

SUB: Benchmarking CBM Generalization via Synthetic Attribute Substitutions

Jessica Bader¹Leander Girrbach¹Stephan Alaniz^{2*}Zeynep Akata¹¹Technical University of Munich, Helmholtz Munich, Munich Center for Machine Learning (MCML)² LTCI, Télécom Paris, Institut Polytechnique de Paris, France

jessica.bader@tum.de

Abstract

Concept Bottleneck Models (CBMs) and other concept-based interpretable models show great promise for making AI applications more transparent, which is essential in fields like medicine. Despite their success, we demonstrate that CBMs struggle to reliably identify the correct concepts under distribution shifts. To assess the robustness of CBMs to concept variations, we introduce SUB: a fine-grained image and concept benchmark containing 38,400 synthetic images based on the CUB dataset. To create SUB, we select a CUB subset of 33 bird classes and 45 concepts to generate images which substitute a specific concept, such as wing color or belly pattern. We introduce a novel Tied Diffusion Guidance (TDG) method to precisely control generated images, where noise sharing for two parallel denoising processes ensures that both the correct bird class and the correct attribute are generated. This novel benchmark enables rigorous evaluation of CBMs and similar interpretable models, contributing to the development of more robust methods. Our code is available at <https://github.com/ExplainableML/sub> and the dataset at <http://huggingface.co/datasets/Jessica-bader/SUB>.

1. Introduction

While deep learning models excel on complex tasks, they are often criticized for lack of transparency in their reasoning, causing a severe bottleneck in the deployment of deep models in real-world contexts. For example, in the medical field, the model’s reasoning must be present in order to be used by physicians. Interpretable models are essential to address these needs. One core method is the Concept Bottleneck Model (CBM) [30], which generates intermediate, interpretable concepts to inform the final prediction.

CBM evaluation exhibits a limitation, as demonstrated in Fig. 1. We expect the CBM to assign the bottom bird the label *yellow crown*, but it still identifies a *blue crown*. In fact, the CBM predicts the exact concept vector associ-

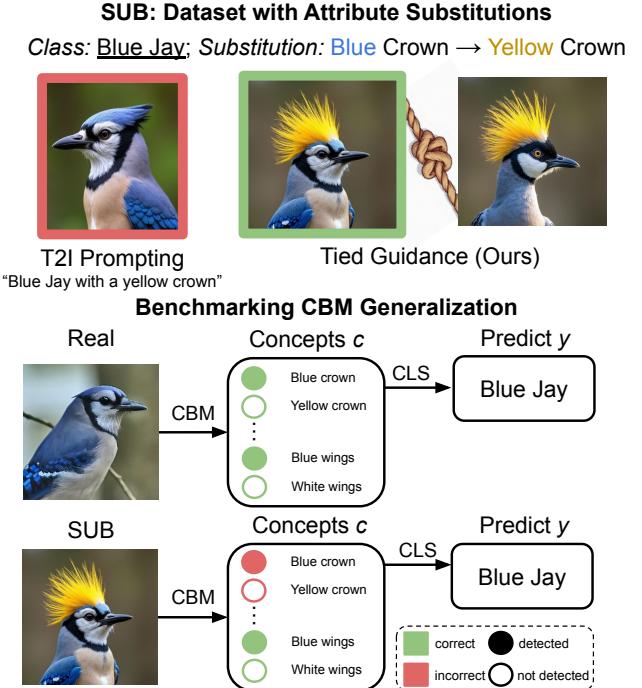


Figure 1. (Top) TGD modifies attributes where prompting fails. (Bottom) The CBM generalizes poorly, memorizing the “Blue Jay” concept vector and mis-classifying the modified concept.

ated with the Blue Jay class (middle), despite clear differences. This observation suggests that the CBM may ground predictions in extraneous factors, potentially entirely unrelated to the visible concepts, rather than those in the image. Ergo, the CBM defaults to predicting the concept vector of the most similar training class. Such behavior raises doubts about the validity of the so-called *concept predictions* as reliable interpretability tools. As Koh et al. [30] evaluated their CBM solely on training classes, we cannot distinguish if the model genuinely learned to identify concepts or simply memorized the concept vectors of the target classes.

In this work, our objective is to evaluate the generalization of concept predictions in CBMs and Vision Lan-

*Work was done at TUM and Helmholtz Munich.

guage Models (VLM) that underpin interpretable architectures, particularly when input images contain novel combinations of known concepts. More specifically, we focus on small deviations from the training classes, with single concept alterations. In this way, we isolate individual attributes, ensuring the model cannot rely on other cues to shortcut predictions. Furthermore, we propose Tied Diffusion Guidance (TDG) to generate these single concept substitutions. As shown in Fig. 1 (top), naively prompting a latent diffusion model (LDM) for a “Blue Jay with a yellow crown” does not yield the desired result. Instead, TDG generates a second image where “yellow crown” appears naturally. By tying the diffusion processes, we are able to successfully substitute the desired attribute on the “Blue Jay” class.

Using TDG, we create our synthetic dataset: Substitutions on Caltech-UCSD Birds-200-2011 (SUB), consisting of 38,400 images for evaluating interpretable models trained on the CUB dataset [63] or with open vocabularies. Leveraging SUB, we find that CBMs and VLMs fail to generalize to novel combinations of known concepts. In particular, we show that a number of CBMs trained with both class- and image-level concept labels, as well as a variety of leading VLMs, cannot reliably detect concepts. This provides strong evidence that these models infer concepts from the predicted class rather than grounding them in the image.

In summary, our contributions are the following: (1) We propose a test-time LDM modification to generate novel combinations of known concepts. (2) We release SUB, an evaluation dataset for interpretable models, which is the first photorealistic image dataset to isolate concepts before evaluating classification. (3) We reveal that existing CBMs and VLMs fail to generalize to new combinations of known concepts, raising concerns about their interpretability.

2. Related Work

In explainable AI, concept-based models have emerged as a powerful interpretability tool, as prototypical parts [3, 5, 40, 43, 52, 53, 62], sparse auto-encoders [11, 28, 35, 50, 60, 70], self-explaining models [2], or Concept Bottleneck Models (CBM) [30, 42, 44, 59, 66]. CBMs in particular are valued for their ability to pre-define key concepts and enable interventions, making them useful in fields like medicine [1, 10, 41]. Since their creation, CBMs have become more flexible by eliminating the need for labeled data [42, 66], facilitating open-vocabulary concept addition and deletion at test time [59], allowing the integration of unsupervised concepts [55], improving intervention success [56], and more.

Nonetheless, follow-up CBM evaluation has revealed that many do not function as intended [23, 34, 37, 48, 57]. Much of this research has focused on information leakage in the pre-defined concepts [21, 34, 36, 37, 68], a phenomenon that has been linked to soft labels [34]. Although

our work does not focus on dataset leakage, these related works have revealed that CBMs tend to focus on incorrect visual cues and are prone to overfitting to irrelevant information. Other CBM analyses have explored their robustness [57], how they respect image locality [48], the impact of concept correlation [23, 49], and their performance on cleaner tasks [19]. Heidemann et al. [23] demonstrated that CBMs struggle with attribute classification, specifically when given highly correlated concepts. Different from the previous work, we evaluate novel combinations of known concepts, showing that concept predictions are not grounded in the image. This may reflect limited training-time attribute combinations, indicating insufficient compositional support for generalization. [65].

