# Simulating Concept Bottlenecks with Vision-Language Models

Anonymous ACL submission

## Abstract

In high-stakes domains such as healthcare and finance, understanding why a model makes a prediction is often as important as the prediction itself. Concept Bottleneck Models (CBMs) enhance transparency by first predicting human-interpretable concepts before producing the final decision, allowing experts to inspect and correct intermediate reasoning. We demonstrate that large vision–language models (VLMs) can naturally support this paradigm by leveraging their parametric knowledge and generative capabilities. We introduce LangCBM, which uses VLMs to generate textual descriptions of visual concepts, followed by a lightweight extraction and classification pipeline. Training via supervised fine-tuning (SFT), optionally followed by reinforcement learning (RL), yields accurate concept predictions. Across synthetic and real-world benchmarks, LangCBM achieves competitive concept and label accuracy compared to alternative CBM formulations, establishing VLM-generated text as a viable, interpretable bottleneck representation.

## 1 Introduction

AI systems are increasingly used in critical domains such as healthcare, finance, and scientific discovery, where transparent and accountable decision-making is essential. In medical applications, for example, experts need not only accurate predictions but also clear justifications.

To address these needs, researchers have turned to self-explainable models, which provide built-in transparency instead of relying solely on post-hoc explanations. One prominent approach in this direction is Concept Bottleneck Models (Koh et al., 2020). Instead of mapping raw input data (usually, an image) directly to final predictions, CBMs first predict a set of concepts, which are then used as the only inputs for the component making the final decision. This explicit separation enhances human oversight, allowing domain experts to inspect, validate, and modify the predicted concepts. Typical CBMs rely on pre-trained image encoders and do not use natural language at all.

Modern vision-language models (VLMs) present an interesting opportunity: they possess extensive knowledge about visual concepts and can express this knowledge through natural language. The key question becomes: how can we exploit these pre-trained capabilities to build interpretable and effective concept bottleneck systems? While constructing a CBM typically assumes the reliance on specialized architectures, we show that the concept bottleneck can be effectively emulated within a text generation framework using a vision-language model (VLM). Specifically, in our approach (**LangCBM**), a VLM generates a description of an input image's concepts, which are then mapped to labels by a separate language model (see Figure 1 for SFT training). Note that this second-stage model has no access to the original image.

While recent works (Sun et al., 2024; Ismail et al., 2024) integrate LLMs into CBMs, they do so by introducing non-standard components or specialized training objectives. These modifications increase architectural complexity; this additional complexity makes real-world deployment harder. Our main contributions are:

- we demonstrate that concept bottlenecks can be effectively simulated within VLM text generation without architectural modifications, using only standard supervised fine-tuning;
- we show that randomizing concept order during training serves as a crucial augmentation that prevents mode collapse in imbalanced concept distributions;
- we provide empirical evidence that reinforcement learning with LLM-as-Judge improves concept prediction;

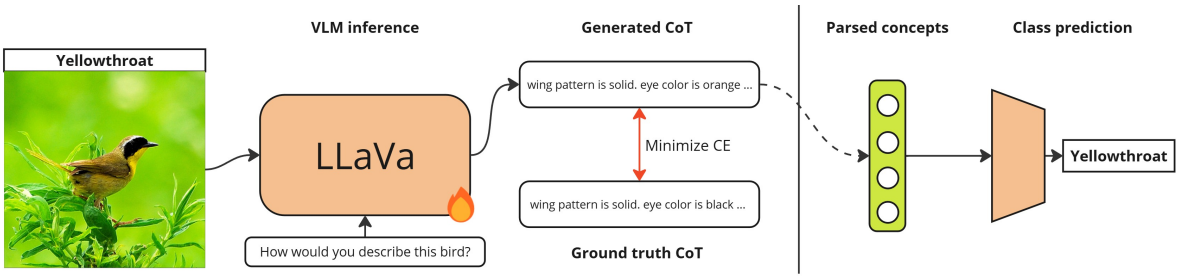


Figure 1: LangCBM pipeline: a Concept Bottleneck implemented within VLM generation. Training uses supervised fine-tuning (SFT) with cross-entropy (CE) loss, or reinforcement learning (RL) where the CE loss is replaced by an LLM-judged reward encouraging inclusion of correct concepts and penalizing errors.

- we validate our approach on multiple datasets, showing it matches or outperforms alternative CBM formulations while maintaining interpretability and intervention capabilities.[1]

## 2 Background: Concept Bottleneck Models

Concept bottleneck models (CBMs) (Koh et al., 2020) enable explanation of neural network decisions through human-understandable concepts, allowing for concept-based corrections and improved robustness to covariate shifts. Formally, CBMs compose two functions: $\hat{y} = f(g(x))$, where $g : \mathbb{R}^D \to \mathbb{R}^k$ maps inputs to concept space and $f : \mathbb{R}^K \to \mathbb{R}$ maps concepts to predictions, requiring supervised concept labels during training. This architectural design creates an interpretable intermediate representation that domain experts can inspect and potentially intervene upon, making CBMs particularly valuable in high-stakes applications where understanding model decisions is crucial.

