). 9/10 models agree on *tie*'s masculine direction (BakLLaVA is the lone, near-zero exception); 7/10 models agree on *cake*'s feminine direction, with the 3 disagreeing models (LLaVA, BakLLaVA, GPT-4o) all showing a positive value instead.

K FTW Country Level

The field segmentation experiment (Section 5.2) uses the whole FTW dataset as the retrieval pool. This can hide real variation between sub-populations: a concept the model relies on heavily in one country may matter little in another. To check this, we break the same attribution down by country rather than pooling it.

To ensure diversity in the image database, we select countries with at least 2,000 observations. For each country with at least 2,000 matched dual-window image pairs (Austria, Corsica, Croatia, Denmark, Estonia, Finland, France, Latvia, Lithuania, Netherlands, Slovakia, Slovenia, Spain, and Sweden), we rerun the Concept Mask using the images of a single country as the retrieval pool.

Figure 33 shows the resulting country-by-concept attribution grid. A consistent pattern holds across nearly every country: agricultural concepts (*Arable land*, *Permanent crops*, *Complex cultivation*)

support the field prediction, while built and forested concepts (*Urban fabric*, *Industrial or commercial units*, *Mixed forest*) suppress it. This consistency across 14 countries is itself evidence that Concept Mask's attributions reflect a real property of the model rather than an artifact specific to one country's data.

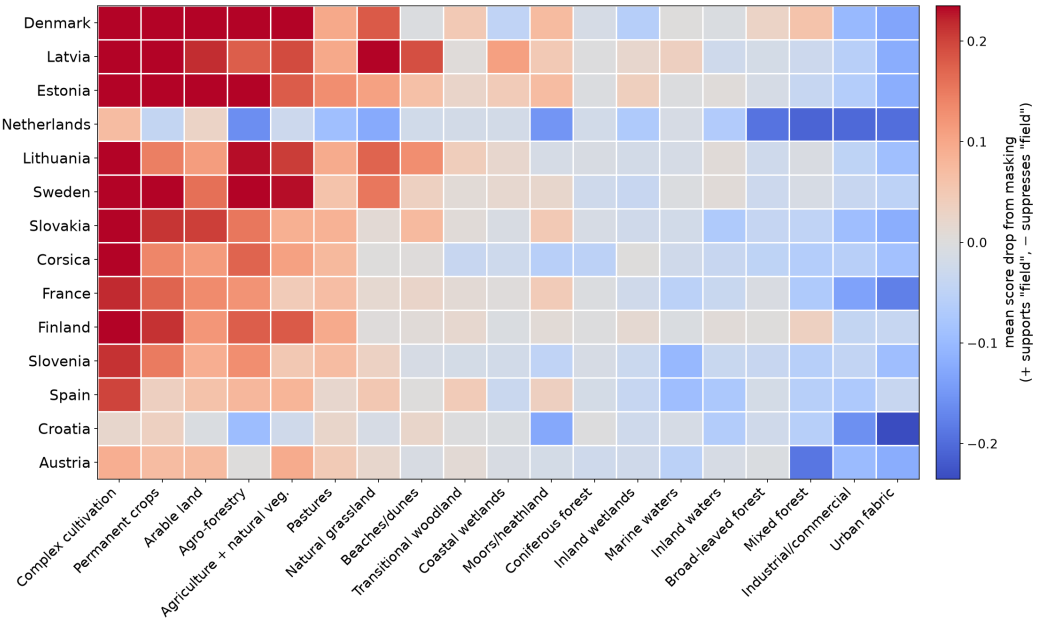


Figure 33: Hide and Seek's land-cover concept attributions for the FTW field-segmentation model, across the 14 countries with sufficient retrieved locations. Each cell is the mean drop in the model's predicted field score when that concept's most-associated region is masked (normalized by 255), colored on a diverging scale: red indicates the concept supports a "field" prediction (e.g., Arable land), blue indicates it suppresses one (e.g., Urban fabric). Concepts are ordered left to right by their mean effect across countries; countries are ordered top to bottom by mean |effect|. The agriculture-supports/urban-suppresses gradient holds consistently across most countries.