Tangentially, image generation models have been gaining attention for impressive generation capabilities [31, 46, 51]. Research has focused on enhancing prompt-following [4, 6, 17, 18, 64] and improving controllability [39, 71, 72]. Efforts to support compositionality have explored the combination of models [13], objects [14, 15, 33], attributes [20], relations [32]. Composed GLIDE [33] favors object co-occurrence (e.g., showing multiple birds) through their noise injection method, compared to TDG which excels at texture edits and attribute manipulation. While CoInD [20] handles attributes, it requires training-time integration, unlike TDG, which adapts at test time.

Alongside the improved capabilities of these models, there is increasing exploration into synthetic data, both for training [12, 16, 22, 29, 54] and evaluation [24, 27, 45]. Previous works have explored the use of synthetic datasets specifically to enhance explainability [23, 24]. Synthetic images enable precise manipulations to isolate individual features, which is particularly valuable for evaluating and enforcing explainability. FunnyBirds [24] consists of imaginary birds created from independent attributes, which are removed and replaced to evaluate model explanations. Similarly, Heidemann et al. [23] tested CBMs using a comparable dataset. Different from the previous works, the images in SUB are far more natural, bringing them much closer to real-world problems. Since our generated images resemble the types of birds found in the widely used CUB dataset [63], they can be leveraged to evaluate existing models trained on bird classes, eliminating the need for specialized training and creating a natural evaluation environment.

3. Tied Diffusion Guidance (TDG)

To build our SUB benchmark, we make fine-grained edits of individual bird attributes from the CUB dataset [63]. While LDMs follow prompts well, they falter on novel attribute combinations. To address this, we propose Tied Diffusion Guidance (TDG), a test-time method that enhances attribute-level control in text-to-image LDMs.

Latent Diffusion Models (LDMs) generate images by

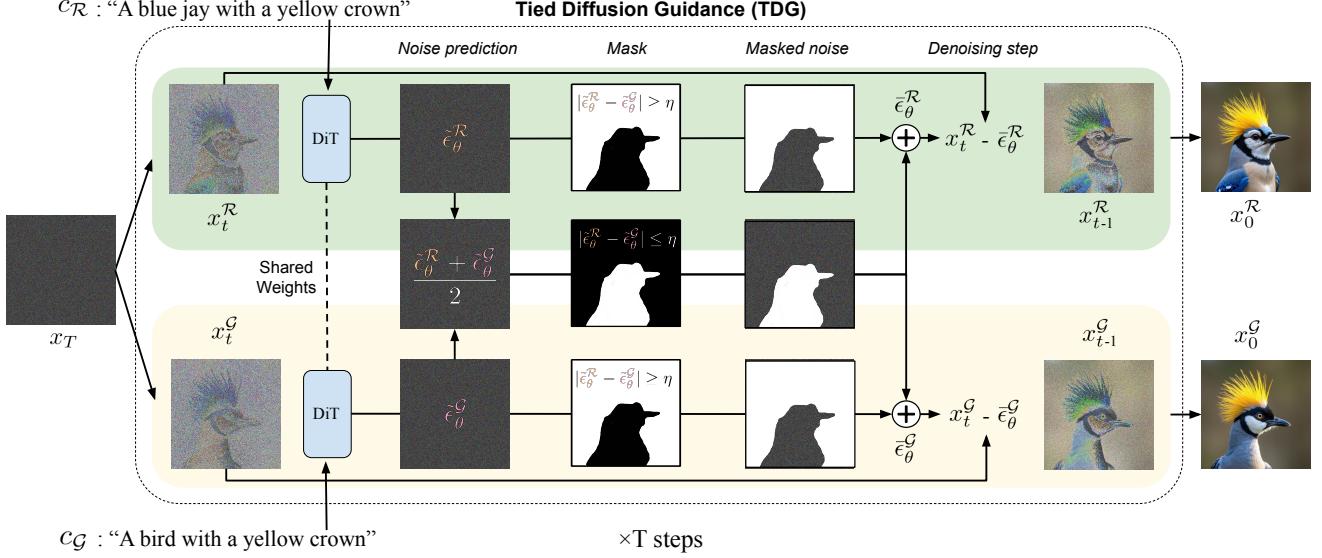


Figure 2. In Tied Diffusion Guidance, two images are generated from related prompts. At each step, the Diffusion Transformer (DiT) predicts the noises $\tilde{\epsilon}_{\theta,t}^R$ and $\tilde{\epsilon}_{\theta,t}^G$ for each image. We compare these predictions and, thresholded by η , retain the original noise where they differ and average the predictions where they are similar. The modified noises $\bar{\epsilon}_{\theta,t}^R$ and $\bar{\epsilon}_{\theta,t}^G$ are subtracted from the images, denoising them. This is repeated for T steps with decreasing η , ensuring the images are highly constrained at the start but independent by the end.

denoising a sample from Gaussian noise x_T to a target image x_0 over T steps. For text-to-image tasks, they model $p(x|c)$, where c is a text prompt. In practice, ϵ_θ is trained to predict the noise in x_t given c at each step t , using the loss:

$$\min_{\theta} \mathbb{E}_{(x,c) \sim \mathcal{D}, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(x_t, c, t)\|_2^2 \right]. \quad (1)$$

Diffusion Guidance. The LDM is trained both with text conditions and unconditionally ($c = \emptyset$). Classifier-free guidance [25] is applied at inference time using:

$$\tilde{\epsilon}_\theta(x_t, c, t) = \epsilon_\theta(x_t, t) + s_g (\epsilon_\theta(x_t, c, t) - \epsilon_\theta(x_t, t)) \quad (2)$$

where guidance scale s_g controls prompt condition strength. While text-to-image models generally follow prompts well, they often fail at zero-shot compositions, for example, ignoring the “yellow crown” edit in favor of a typical “Blue Jay” (Fig. 1, top left). As prompts alone do not capture all SUB edits, we propose a test-time adaptation with semantic guidance from a reference image for attribute modifications.

Tied Diffusion Guidance (TDG). Our goal is to generate an image of a *reference* class \mathcal{R} with an *attribute substitution* \mathcal{S} , replacing the original attribute \mathcal{S}^- (e.g., blue crown) with a target attribute \mathcal{S}^+ (e.g., yellow crown), while preserving \mathcal{R} ’s remaining attributes. To overcome LDMs’ struggles with zero-shot composition, we introduce guidance from a *guidance* class \mathcal{G} , where the target attribute \mathcal{S}^+ is in-distribution and easier to generate. As shown in Fig. 2, we propose to tie the generation of the two images, using \mathcal{G} to guide \mathcal{R} to have one attribute substituted according to \mathcal{S} .