Havasi et al. (2022) enhance CBMs with residual connections and side-channel models to account for concept interdependencies, while preserving the main advantages of hard CBMs over soft variants – namely, reducing the likelihood that unintended information is encoded in concept variables. Structurally our CBM is similar to theirs: we also use hard concepts and an autoregressive model to predict them. However, they propose a custom architecture with specialized components for handling concept dependencies, whereas we rely on (V)LMs as building blocks in our pipeline, leveraging their pre-trained capabilities for both concept understanding and generation.

[1] The code is available at https://anonymous.4open.science/r/CB_in_CoT_Reasoning-639C

## 3 Related Work

In this section, we review prior approaches that integrate LLMs, VLMs, or, more broadly, textual supervision into CBMs.

CB-LLMs (Sun et al., 2024) integrate CBMs into the language processing domain, implementing concept bottlenecks for text classification and generation tasks. The approach introduces a significant architectural change: each token's embedding is split into two components - one that encodes concept information and another that remains concept-agnostic. This dual-pathway design allows the model to maintain both interpretable concept-based reasoning and additional representational capacity. While they focus exclusively on text processing and require architectural modifications to standard transformer models, our approach extends to multiple modalities and operates within the standard generation framework of VLMs without requiring architectural changes. Alukaev et al. (2023) leverage textual supervision to produce informative and disentangled concepts in the absence of concept annotations; however, their method does not employ LLMs or VLMs.

In Oikarinen et al. (2023); Yang et al. (2023); Qu and Yatskar (2024), LLMs and VLMs are used to generate concept sets and annotations, addressing the scalability challenge of obtaining human-annotated concepts. These methods leverage the parameteric knowledge to automatically propose relevant concepts for a given task and generate concept annotations at scale. Our approach differs fundamentally in that we instead rely on concept-annotated datasets and demonstrate that a VLM's generations can themselves serve as a concept layer. In fact, their work is orthogonal to ours: concept annotations for our method could be generated using their approaches, potentially enabling fully auto-

mated concept discovery and annotation pipelines.

He et al. (2025) propose Chat-CBM, which uses a frozen LLM as the final classifier in CBMs, creating a hybrid architecture that combines traditional vision encoders with language model reasoning. Their method first extracts concepts from images using standard vision encoders, then converts these concept activations into natural language descriptions. The LLM then predicts labels by reasoning over these concept descriptions rather than numeric scores, enabling more flexible and explainable decision-making. In contrast, our approach uses VLM's textual output to directly predict concepts from images, eliminating the need for separate vision encoders and leveraging the VLM's integrated multimodal understanding rather than using an LLM only for the final classification step. Nevertheless, the two paradigms are compatible: in principle, the extraction and classification stages in our framework could also be replaced with an LLM.

Yan et al. (2023) address attribute redundancy by learning to select concise subsets from large LLM-generated pools, showing that 32 attributes can match the performance of thousands. However, their approach requires pre-generating massive attribute sets and a separate selection mechanism. In contrast, our method directly embeds the concept bottleneck within VLM generation itself, learning concepts through the model's natural language output without needing extensive pre-generated pools or post-hoc selection.

Our approach connects to recent work on chain-of-thought (CoT) reasoning, where models generate intermediate traces of their decision process. Unlike CoT methods focused on multi-step symbolic or numerical reasoning (DeepSeek-AI et al., 2025), we use text generation to produce structured concept descriptions that serve as interpretable bottlenecks before classification, yielding explanations that are both transparent and intervenable. While CoTs can, in principle, expose intermediate computations akin to concept layers in CBMs, prior work (Lanham et al., 2023; Turpin et al., 2024) shows that such traces often diverge from a model's true reasoning, raising concerns about their reliability in high-stakes domains.

# 4 Methodology

We present a simple method for implementing CBMs with VLMs. We hypothesize that the ability of VLMs to describe images in natural language provides a strong foundation for constructing interpretable concept representations.

Following (Koh et al., 2020), our approach, LangCBM, assumes the availability of concept-annotated training data, where each image is labeled with binary indicators for the presence or absence of predefined concepts. We first convert these binary annotations into categorical ones using the groupings provided in the dataset and then create natural language statements about these categorical concepts.

## 4.1 Converting concept annotations to natural language

Given an image with binary concept annotations, we convert each annotation into a natural language statement. For every concept $c_i$ with binary value $b_i \in \{0, 1\}$, we generate:

- If $b_i = 1$: *"This [object] has [concept description]."*
- If $b_i = 0$: *"This [object] does not have [concept description]."*

For example, given concept "yellow_throat" with value 1 for a bird image, we generate "This bird has a yellow throat." These sentences are concatenated in **random order** to form the target output. As we will see in our experiments, this randomization is crucial: concept labels in real datasets are often highly imbalanced, and random ordering prevents the model from learning spurious position-based patterns, effectively serving as an augmentation technique that improves robustness to class imbalance (see Section 6).

## 4.2 Training process

**Stage 1: VLM fine-tuning.** We fine-tune a VLM using LoRA adapters. The model is conditioned on images with a fixed prompt (for example, "Describe this bird" for CUB dataset) and trained to generate the concatenated concept descriptions using the standard cross-entropy loss. Prompt tuning was performed on a held-out dataset.