To achieve single-attribute substitution, we generate paired images with separate prompts, c_R and c_G , related to \mathcal{R} and \mathcal{G} with target attribute \mathcal{S}^+ , respectively. We start from the same noise $x_T^R = x_T^G$, and tie the noise predictions element-wise. Given two independent noise predictions $\tilde{\epsilon}^{(1)}$ and $\tilde{\epsilon}^{(2)}$, we apply

$$\mu(\tilde{\epsilon}^{(1)}, \tilde{\epsilon}^{(2)}, \eta)_i = \begin{cases} \frac{\tilde{\epsilon}_i^{(1)} + \tilde{\epsilon}_i^{(2)}}{2} & \text{where } |\tilde{\epsilon}_i^{(1)} - \tilde{\epsilon}_i^{(2)}| \leq \eta^{\text{th}} \text{ percentile} \\ \tilde{\epsilon}_i^{(1)} & \text{otherwise} \end{cases} \quad (3)$$

to obtain the mean noise prediction for all elements i (i.e. image pixels) where the prediction difference is below the η^{th} percentile and keep the noise prediction $\tilde{\epsilon}^{(1)}$ otherwise. We define an η schedule that begins with the two predictions strongly tied and is loosened towards the end of generation:

$$\eta(t, t_{\min}, t_{\max}, k) = \begin{cases} 1 & \text{if } t > t_{\max} \\ \left(\frac{t - t_{\min}}{t_{\max} - t_{\min}} \right)^k & \text{if } t_{\min} \leq t \leq t_{\max} \\ 0 & \text{if } t < t_{\min} \end{cases} \quad (4)$$

where t_{\max} controls the length of the initial strict noise tying phase, and k regulates the transition to independent generation (from t_{\min} onwards).

We apply noise tying symmetrically to both images and update their individual noise predictions using:

$$\bar{\epsilon}_{\theta,t}^R = \mu(\tilde{\epsilon}_{\theta,t}^R, \tilde{\epsilon}_{\theta,t}^G, \eta(t, t_{\min}, t_{\max}, k)) \quad (5)$$

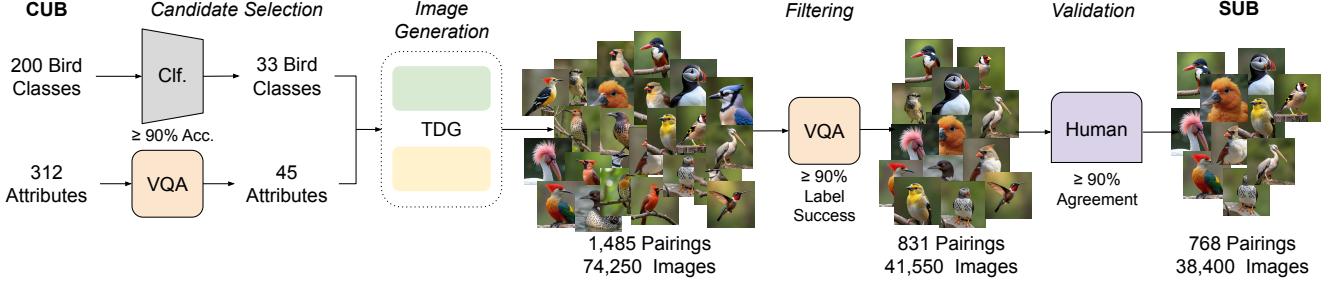


Figure 3. Meticulous filtering ensures that SUB is faithful to the target classes and attributes. Starting with the CUB label set [63], we retain only the best images from the most consistent and detectable bird-attribute pairings.

$$\epsilon_{\theta,t}^{\mathcal{G}} = \mu(\tilde{\epsilon}_{\theta,t}^{\mathcal{G}}, \tilde{\epsilon}_{\theta,t}^{\mathcal{R}}, \eta(t, t_{\min}, t_{\max}, k)) \quad (6)$$

where $\tilde{\epsilon}_{\theta,t}^{\mathcal{R}} := \tilde{\epsilon}_{\theta}(x_t^{\mathcal{R}}, c_{\mathcal{R}}, t)$ and $\tilde{\epsilon}_{\theta,t}^{\mathcal{G}} := \tilde{\epsilon}_{\theta}(x_t^{\mathcal{G}}, c_{\mathcal{G}}, t)$. In summary, TDG ties noise predictions for similar pixels but allows divergence where prompt guidance differs. During generation, we gradually loosen the constraint, ultimately generating the images independently with $c_{\mathcal{R}}$ and $c_{\mathcal{G}}$, and ultimately discarding the guide image.

4. SUB Dataset

We introduce SUB, a benchmark with fine-grained attribute edits to evaluate concept prediction faithfulness. SUB builds upon CUB [63], comprising 38,400 synthesized images of bird classes with substituted attributes generated with TDG. It consists of 768 unique combinations of CUB species \mathcal{R} and target attributes \mathcal{S}^+ , with 50 images per combination. The creation process is detailed in this section and visualized in Fig. 3.

CUB Preliminaries. The CUB dataset [63] consists of 11,788 images of 200 bird species, annotated with attributes about bird parts and their properties. It is commonly used for fine-grained classification and model explainability. CUB includes 28 attribute groupings \mathcal{A} (e.g., leg color, belly pattern), where each image is labeled with 312 binary values representing the presence or absence of individual attributes (e.g., black leg, spotted belly).

4.1. Prompts

We generate synthetic images using our novel TDG (described in Section 3) with the state-of-the-art (SOTA) text-to-image DM FLUX.1-dev [31]. Attribute substitutions \mathcal{S} are chosen from three categories: color, shape, and pattern, where a target attribute $\mathcal{S}^+ \in \mathcal{A}_{\mathcal{S}} \setminus \{\mathcal{S}^-\}$ is selected to differ from the reference bird. As the three categories vary in difficulty, prompts are adjusted to achieve the desired modification and avoid attribute ambiguity, requiring a minimal amount of human intervention. The full details on reference birds \mathcal{R} , guidance birds \mathcal{G} , substitutions \mathcal{S} , and example prompts $c_{\mathcal{R}}, c_{\mathcal{G}}$ are found in Appendices A, B, and D.

Color is the easiest substitution for FLUX, allowing the generic term $\mathcal{G} = \text{"bird"}$. We use $c_{\mathcal{R}} = \text{"a photo of a } \{\mathcal{R}\} \text{ with a } \{\mathcal{S}^+\}"$ and $c_{\mathcal{G}} = \text{"a photo of a bird with a } \{\mathcal{S}^+\}"$.

Shape is of medium modification difficulty, hence we explicitly specify the guide bird (e.g. $\mathcal{G}_{\text{cone beak}} = \text{"song sparrow"}$). We discourage excessive modifications by encouraging the retention of the reference bird’s body shape, using $c_{\mathcal{R}} = \text{"a photo of a } \{\mathcal{R}\} \text{ with the body of a } \{\mathcal{R}\} \text{ and a beak like a } \{\mathcal{G}_{\mathcal{S}^+}\}"$ and $c_{\mathcal{G}} = \text{"a photo of a } \{\mathcal{G}_{\mathcal{S}^+}\} \text{ with a } \{\mathcal{S}^+\}"$.

Texture is the toughest substitution, also requiring a specific guidance bird $\mathcal{G}_{\mathcal{S}^+}$ per target attribute \mathcal{S}^+ . \mathcal{S}^+ is included in $c_{\mathcal{G}}$ to ensure \mathcal{S}^- from \mathcal{R} is changed to match \mathcal{S}^+ from \mathcal{G} , rather than vice versa. We use $c_{\mathcal{R}} = \text{"a photo of a } \{\mathcal{R}\} \text{ with a } \{\mathcal{S}^+\} \text{ like a } \{\mathcal{G}_{\mathcal{S}^+}\}"$ and $c_{\mathcal{G}} = \text{"a photo of a } \{\mathcal{G}_{\mathcal{S}^+}\} \text{ with a } \{\mathcal{S}^+\}"$.