**Stage 2: Concept-to-Label classifier.** After VLM training, we extract concept predictions from generated text and use them to train a linear classifier mapping concepts to final labels. This classifier is trained on the same data splits as the VLM, using only the predicted concepts as features.

To extract structured concept predictions from the model's free-form text generations, we employ

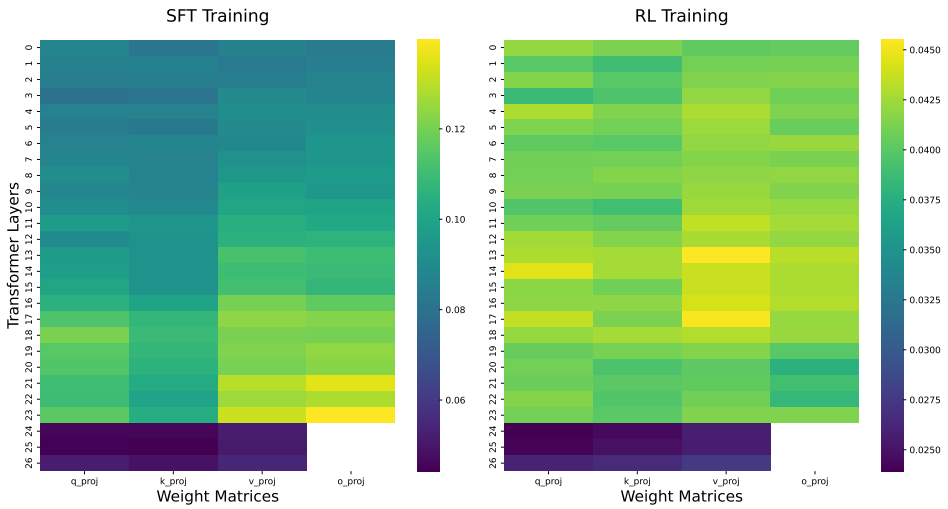


Figure 2: Change in LoRA weight norms compared to init weights when training with SFT and RL

a two-step process. First, we segment the generated text into individual sentences. Then, we map each sentence to a binary concept (presence or absence) using one of two methods: (1) direct string matching for models trained with supervised fine-tuning, where the output format is predictable (as will be with SFT), or (2) evaluation by a separate LLM for reinforcement learning models, where the output format may vary (with RL).

This approach effectively simulates the behavior of basic concept bottlenecks within a VLM's generations while maintaining architectural simplicity and leveraging the extensive knowledge of pre-trained VLMs.

### 4.3 Reinforcement Learning

Fixed concept verbalizations in SFT can be suboptimal: unnatural phrasing may force large updates and reduce generalization. Allowing the model to learn verbalizations that are both correct and natural preserves internal knowledge and can improve generalization to new images and concept combinations. To achieve this, we train our VLMs using GRPO (Shao et al., 2024).

For each training image, we generate several hypothesis trajectories. A different LLM evaluates each trajectory using a prompt that checks whether generated text explicitly confirms presence or implies absence of each ground-truth concept (see Appendix C for the full prompt).

The reward function accumulates scores across all concepts, assigning +1 for each correct classification and -1 for each incorrect one. This allows the model to generate free-form outputs as long as the judge LLM can parse the concept annotations. We include KL regularization with respect to the reference model to promote training stability and prevent excessive policy deviation, which could otherwise impair generalization to new examples.

## 5 Datasets

We benchmarked our approach on four datasets: our own synthetic dataset, CUB-200 (Wah et al., 2011), AwA2 (Xian et al., 2019) and ISIC (Kurtansky et al., 2024)

### 5.1 Synthetic lines dataset

We randomly chose the number of lines to draw on the image, uniformly between 1 and 3. The angle and offset of each line were sampled randomly uniformly from $[-\pi; \pi) \times [0.2 * image_size; image_size]$. To increase variety, the line colors were also chosen randomly. The dataset includes 4 concepts, representing the number of intersections on the image as one-hot vectors. The final task is to predict whether a triangle appears in the image. This simple setup serves as a controlled sanity check to ensure that our method behaves as expected under controlled conditions.

### 5.2 CUB-200

CUB-200 (Wah et al., 2011) dataset is the most common benchmark for CBMs. We use concepts filtering procedure of Koh et al. (2020) and keep only concepts which are present for at least 10 classes. The train, validation and test split as well as concepts annotations come from Koh et al. (2020); there are 4796 training, 1198 validation

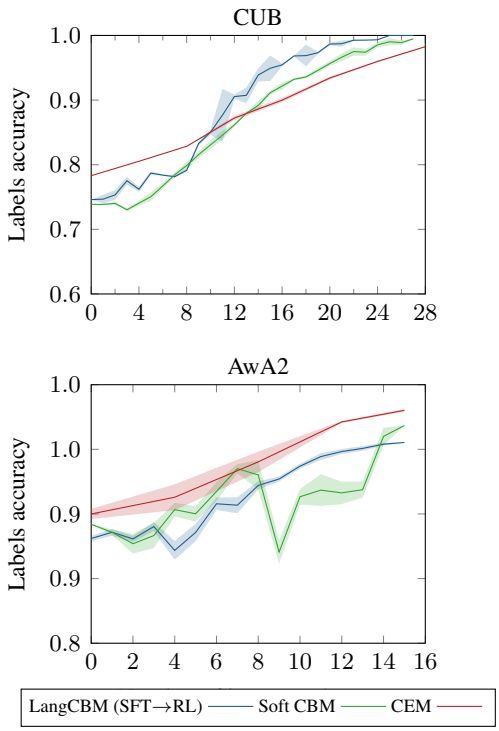


Figure 3: Change in target prediction accuracy after intervening on concept groups. LangCBM (SFT→RL) is our VLM-based approach, compared against Soft CBM (baseline concept bottleneck model) and CEM (Concept Embedding Models).

and 5794 test images, annotated with 112 concepts and 200 classes.