4.2. Data Filtering

We use a filtering mechanism, common in synthetic dataset creation [16, 22], to ensure the faithful representation of manipulated attributes. First, we identify suitable candidates for reference birds and attribute substitutions and generate N images for each bird-attribute pairing. Next, we use an automatic visual-question-answering (VQA) evaluation, together with a human validation step to identify and remove images that did not modify \mathcal{S}^+ correctly or deviate from the reference bird \mathcal{R} . This process helps quantify the added variety and reliability of our attribute modifications.

Candidate Selection. We start by identifying suitable reference birds and attributes. Reference birds must be reliably depicted by the generative model. Hence, we generate 20 images per CUB class with FLUX [31] and measure per-class classification accuracy with a CUB pre-trained model¹ which achieves 88% accuracy on the CUB test set. We select the 33 classes for which the classifier achieves 100% accuracy on the listed in Appendix A).

For attribute candidates, we evaluate a VQA model’s ability to identify them by creating questions for the 28 attribute groups \mathcal{A}_i with the prompt

¹<https://huggingface.co/Emiel/cub-200-bird-classifier-swin>



Figure 4. Images generated with TDG (green) are high-quality and more faithfully represent both the reference bird \mathcal{R} and target attribute substitution \mathcal{S} than those generated with prompting alone (red). TDG generates an additional guidance image x^G , which we discard.

What type of $\{\mathcal{A}_i\}$ does this bird have?

Please pick between

- A) $\{a_1\}$
- A) $\{a_2\}$
- C) $\{\dots\}$
- D) Other

where $\{\mathcal{A}_i\}$ corresponds to the i -th attribute group (e.g. “eye color”) and the options $a_j \in \mathcal{A}_i$ contain all manifestations of that attribute group (e.g. “red”, “black”, etc). We also include “Other”, for when none of the attributes match. As model, we use InternVL-2.5-8B [7–9], which yields the best performance-efficiency tradeoff. We run the VQA evaluation on the full CUB dataset and choose the 45 attributes that obtain accuracy $\geq 90\%$ when optimizing the answer probability threshold between 60% and 95%.

VQA Filtering. We construct substitutions S by pairing attribute candidates \mathcal{S}^+ with bird candidates \mathcal{R} that do not already have this attribute ($\mathcal{S}^+ \neq \mathcal{S}^-$), resulting in 1,485 bird-attribute combinations. We generate images for these combinations using TDG and verify the correctness of \mathcal{S}^+ with a VQA model (InternVL-2.5-8B) using the same prompt as attribute candidate selection. Images with incorrect attribute predictions are discarded, and we rank those remaining by target answer confidence (i.e., probability). We retain only the top 10% of images, corresponding to 50

out of 500 images generated per pairing. If filtering retains less than 10%, the pairing is eliminated. After this stage, 831 bird-attribute pairings remain for the SUB dataset.

4.3. Human Validation

To assess the quality of our synthetic images, we conduct a human study to verify the per-attribute VQA accuracy and the faithfulness of attribute modifications \mathcal{S}^+ to the reference bird classes \mathcal{R} . Human annotators evaluate 40 randomly selected images per attribute, answering whether the attribute modification was faithfully applied and whether the generated bird deviates significantly from the reference bird. For both questions, there are three possible responses: “yes”, “somewhat”, and “no” (the human annotation interface is shown in Appendix E). We discard candidate attributes with $< 90\%$ accuracy across all reference birds, based on annotators choosing “yes”. Additionally, we remove individual bird-attribute pairings with low attribute accuracy, excessive modifications beyond the target attribute, or where the reference birds already displayed the target attribute ($\mathcal{S}^+ = \mathcal{S}^-$). The resulting SUB dataset consists of 33 CUB bird classes, 15 attribute groupings with 32 unique target attributes, and 768 unique bird-attribute pairings, totaling 38,400 images (50 images per pairing).

5. Experiments

In this section, we first examine the quality of our SUB dataset, followed by an evaluation of CBM models on SUB to assess their performance independently of the reference bird’s appearance. SUB’s images are generated with FLUX.1-dev [31] with default guidance scale 3.5. For TDG, we use $k = 10$, $t_{\min} = 0.2$, $t_{\max} = 0.6$ for pattern attributes, and $t_{\max} = 0.9$ for shape and color attributes. These values were selected by looking at 3-5 birds across 2-4 specific attributes for each category (color, texture, shape). We generated 50 images per pair for all 33×45 bird-attribute pairs, and checked that 5+ images per pair for 10+ birds were suitable on 2-3 attributes. Optimizing the hyperparameters across the full pipeline would be too costly.

5.1. Qualitative TDG Examples from SUB

Figure 4 illustrates eight examples generated with TDG. For each image, we include the real reference bird \mathcal{R} , and a sample generated by prompting FLUX [31] out-of-the-box to produce the target substitution \mathcal{S}^+ , using the same prompt as the reference image $x^{\mathcal{R}}$ from TDG (exact prompts are described in Section 4.1). Finally, we present both the reference image $x^{\mathcal{R}}$ from SUB, and the corresponding guidance image $x^{\mathcal{G}}$. From these examples, we observe that TDG is capable of generating realistic attribute-modified images. TDG reliably alters the target attribute’s shape, color, or texture while maintaining the overall recognizability of the reference bird, as seen in the top-left image of the Cardinal with a needle-shaped beak. While standard diffusion generates simple modifications successfully, like changing the White Pelican’s crown to pink (bottom left), TDG is capable of modifying more challenging attributes such as texture (second row). Interestingly, although the resulting guidance bird image $x^{\mathcal{G}}$ is usually visually distinct from $x^{\mathcal{R}}$, it sometimes converges to the same image (bottom left, third row on the right). Despite this, TDG remains essential for applying target substitutions.

5.2. VQA and Human Filtering Results

VQA filtering eliminates images where TDG substituted improperly. In Fig. 5, we present results for errors flagged by our filtering method and images that passed. InternVL-2.5-8B filters images where the target attribute is not visible (top left), the wrong attribute is modified (top right), or the model does not make meaningful modifications (bottom left). Through human verification, we further find and remove bird-attribute combinations where the VQA model is unreliable. The images contained in SUB (green) are high quality and consistently show the target attribute.

5.3. SUB vs. CUB Label Correctness

Although widely used, CUB [63] has faced criticism for labeling errors [26]. To address inconsistent labeling, CBMs

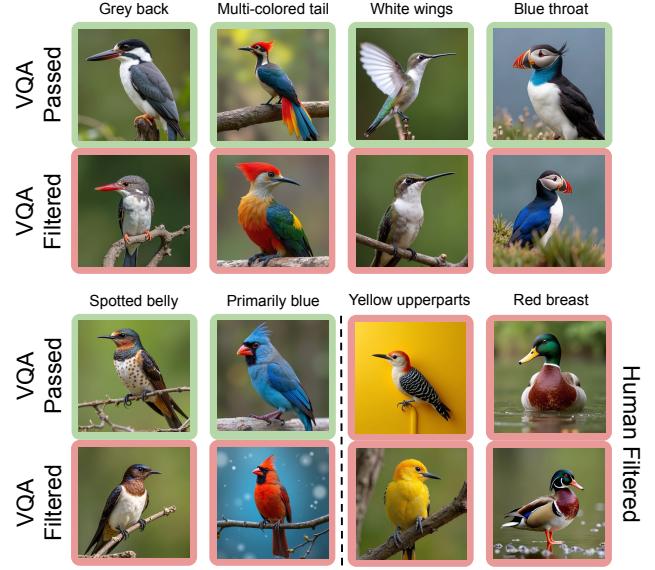


Figure 5. Many failure cases are flagged by our VQA filtering (red), leaving well-modified images in SUB (green).

are trained with per-class attribute vectors determined by majority voting [30]. We ensure that SUB does not inherit these inconsistencies by evaluating attribute quality with human annotations. For CUB, we group images by the presence of an attribute label in the per-class attribute vector and calculate the ratio of images where the per-image label matches (i.e. individual annotations align with the class concepts). For SUB, we calculate average score (“yes” = 1, “somewhat” = 0.5, “no” = 0) per attribute across the images in the human validation study (see Sec. 4.2). The results for 17 attributes (overlap of 32 SUB attributes and 112 attributes typically used for CBMs) are presented in Fig. 6. As previously mentioned, CUB image-level annotations are often inconsistent with class-level attribute vectors, with only 57.50% of labels agreeing with the class-level annotations. On the other hand, SUB’s annotations accurately represent the images, with 98.90% of all attributes being correctly labeled according to human agreement.