### 5.3 AwA2

Animals with attributes Xian et al. (2019) contains 37K images of 50 animal species, described by 85 concepts. For this data set, we follow the concept filtering procedures of Kim et al. (2023).

### 5.4 ISIC

This dataset consists of more than 10K images of skin lesions annotated with seven categories of skin cancer Kurtansky et al. (2024). For this dataset, we only experimented with VLMs pretrained on medical imaging datasets for stronger initialization and prior knowledge.

## 6 Experiments and results

### 6.1 Synthetic dataset validation

On our synthetic lines dataset, our model achieves 100% accuracy in both concept prediction and final classification tasks. While conceptually simple, this dataset serves as an important proof-of-concept, demonstrating that our architecture can perfectly capture geometric relationships between visual concepts (number of line intersections) and target classes (presence of triangles).

### 6.2 Benchmarks

We evaluated our LangCBM model against several state-of-the-art concept bottleneck approaches on the CUB-200-2011, AwA2 and ISIC datasets, following the evaluation protocols established in previous work (Koh et al., 2020; Havasi et al., 2022; Kim et al., 2023). For most baselines, we adopted the experimental setups from Kim et al. (2023) while changing the image encoder backbone to one used to initialize VLM in LangCBM approaches. For initialization, we pick LLaVA 7B 1.5 for general real-worlds datasets such as CUB and AwA2, and LLaVA-Med for medical data. Table 1 presents both concept prediction accuracy and label prediction accuracy for all models.

Our approach achieves competitive performance across all datasets, with notable variations in relative performance. On CUB, LangCBM (SFT→RL) achieves 95.9% concept accuracy and 74.6% target label accuracy, performing comparably to alternative CBM version while remaining 3-4 percentage points below CEM and ProbCBM on label prediction.

On AwA2, the performance gap narrows significantly: LangCBM (SFT→RL) achieves 98.0% concept accuracy and 88.1% label accuracy, nearly matching the best alternative (CEM at 90.0%). The smaller gaps in performance on AwA2 suggest that this dataset is easier compared to CUB as all methods achieve similar results.

On the medical ISIC dataset, LangCBM (SFT→RL) achieves 94.4% concept accuracy and notably outperforms alternative CBM formulations on label prediction (86.8% vs. 84.7% for ProbCBM), demonstrating that domain-specific pretraining (LLaVA-Med) provides substantial benefits.

The consistent drop in performance relative to black-box models (2-3% on AwA2, 4-5% on ISIC) is expected and aligns with CBM literature: the interpretability bottleneck necessarily discards some task-relevant information. However, this trade-off is acceptable given the gains in transparency and intervention capability.

The zero-shot LLaVA baseline (3.0% concept accuracy on CUB, 1.3% on AwA2) confirms that while pretrained models possess relevant world knowledge, task-specific fine-tuning remains essen-

| Dataset | Model | concepts | labels |
|---|---|---|---|
| CUB | Black-box | – | 91.9±0.2 |
| | CBM (ResNet) | 95.6±0.1 | 70.8±0.6 |
| | ProbCBM (ResNet) | 95.6±0.1 | 71.8±0.6 |
| | CEM (ResNet) | 95.4±0.1 | 75.9±0.2 |
| | Hard AR CBM (ResNet) | 95.7±0.1 | 75.4±0.1 |
| | CBM (ViT + LoRA) | 95.9 ± 0.2 | 73.87 ± 0.7 |
| | ProbCBM (ViT + LoRA) | 96.0 ± 0.3 | 77.6 ± 0.4 |
| | CEM (ViT + LoRA) | 96.0 ± 0.2 | **78.3 ± 0.5** |
| | LangCBM (SFT→RL) | **95.9± 0.3** | 74.6 ± 0.3 |
| AwA2 | Black-box | – | 89.3±0.0 |
| | CBM (ResNet) | 97.5±0.0 | 87.7±0.4 |
| | ProbCBM (ResNet) | 97.5±0.0 | 88.0±0.2 |
| | CEM (ResNet) | 97.9±0.1 | 88.4±0.2 |
| | CBM (ViT + LoRA) | 98.0 ± 0.3 | 89.2 ± 0.1 |
| | ProbCBM (ViT + LoRA) | 98.1 ± 0.4 | 89.8 ± 0.2 |
| | CEM (ViT + LoRA) | **98.2 ± 0.5** | **90.0 ± 0.1** |
| | LangCBM (SFT→RL) | 98.0 ± 0.3 | 88.1 ± 0.4 |
| ISIC | Black-box | – | 89.3±1.0 |
| | BiomedCLIP (zero-shot) | – | 24.5±0.0 |
| | CBM | 93.9 ± 1.0 | 82.2 ± 1.7 |
| | ProbCBM | 94.3±0.7 | 84.7±0.7 |
| | LangCBM (SFT→RL) | **94.4±2.0** | **86.8±2.1** |

Table 1: Concept and label prediction accuracies

tial. The dramatic improvement from fine-tuning validates our approach of using pre-trained VLMs for fine-tuning rather than relying on zero-shot capabilities alone.

Notably, randomizing concept order during training significantly improved performance compared to fixed ordering. The "LangCBM (fixed order)" baseline on AwA2 achieves 95.7% concept accuracy but only 2.0% label accuracy - performance collapses catastrophically. This occurs because concept annotations are highly imbalanced: most concepts are absent in most images. With fixed ordering, the model typically learns to always predict "absent" for all concepts, achieving high concept accuracy by exploiting the class imbalance but rendering the bottleneck useless for classification. Random ordering prevents this degenerate solution by disrupting position-based shortcuts, effectively serving as a crucial data augmentation.

### 6.3 Interventions

The ability of a human expert to correct the predicted concepts (i.e., *intervene*), thereby influencing the model's final prediction, is one of the key advantages of CBMs. In this experiment, we simulate interventions, by applying corrections towards ground-truth concepts at test time. We show in Figure 3 that our model's performance on the final task benefits from correcting interventions; moreover, its interventions curve is similar to that of a hard sequential CBM (Havasi et al., 2022).

### 6.4 Concept leakage analysis

Concept leakage in CBMs refers to the undesirable situation where concept activations encode information unrelated to their designated concepts (Havasi et al., 2022). Although leakage can improve target prediction accuracy, it undermines interpretability and intervenability: the model's predictions no longer rely solely on the declared concepts, so changing or inspecting those concepts no longer faithfully explains the model's behavior.

In alternative CBM formulations, leakage occurs through direct encoding within the concept activation vectors. In our VLM-based framework, however, any leakage would manifest through the generated text itself - for instance, if the model inserted additional informative cues beyond the defined concepts. This risk is mitigated in our setup, since we do not optimize the VLM for final label prediction (which could incentivize such behavior) but rather for accurately producing informative concept descriptions.

To probe whether any leakage remains, we conducted a concept corruption experiment: replacing original binary concept labels with random Bernoulli noise and generating textual descriptions accordingly. Figure 4 shows that classification accuracy drops sharply for both the standard CBM and our LangCBM as the proportion of corrupted concepts increases.

A model could, in principle, compensate for missing concept information by encoding it implicitly elsewhere. Yet our empirical results suggest this is not the case: the reliance on (V)LMs and the absence of joint training prevent such shortcuts. The consistent performance decline confirms that both models depend on the intended concept information rather than bypassing the bottleneck.

This result aligns with our intervention analysis, where increasing the number of intervened concept groups improves performance. Together, these findings indicate that LangCBM exhibits minimal concept leakage - at least comparable to, if not better than, alternative CBMs. If substantial leakage were present, we would expect resilience to concept corruption or muted gains from interventions; instead, we observe the opposite.

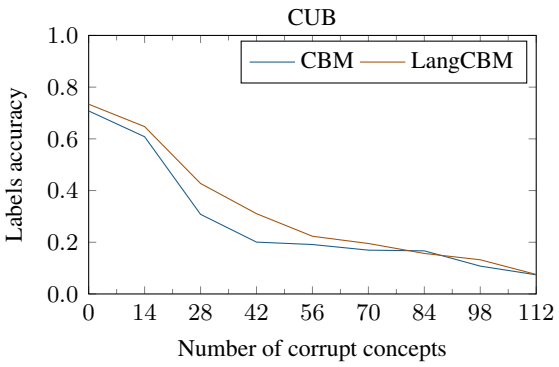


Figure 4: Decrease in target prediction accuracy as more and more concepts are replaced with random noise

| Dataset | Model | concepts | labels |
|---|---|---|---|
| CUB | Zero-shot LLaVA 1.5 | 3.0±0.1 | 5.0±0.2 |
| | LangCBM (SFT) | 95.7±0.1 | 73.4±0.1 |
| | LangCBM (RL) | 95.8± 0.2 | 74.1 ± 0.2 |
| | LangCBM (SFT→RL) | **95.9± 0.3** | **74.6 ± 0.3** |
| AwA2 | Zero-shot LLaVA 1.5 | 1.3±0.1 | 2.1±0.3 |
| | LangCBM (fixed order) | 95.7 ± 1.4 | 2.0 ± 0.4 |
| | LangCBM (SFT) | 97.6±0.1 | 87.8±0.2 |
| | LangCBM (RL) | 98.0 ± 0.2 | 88.0 ± 0.3 |
| | LangCBM (SFT→RL) | **98.0 ± 0.3** | **88.1 ± 0.4** |
| ISIC | LangCBM (SFT) | 93.9±1.0 | 85.5±1.3 |
| | LangCBM (RL) | 94.4±1.5 | 86.2±1.4 |
| | LangCBM (SFT→RL) | **94.4±2.0** | **86.8±2.1** |

Table 2: Ablation of LangCBM variants

## 7 Ablations and setups

### 7.1 Details on setups

We compare against ProbCBM (Kim et al., 2023), CEM (Zarlenga et al., 2022) and Hard AR CBM (Havasi et al., 2022). All models, both in LangCBM experiments and in baselines are trained with exactly the same ViT backbone with same LoRA parameters (r=128, $\alpha$=256) to ensure fair comparison. LangCBM variants: standard SFT, pure RL, and SFT→RL (0.1 epochs SFT followed by regular RL tuning). See Appendix B for details about baselines.