5.4. CBM Evaluation Setup

Using SUB as an evaluation set, we assess CBMs ability to generalize to our novel combinations of known concepts.

Base CBMs. CBMs can be trained in three ways [30]: 1) jointly, where the image-to-concept and concept-to-label components are trained simultaneously using supervised learning; 2) sequentially, where the image-to-concept component is trained first, then the concept-to-label component with the former frozen; and 3) independently, where the two components are trained separately, with the class-to-label network receiving only ground-truth concept labels. We ex-

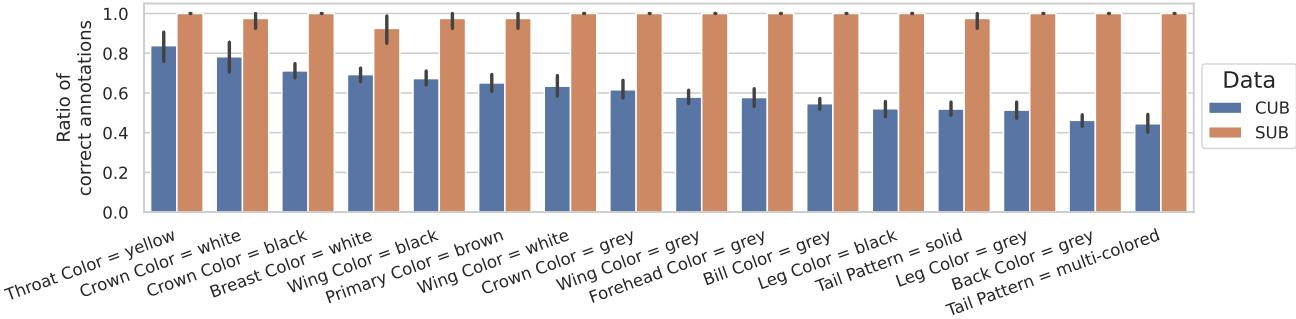


Figure 6. For all 17 attributes included in both SUB and CUB [63], SUB has higher annotation accuracy (std. as error bar). For CUB, we measure agreement between image-level and class-level attributes. For SUB, we use the attribute score from our human validation study.

clude sequential training from our evaluation, as it yields the same concept predictions as independent training.

Per-Concept CBM. CBMs can be trained individually per concept to improve concept accuracy [23], making the model less likely to look at spurious information, as it does not share weights with other attributes.

Concept Embedding Models (CEM). CEMs attempt to overcome the interpretability-performance tradeoff seen in CBMs by using a concept vector rather than a binary prediction [67]. This vector contains two activations, one symbolizing a concept’s presence and the second its absence.

Concept Labels. We explore labeling choices. The CBM authors proposed to train the model with fixed attribute vectors per class [30]. However, fixing the labels across images could hurt generalization by forcing the model to output identical predictions regardless of inter-image fluctuations. We explore models trained with class-defined attribute vectors and with the original per-image labels. Moreover, concepts can be supervised by binary attribute labels (hard) or by annotator confidence (soft).

Human Labeling: SUB. To ensure SUB’s high quality, we establish a human baseline where volunteers label our modified attributes. For each bird-attribute pairing, we test three images and prompt humans to select which attribute within the target group is present. We then calculate the accuracy of the human labels with respect to our intended labels. The User Interface is shown in Appendix E.

Human Labeling: CUB. We calculate the accuracies on the original CUB classes by generating class-level labels through majority voting across all images in each bird class. We then compute the proportion of image-attribute labels where the image-level annotation agrees with the class-level majority. This helps assess how well the class-level majority approximates the image-level attribute values.

5.5. Benchmarking CBMs on SUB

We report each model’s performance in detecting the substituted target attribute (\mathcal{S}^+) in our synthetic SUB data, as

presented in Table 1. \mathcal{S}^+ evaluates the 17 concepts in the overlap between attributes in SUB (33 attributes) and 112 (of 312) attributes used for CBM training [30]. We also measure the models’ ability to remove the original attribute (\mathcal{S}^-); 100% accuracy means that the CBM never predicted the removed attribute in the SUB data. For comparison with the accuracy of the classes seen in training, we include the CUB [63] test set on the subset of attributes used in SUB (\mathcal{T}_A), and the test set accuracy on all CUB concepts (\mathcal{T}).

We observe that annotators consistently classify SUB attributes, achieving 94.0% accuracy on the target attribute \mathcal{S}^+ , and 96.8% accuracy on not picking the removed attribute \mathcal{S}^- . These accuracies are much higher than CUB’s (79.4% for \mathcal{T}_A and 82.5% for \mathcal{T}). Despite all CBMs having high accuracy in \mathcal{T}_A and \mathcal{T} (up to 96.7%), they generalize poorly to our novel attribute combinations (highest \mathcal{S}^+ : 45.7% from CEM [67]). In fact, all tested CBMs detect \mathcal{S}^+ less accurately than random chance (50%). Also, high \mathcal{S}^- removal accuracy is often paired with low \mathcal{S}^+ (e.g. per-class soft labels with 91.6% on \mathcal{S}^- , but 11.2% on \mathcal{S}^+), suggesting that some models may simply have a greater tendency to predict *false* overall. Among all tested CBMs, the CEM is the best at predicting \mathcal{S}^+ with 45.7%, albeit still below chance, while maintaining relatively high performance of 87.2% on \mathcal{S}^- . All tested CBMs generalize poorly, showing that their predictions are not grounded in the target concepts and the performance on the training classes is misleading about the models’ true interpretability.

5.6. Benchmarking Other VLMs on SUB

Many interpretable models rely on VLM backbones such as CLIP [47] to generalize to open-vocabulary settings without explicit training data [38, 42, 50, 66]. If these backbones also poorly ground predictions, all downstream models will be affected. In testing VLMs, we evaluate whether large-scale training can mitigate the challenges CBMs face and recognize SUB attributes, exhibited in Table 2.