### 7.2 Training details and hyperparameters

**Supervised Fine-Tuning (SFT).** We fine-tune LLaVA 7B 1.5 on CUB and AwA2, and LLaVA-Med on ISIC using LoRA (Hu et al., 2022) adapters with rank $r = 128$ and scaling factor $\alpha = 256$. LoRA is applied to the query, key, value, and output projection matrices (q_proj, k_proj, v_proj, o_proj) in all transformer layers. We use the AdamW optimizer with learning rate $lr = 0.0002$, weight decay $weight_decay = 0.001$, and batch size $batch_size_per_device = 8$. Training employs cosine learning rate scheduling with 3% of iterations as warmup, and continues until convergence on validation loss.

**Reinforcement Learning with GRPO.** For policy gradient training, we use the same LoRA configuration but with reduced learning rate $lr = 0.00001$ and no weight decay ($weight_decay = 0.0$). We sample 4 generations per prompt with temperature $T = 0.7$ and nucleus sampling $top_p = 0.85$, training for one epoch over the sampled trajectories, adding KL regularization with coefficient $\beta = 0.001$. We use Llama-3.2-3B-Instruct as the judge LLM to evaluate concept predictions in generated text.

**SFT→RL Training.** For the combined approach, we first perform a brief SFT warm-up (0.1 epochs) to teach the model the output format and concept vocabulary, then continue with standard GRPO training as described above.

### 7.3 Role of fine-tuning

To assess the role of fine-tuning, we include zero-shot performance of the base LLaVa model. The significant gap between zero-shot and fine-tuned performance demonstrates that while pretrained models possess relevant world knowledge and are a good initialization point, task-specific training remains crucial for concept-based classification. Notably, randomizing concept order during training significantly improved performance compared to fixed ordering, acting as a regularization for imbalanced datasets. In Table 2 we include results for fixed concepts order as "LangCBM (fixed order)" and performance degradation for these models is evident. This might be due to concept annotations being imbalanced: an arbitrary concept is absent for most images and when training with fixed concept order the model collapses to predict all concepts as absent: thus concept accuracy is quite high but class prediction accuracy is poor.

### 7.4 SFT vs RL: quality and weight updates

Our experiments have shown that a model trained using RL performs better than SFT tuning. For RL, we consider two versions: pure RL, and RL with a SFT warm-up. Since concept descriptions require more detail than standard captions a VLM is used to, pure RL could struggle to learn the output format from sparse rewards alone, thus the warm-up

SFT is beneficial.

Figure 2 shows that SFT training leads to substantially larger weight updates than RL, yet achieves lower concept prediction accuracy across all datasets. Interestingly, in RL the change in weight norms is nearly uniform across layers, whereas in SFT the magnitude of change increases toward the output layers. The SFT→RL approach in Fig. 6 exhibits an interesting intermediate pattern: early layers show RL-like uniform updates while later layers retain some SFT-like gradient concentration. Quantitatively, the average Frobenius norm change for SFT→RL falls between pure SFT and pure RL.

Similar findings were presented in Rajani et al. (2025) for reasoning models: RL-based approaches achieve comparable or better results than SFT while making smaller weight updates from initialization. However, SFT remains useful as an initial warm-up training stage (Chu et al., 2025).

Figure 5 shows the intervention behavior of our three LangCBM training variants on the CUB dataset. All models exhibit similar intervention curves, with label accuracy improving from approximately 73% to near-perfect accuracy as concept groups are corrected. The SFT→RL model demonstrates slightly better performance overall, while maintaining comparable behavior to pure SFT and RL approaches. The convergence of all three curves confirms that our text-based concept bottleneck architecture preserves the key CBM property: human experts can effectively improve model predictions by correcting intermediate concepts, regardless of the specific training methodology employed.

In this work, we show that for downstream concept prediction tasks, it is beneficial to perform a brief SFT phase before applying RL, using the resulting model as the initialization point. We hypothesize that this short SFT stage helps the model learn the appropriate output format, terminology, and level of detail required for effective concept extraction.

## 8 Conclusion

In this work we showed that VLMs can be adapted into concept bottleneck systems by exploiting their pretrained multimodal understanding. LangCBM relies on the model's existing ability to describe visual content in language and treats these generated descriptions as interpretable concept representations. This approach preserves the key advantages of concept bottleneck models (transparency, intervention, and conceptual evaluation) while operating largely within a standard VLM framework.

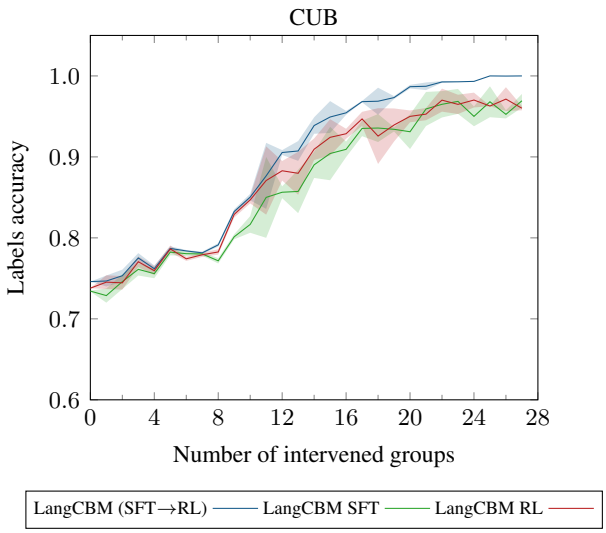


Figure 5: Change in target prediction accuracy after intervening on concept groups between different VLM training modes

Empirically, we found that (1) randomly ordering concept descriptions during training prevents mode collapse on imbalanced datasets, without this, performance drops catastrophically (see results on CUB and AwA2 with fixed ordering); and (2) combining a short format-learning SFT stage with reinforcement learning yields better results than either method alone. Reinforcement learning helps the model express concept knowledge presents in its parameters while keeping the updates smaller and more evenly distributed, suggesting that they may make better use of the pretrained structure. LangCBM achieves 95.9% concept accuracy on CUB and supports human interventions as effectively as alternative CBMs, while being simpler to implement through standard fine-tuning of existing VLMs.

## 9 Limitations

CBMs are designed to enhance collaboration between human experts and AI tools and should ideally be evaluated through user studies. As most previous work on CBMs, due to cost and time constraints, we did not conduct such studies in this work, and instead relied on automatic metrics. Our focus was on fine-tuning, which may not always be the best approach depending on the available infrastructure. Using alternatives to fine-tuning, such

as in-context learning, are possible in LangCBM but were not explored here.

## References


Danis Alukaev, Semen Kiselev, Ilya Pershin, Bulat Ibragimov, Vladimir Ivanov, Alexey Kornaev, and Ivan Titov. 2023. Cross-modal conceptualization in bottleneck models. *EMNLP*.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *Preprint*, arXiv:2501.17161.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. 2022. Addressing leakage in concept bottleneck models. *Advances in Neural Information Processing Systems*, 35:23386–23397.

Hangzhou He, Lei Zhu, Kaiwen Li, Xinliang Zhang, Jiakui Hu, Ourui Fu, Zhengjian Yao, and Yanye Lu. 2025. Chat-cbm: Towards interactive concept bottleneck models with frozen large language models. *Preprint*, arXiv:2509.17522.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Aya Abdelsalam Ismail, Tuomas Oikarinen, Amy Wang, Julius Adebayo, Samuel Stanton, Taylor Joren, Joseph Kleinhenz, Allen Goodman, Héctor Corrada Bravo, Kyunghyun Cho, and Nathan C. Frey. 2024. Concept bottleneck language models for protein design. *Preprint*, arXiv:2411.06090.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. 2023. Probabilistic concept bottleneck models. volume 202 of *Proceedings of Machine Learning Research*, pages 16521–16540. PMLR.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR.

Nicholas Kurtansky, Veronica Rotemberg, Maura Gillis, Kivanc Kose, Walter Reade, and Ashley Chow. 2024. Isic 2024 - skin cancer detection with 3d-tbp. `https://kaggle.com/competitions/isic-2024-challenge`. Kaggle.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. 2023. Label-free concept bottleneck models. In *International Conference on Learning Representations*.

Renyi Qu and Mark Yatskar. 2024. Llm-based hierarchical concept decomposition for interpretable fine-grained image classification. *Preprint*, arXiv:2405.18672.

Neel Rajani, Aryo Pradipta Gema, Seraphina Goldfarb-Tarrant, and Ivan Titov. 2025. Scalpel vs. hammer: Grpo amplifies existing capabilities, sft replaces them. *ArXiv*, abs/2507.10616.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. 2024. Concept bottleneck large language models. *Preprint*, arXiv:2412.07992.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2024. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. Cub-200-2011. Technical Report CNS-TR-2011-001, California Institute of Technology.

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. 41(9):2251–2265.

An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. 2023. Learning concise and descriptive attributes for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3090–3100.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. 2023. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. 2022. Concept embedding models.