As our tested VLMs cannot easily make binary predic-

Per-Img. Soft	SUB				CUB			
	\mathcal{S}^+	\mathcal{S}^-	\mathcal{T}_A	\mathcal{T}	\mathcal{S}^+	\mathcal{S}^-	\mathcal{T}_A	\mathcal{T}
<i>Random Chance</i>	50.0	50.0	50.0	50.0				
CBM (ind.) [30]	40.8	51.6	95.9	96.7				
CBM (joint) [30]	34.3	54.0	96.1	96.9				
CBM (per c.) [23]	0.39	100.0	78.3	79.4				
CEM [67]	45.7	87.2	77.9	81.3				
CBM (ind.) [30]	✓	12.9	94.0	84.8	85.5			
CBM (joint) [30]	✓	5.78	94.0	85.8	86.1			
CBM (ind.) [30]	✓	32.8	73.6	81.6	85.3			
CBM (joint) [30]	✓	27.1	67.1	83.4	86.0			
CBM (ind.) [30]	✓	11.2	91.6	80.9	82.6			
CBM (joint) [30]	✓	6.06	92.4	74.9	75.8			
Human		94.0	96.8	79.4	82.5			

Table 1. We evaluate CBM accuracy on SUB by measuring the substituted attribute (\mathcal{S}^+) and the removed attribute (\mathcal{S}^-), as well as on CUB (\mathcal{T}) and the SUB attribute label subset (\mathcal{T}_A).

tions for individual attributes, we classify the target attribute among all within the attribute group or *none*. The attribute with the highest cosine similarity compared to the image is labeled *true* while all others are predicted *false*. These open-set vocabulary models choose from the 312 CUB attributes [63], and we evaluate on the cleaner CBM class-aggregated labels [30]. We re-calculate random and human baselines given this multiclass classification setting (see Appendix F).

Even with large-scale pre-training, CLIP [47], SigLIP [69], and EVA-CLIP [58] continue to face challenges consistently identifying \mathcal{S}^+ . While their overall accuracy is higher (with EVA-CLIP [58] achieving 46.8%), a deeper look reveals interesting patterns. For instance, CLIP, SigLIP, and EVA-CLIP demonstrate a tendency to select the original attribute two to three times more often than random chance (9.3%). This suggests a form of hallucination, where the models incorrectly identify the original attribute even when it is not present. For example, SigLIP 400m/16 [69], the model with the lowest \mathcal{S}^- hallucination rate, achieves only 81.7%, incorrectly selecting the original attribute 18.3% of the time. This analysis suggests that despite advances in large-scale pre-training, the problem of generalization for individual concept classification remains a persistent hurdle for VLMs.

6. Limitations

Since TDG requires some human intervention for prompt creation and filtering verification, future improvements could focus on fully automating these processes. This would enable the creation of more explainable datasets, addressing the limited scope of CUB. Additionally, due to the

	SUB		CUB	
	\mathcal{S}^+	\mathcal{S}^-	\mathcal{T}_A	\mathcal{T}
<i>Random Chance</i>	9.3	90.7	73.3	74.6
CLIP ViT-B32 [47]	39.2	73.1	78.4	79.7
CLIP ViT-L14 [47]	45.5	73.2	78.6	80.1
SigLIP B/16 [69]	45.2	77.5	78.4	79.2
SigLIP 400m/16 [69]	45.7	81.7	77.6	79.2
SigLIP2 B/16 [61]	40.0	76.6	77.6	79.1
EVA-CLIP [58]	46.8	77.6	78.5	79.4
Human	92.4	97.3	69.3	65.3

Table 2. We evaluate VLM accuracy on SUB by measuring the substituted attribute (\mathcal{S}^+) and removed attribute (\mathcal{S}^-), along with CUB accuracy (\mathcal{T}) and SUB’s attribute subset (\mathcal{T}_A).

automated filtering system, we cannot guarantee that every image in SUB perfectly represents the target attribute and bird. However, our human validation strongly suggests that SUB is more consistently and accurately labeled than CUB. Lastly, although our human studies involve fewer participants and annotations compared to CUB, we believe that the careful automatic filtering makes validating a subset sufficient as opposed to labeling the complete dataset.

7. Conclusion

We proposed a new dataset, SUB, to benchmark the grounding of attribute predictions in concept models. SUB consists of 38,400 images representing 768 unique bird-attribute pairs, where the given attribute is applied to the chosen bird. To generate SUB, we proposed TDG, a test-time adaptation to faithfully generate novel attribute-object combinations by tying them to a second, easier image. Through rigorous filtering of the birds, attributes, and resulting images, we ensured the quality of SUB. We demonstrated that CBM concept predictions are poorly grounded in the target concepts and fail to generalize to unseen combinations of known concepts. We also revealed that SOTA VLMs experience this issue as well, though to a lesser extent. We hope that SUB will pave the way for the next generation of CBMs with more robust and well-grounded explanations.

Acknowledgments

This work was partially funded by the ERC (853489 - DEXIM) and the Alfried Krupp von Bohlen und Halbach Foundation, which we thank for their generous support. The authors gratefully acknowledge the scientific support and resources of the AI service infrastructure *LRZ AI Systems* provided by the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities (BAdW), funded by Bayerisches Staatsministerium für Wissenschaft und Kunst (StMWK).

References

- [1] Hasan Md Tusfiqur Alam, Devansh Srivastav, Md Abdul Kadir, and Daniel Sonntag. Towards interpretable radiology report generation via concept bottlenecks using a multi-agentic rag. In *arXiv*, 2024. [2](#)
- [2] David Alvarez-Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *NeurIPS*, 2018. [2](#)
- [3] Ananthu Aniraj, Cassio F Dantas, Dino Ienco, and Diego Marcos. Pdiscoformer: Relaxing part discovery constraints with vision transformers. In *ECCV*, 2024. [2](#)
- [4] Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. SegA: Instructing text-to-image models using semantic guidance. In *NeurIPS*, 2024. [2](#)
- [5] Chaofan Chen, Oscar Li, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, 2018. [2](#)
- [6] Hongyu Chen, Yi-Meng Gao, Min Zhou, Peng Wang, Xubin Li, Tiezheng Ge, and Bo Zheng. Enhancing prompt following with visual control through training-free mask-guided diffusion. In *arXiv*, 2024. [2](#)
- [7] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, ZhaoYang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. In *arXiv*, 2024. [5](#)
- [8] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. In *arXiv*, 2024.
- [9] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024. [5](#)
- [10] Townim Faisal Chowdhury, Vu Minh Hieu Phan, Kewen Liao, Minh-Son To, Yutong Xie, Anton van den Hengel, Johan W. Verjans, and Zhibin Liao. Adacbm: An adaptive concept bottleneck model for explainable and accurate diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2024. [2](#)
- [11] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *arXiv*, 2023. [2](#)
- [12] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. [2](#)
- [13] Yilun Du and Leslie Pack Kaelbling. Position: Compositional generative modeling: A single model is not all you need. In *ICML*, 2024. [2](#)
- [14] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. In *NeurIPS*, 2020. [2](#)
- [15] Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Narain Sohl-Dickstein, A. Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *ICML*, 2023. [2](#)
- [16] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E. Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. In *NeurIPS*, 2023. [2, 4](#)
- [17] Luca Vincent Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. In *NeurIPS*, 2024. [2](#)
- [18] Yutong Feng, Biao Gong, Di Chen, Yujun Shen, Yu Liu, and Jingren Zhou. Ranni: Taming text-to-image diffusion for accurate instruction following. In *CVPR*, 2024. [2](#)
- [19] Jack Furby, Daniel Cunnington, Dave Braines, and Alun David Preece. Can we constrain concept bottleneck models to learn semantically meaningful input features? In *arXiv*, 2024. [2](#)
- [20] Sachit Gaudi, Gautam Sreekumar, and Vishnu Naresh Boddeti. Coind: Enabling logical compositions in diffusion models. *ICLR*, abs/2503.01145, 2025. [2](#)
- [21] Marton Havasi, S. Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In *NeurIPS*, 2022. [2](#)
- [22] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip H. S. Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023. [2, 4](#)
- [23] Lena Heidemann, Maureen Monnet, and Karsten Roscher. Concept correlation and its effects on concept-based models. In *WACV*, 2023. [2, 7, 8](#)
- [24] Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. Funnybirds: A synthetic vision dataset for a part-based analysis of explainable ai methods. In *ICCV*, 2023. [2](#)
- [25] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021. [3](#)
- [26] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panagiotis G. Ipeirotis, Pietro Perona, and Serge J. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*, 2015. [6](#)
- [27] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. [2](#)
- [28] Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. Reveilio: Interpreting and leveraging semantic information in diffusion models. In *arXiv*, 2024. [2](#)
- [29] Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. Datadream: Few-shot guided dataset generation. In *ECCV*, 2024. [2](#)
- [30] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *ICML*, 2020. [1, 2, 6, 7, 8](#)

- [31] Black Forest Labs. Announcing black forest labs. In *Black-ForestLabs Blog*, 2024. 2, 4, 6
- [32] Nan Liu, Shuang Li, Yilun Du, Joshua B. Tenenbaum, and Antonio Torralba. Learning to compose visual relations. *NeurIPS*, 2021. 2
- [33] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. *ECCV*, 2022. 2
- [34] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. In *ArXiv*, 2021. 2
- [35] Alireza Makhzani and Brendan J. Frey. k-sparse autoencoders. In *arXiv*, 2013. 2
- [36] Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretable, leak-proof concept-based models. In *NeurIPS*, 2022. 2
- [37] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? In *arXiv*, 2021. 2
- [38] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *ICLR*, 2023. 7
- [39] Sicheng Mo, Fangzhou Mu, Kuan Heng Lin, Yanli Liu, Bochen Guan, Yin Li, and Bolei Zhou. Freecontrol: Training-free spatial control of any text-to-image diffusion model with any condition. In *CVPR*, 2024. 2
- [40] Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *CVPR*, 2023. 2
- [41] Micky C. Nnamdi, Wensi Shi, Junior Ben Tamo, Henry J. Iwinski, J. Michael Wattenbarger, and May Dongmei Wang. Concept bottleneck model for adolescent idiopathic scoliosis patient reported outcomes prediction. In *International Conference on Biomedical and Health Informatics (BHI)*, 2023. 2
- [42] Tuomas P. Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 2, 7
- [43] Mateusz Pach, Dawid Rymarczyk, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Lucidppn: Unambiguous prototypical parts network for user-centric interpretable computer vision. In *arXiv*, 2024. 2
- [44] Konstantinos Panousis, Dino Ienco, and Diego Marcos. Coarse-to-fine concept bottleneck models. In *NeurIPS*, 2024. 2
- [45] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *ICCV*, 2019. 2
- [46] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 2
- [47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 7, 8
- [48] Naveen Raman, Mateo Espinosa Zarlenga, Juyeon Heo, and Mateja Jamnik. Do concept bottleneck models respect localities? In *arXiv*, 2024. 2
- [49] Naveen Raman, Mateo Espinosa Zarlenga, and Mateja Jamnik. Understanding inter-concept relationships in concept-based models. In *ICML*, 2024. 2
- [50] Sukrut Rao, Sweta Mahajan, Moritz Bohle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *ECCV*, 2024. 2, 7
- [51] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2
- [52] Dawid Rymarczyk, Lukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2020. 2
- [53] Dawid Rymarczyk, Lukasz Struski, Michal G'orszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *ECCV*, 2021. 2
- [54] Mert Bulent Sarıyıldız, Alahari Karteek, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *CVPR*, 2022. 2
- [55] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. In *IEEE Access*, 2022. 2
- [56] Nishad Singhi, Jae Myung Kim, Karsten Roth, and Zeynep Akata. Improving intervention efficacy via concept realignment in concept bottleneck models. In *ECCV*, 2024. 2
- [57] Sanchit Sinha, Mengdi Huai, Jianhui Sun, and Aidong Zhang. Understanding and enhancing robustness of concept-based models. In *AAAI*, 2022. 2
- [58] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv*, 2023. 8
- [59] Andong Tan, Fengtao Zhou, and Hao Chen. Explain via any concept: Concept bottleneck model with open vocabulary concepts. In *ECCV*, 2024. 2
- [60] Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *arXiv*, 2025. 2
- [61] Michael Tschanne, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv*, 2025. 8
- [62] Robert van der Klis, Stephan Alaniz, Massimiliano Mancini, Cassio F Dantas, Dino Ienco, Zeynep Akata, and Diego Marcos. Pdisconet: Semantically consistent part discovery for fine-grained recognition. In *ICCV*, 2023. 2
- [63] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. In *California Institute of Technology Technical Report*, 2011. 2, 4, 6, 7, 8

- [64] Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. On discrete prompt optimization for diffusion models. In *ICML*, 2024. [2](#)
- [65] Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. *NeurIPS*, 2023. [2](#)
- [66] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *CVPR*, 2023. [2](#), [7](#)
- [67] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frédéric Precioso, Stefano Melacci, Adrian Weller, Pietro Lio', and Mateja Jamnik. Concept embedding models. In *NeurIPS*, 2022. [7](#), [8](#)
- [68] Mateo Espinosa Zarlenga, Pietro Barbiero, Zohreh Shams, Dmitry Kazhdan, Umang Bhatt, Adrian Weller, and Mateja Jamnik. Towards robust metrics for concept representation evaluation. In *AAAI*, 2023. [2](#)
- [69] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. [8](#)
- [70] Kaichen Zhang, Yifei Shen, Bo Li, and Ziwei Liu. Large multi-modal models can interpret features in large multi-modal models. In *arXiv*, 2024. [2](#)
- [71] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. [2](#)
- [72] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. In *NeurIPS*, 2023. [2](#)

Attribute	Guidance Bird
Needle bill shape	Needle bill shape
Spotted breast pattern	Brown Thrasher
Striped breast pattern	Song Sparrow
Solid tail pattern	Gray Catbird
Multi-colored tail pattern	Cedar Waxwing

Table 3. Guidance birds used the the generation of SUB.

A. Reference Birds

For SUB, we use the following 33 reference birds: Western Grebe, Black and white Warbler, European Goldfinch, Pacific Loon, White Pelican, Cedar Waxwing, Gadwall, Downy Woodpecker, Pileated Woodpecker, Purple Finch, Common Raven, White breasted Nuthatch, Northern Flicker, Mallard, Tropical Kingbird, Tree Swallow, Song Sparrow, Green Violetear, Gray Catbird, Green Jay, Cardinal, Red bellied Woodpecker, Pied Kingfisher, Rufous Hummingbird, Dark eyed Junco, Green Kingfisher, Horned Puffin, Anna Hummingbird, Barn Swallow, American Goldfinch, Lazuli Bunting, Blue Jay, Painted Bunting.

B. Guidance Birds

Guidance birds are used for pattern and shape modifications. We include in Table 3 the guidance birds chosen for each attribute when generating SUB.

C. Substitutions

We use the following list of substitutions in SUB: grey back color, grey bill color, white breast color, red breast color, blue breast color, grey crown color, white crown color, black crown color, pink crown color, yellow eye color, blue eye color, white eye color, grey forehead color, pink leg color, black leg color, grey leg color, green primary color, brown primary color, blue primary color, orange primary color, blue throat color, yellow throat color, green underparts color, red underparts color, white wing color, grey wing color, black wing color, spotted breast pattern, striped breast pattern, solid tail pattern, multi-colored tail pattern, and needle bill shape.