# A Sample full-length generations

```
<image>
How would you describe this animal?
```

Listing 1: Input prompt. <image> token will be replaced by CLIP's embedding inside of the decoder

```
This animal is not black, not white, not blue,
    brown, gray, not orange, not red, not yellow
    , not patches, not spots, not stripes, furry
    , not hairless, not toughskin, not big,
    small, not bulbous, not lean, not flippers,
    not hands, not hooves, pads, paws, not
    longleg, not longneck, tail, chewteeth, not
    meatteeth, buckteeth, not strainteeth, not
    horns, claws, not tusks, not smelly, not
    flys, hops, not swims, not tunnels, walks,
    fast, not slow, not strong, not weak, not
    muscle, bipedal, quadrapedal, active, not
    inactive, not nocturnal, hibernate, agility,
     not fish, not meat, not plankton,
    vegetation, not insects, forager, not grazer
    , not hunter, not scavenger, not skimmer,
    not stalker, newworld, oldworld, not arctic,
     not coastal, not desert, not bush, not
    plains, forest, not fields, not jungle, not
    mountains, not ocean, ground, not water,
    tree, not cave, not fierce, timid, not smart
    , not group, solitary, nestspot, not
    domestic.
```

Listing 2: Response

# B More details on baselines

For CBM, ProbCBM, and CEM implementations, we re-trained the models using ViT backbone with the same LoRA reparameterization as in our main experiments

## B.1 Hard AR CBM

Introduced by Havasi et al. (2022), Hard AR CBM (Autoregressive Concept Bottleneck Model) modifies the standard CBM architecture by making concept predictions autoregressive. When predicting the $(N+1)$-th concept, the model incorporates the predictions of the previous $N$ concepts. This creates a dependency chain where each subsequent concept prediction is conditioned on all previously predicted (binary) concepts.

## B.2 Zero-shot LLaVa

For this baseline, we prompt non-tuned LLaVa with questions about individual concept groups (as it cannot follow the format we defined for training).

# C LLM-as-Judge Prompt

Example for the CUB dataset

```
You are a strict and precise evaluator. Your
    task is to determine if the "Model's Output"
     verifiably confirms the "Ground Truth"
    attribute.

## Evaluation Rules:
```

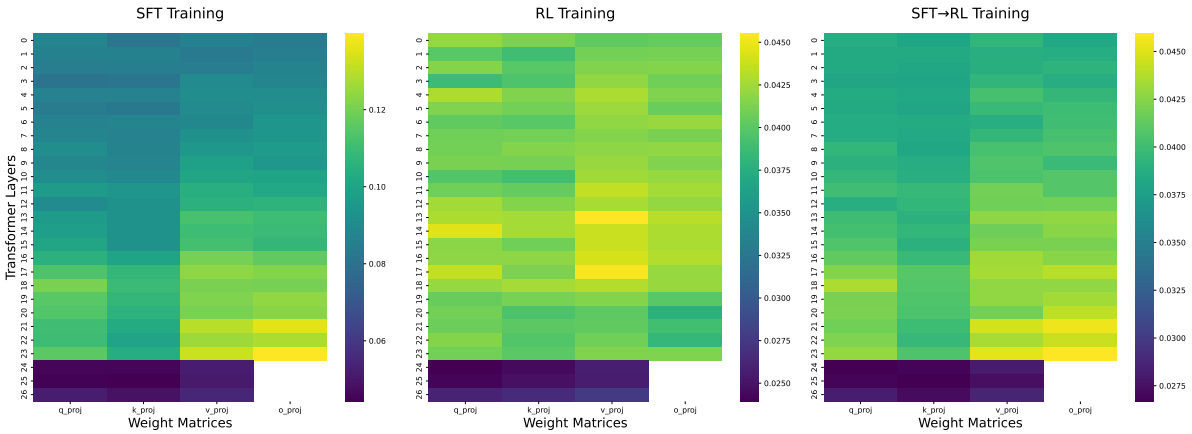


Figure 6: Change in LoRA weight norms compared to init weights when training with SFT and RL

```
1. "yes": The output must explicitly state the
     attribute OR use a clear, common synonym.
     The entire ground truth concept must be
     present.
2. "no": The output is a "no" if it contradicts
     the attribute, is too vague, or omits a key
     descriptive word from the ground truth.

## Examples:
- Ground Truth: yellow belly
- Model's Output: "The bird's underside is
     bright yellow."
- Correct Answer: yes
- Reason: "Underside" is a clear synonym for "
     belly".

- Ground Truth: short, conical beak
- Model's Output: "This bird has a beak for
     eating seeds."
- Correct Answer: no
- Reason: The output confirms a "beak" but omits
      the key descriptors "short" and "conical".

- Ground Truth: black head
- Model's Output: "The top of its head is white
     ."
- Correct Answer: no
- Reason: The output directly contradicts the
     ground truth.

Carefully apply these rules. Your final response
      must be ONLY "yes" or "no", with no
     additional text, explanation, or punctuation
     .
```

Listing 3: Response

# D Sample VLM Generations

We provide sample generations from our VLM-based CBMs for selected concept groups on the CUB dataset under different training regimes: SFT, SFT→RL, RL, and zero-shot settings.

## D.1 Concept Group: beak size

### D.1.1 SFT Model

This bird's beak is longer than head

### D.1.2 RL Model

The bird in this image has a very long bill

### D.1.3 SFT→RL Model

This bird has a large beak

### D.1.4 Zero-shot Model

The bird has a distinctive beak

## D.2 Concept Group: upper body color

### D.2.1 SFT Model

This bird's upper body color is black

### D.2.2 RL Model

The bird on this image is mostly dark

### D.2.3 SFT→RL Model

The bird on this image has a dark upper body

### D.2.4 Zero-shot Model

The bird has a brown back color