D. Prompts

A few example prompts:

\mathcal{R} = European Goldfinch, \mathcal{S}^+ = Black crown color, \mathcal{G} = bird, $c_{\mathcal{R}}$ = A photo of a European Goldfinch with black colored feathers on the crown of its head, $c_{\mathcal{G}}$ = A photo of a bird with black colored feathers on the crown of its head

\mathcal{R} = Downy Woodpecker, \mathcal{S}^+ = Red breast color, \mathcal{G} = bird, $c_{\mathcal{R}}$ = A photo of a Downy Woodpecker with a red

Annotate the Image

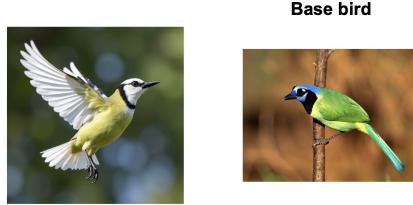


Image 168 of 38399

[Previous](#) [Next](#)

Are these correct?

has_wing_color::white **Green_Jay**

yes somewhat no yes somewhat no

Submit

Figure 7. First human study interface with binary questions for attribute presence and reference bird faithfulness.

colored breast, $c_{\mathcal{G}}$ = A photo of a bird with a red colored breast

\mathcal{R} = Western Grebe, \mathcal{S}^+ = Solid tail pattern, \mathcal{G} = Gray Catbird, $c_{\mathcal{R}}$ = A photo of a Western Grebe with a solid tail like a Gray Catbird, $c_{\mathcal{G}}$ = A photo of a Gray Catbird with a solid tail

\mathcal{R} = Cardinal, \mathcal{S}^+ = Spotted breast pattern, \mathcal{G} = Brown Thrasher, $c_{\mathcal{R}}$ = A photo of a Cardinal with a spotted belly like a Brown Thrasher, $c_{\mathcal{G}}$ = A photo of a Brown Thrasher with a spotted belly

\mathcal{R} = Blue Jay, \mathcal{S}^+ = Needle bill shape, \mathcal{G} = Hummingbird, $c_{\mathcal{R}}$ = A photo of a Blue Jay with the body of a Blue Jay and a beak like a Hummingbird, $c_{\mathcal{G}}$ = A photo of a Hummingbird

E. Human Verification User Interface

Human verification was completed by four volunteers. In Figure 7, we see the user interface used for our first user study, where participants were asked whether \mathcal{S}^+ was present and whether the bird accurately reflected the guide bird. In Figure 8, we show the interface for the second study, where the user is given all options in the target attribute group and asked to label which is present.

E.1. Reference Bird Verification

The underlying objective of specifying reference birds is to increase the overall diversity in birds exhibiting individual attributes. Specifically, we want to test the accuracy of attribute detection when it occurs in combinations not seen during test time. As long as \mathcal{S}^+ is present, it is not imperative that every synthetic bird closely match the reference class, but many should. As described in Section 4.3, we verify this on 40 images per attribute, by checking if the

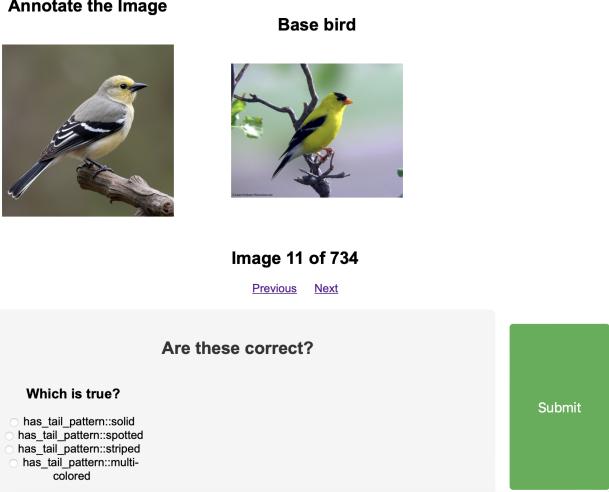


Figure 8. Second human study with attribute labeling within full attribute group.

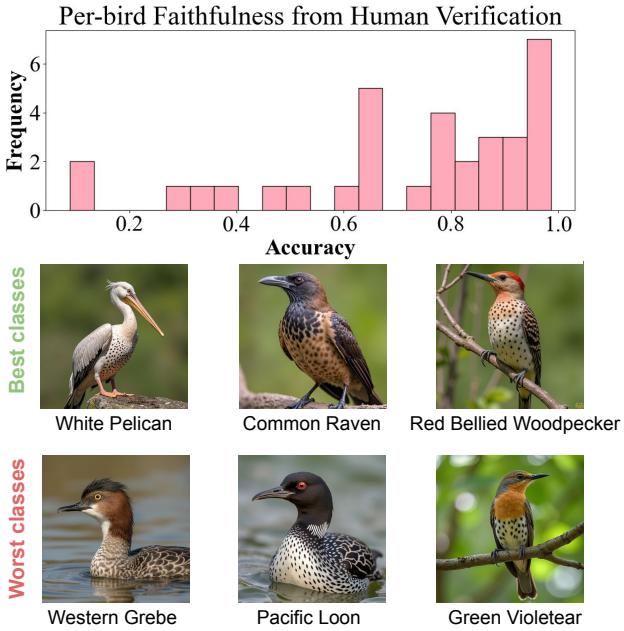


Figure 9. Histogram of the per-bird accuracy results from the human verification, where participants were asked if the attribute-substituted synthetic image represents the original bird. We see that 27 birds exceed 50% accuracy, showing that our generated dataset is very diverse.

synthetic bird is recognizable as the reference bird. We then calculate the percentage of faithful birds of all those generated for each bird class. In Figure 9, we show a histogram of this per-bird faithfulness. From this histogram, we can see that 27 out of 33 classes are faithful over half the time, and 10 classes are over 90% faithful. For the attribute *spot-*

ted breast, we show examples from the three most faithful classes, and the three least faithful. While *Western Grebe*, *Pacific Loon*, and *Green Violetear* diverge from the representative class, we also note that they still provide some diversity to SUB.

F. VLM Random Chance Calculation and Label Set

For the VLMs, we calculate the probability of getting a single prediction correct at random if the target label is 1 as $\frac{1}{|\mathcal{A}|+1}$, where \mathcal{A} is the attribute group corresponding to the target prediction and options $a_j \in \mathcal{A}$ are the manifestations of the attribute group. One is added to $|\mathcal{A}|$ to account for the additional option *none*. If the target label is 0, then it is $1 - \frac{1}{|\mathcal{A}|+1}$.

For SUB, we calculate the S^+ random chance baseline across the modified attribute for each image in SUB, assuming a target label of 1. We calculate S^- from only the samples where the class-wise CBM label included a positive label for another attribute within the attribute group, and we consider that attribute with a target label of 0.

For CUB, we calculate the random chance baseline across all samples and CBM attributes, with the CBM class-wise labels as targets.

For selecting the possible label set \mathcal{A} presented to the VLM, we use the full set of 312 CUB attributes for two reasons: (1) it offers a broader and more challenging set of plausible options than the CBM subset; and (2) the original labels used in CUB collection are well-aligned with expected dataset attributes, increasing the likelihood that the model selects the correct attribute over *none*.