Simulating Concept Bottlenecks Using Chain-of-Thought Reasoning

Anonymous ACL submission

Abstract

In high-stakes domains like 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 providing interpretable concepts – typically from an image – before making the final prediction. This allows experts to validate and correct these intermediate concepts. In this paper, we show how CBMs can be effectively implemented using (Vision-)Language Models by leveraging their chain-of-thought reasoning. We fine-tune the model with the standard cross-entropy loss, and our approach maintains prediction quality and achieves high accuracy for intermediate concepts, effectively simulating CBMs without any architectural modifications. We demonstrate the effectiveness of our method on synthetic and real-world datasets, showing that it matches or exceeds the performance of traditional CBMs. Our method not only simplifies the implementation of CBMs but also leverages the extensive knowledge of VLMs acquired during pretraining.

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 aim to provide inherent transparency rather than requiring users to rely 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.

While constructing a CBM typically assumes the reliance on specialized architectures and training procedures, we show that the concept bottleneck can be effectively emulated within a chain-of-thought framework using a vision-language model (VLM). Specifically, in our approach (**CB-CoT**), 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). Note that this second-stage model does not have access to the image. We demonstrate that this architecture, when fine-tuned with standard cross-entropy loss, not only maintains the quality of final predictions but also achieves high accuracy in predicting intermediate concepts and enables intervention, similar to CBMs.

While recent works (e.g., (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 modification increase architectural complexity and training overhead; this additional complexity makes real-world deployment harder. In contrast, we show that such modifications are unnecessary: a vision-language model can learn to predict concepts through standard supervised fine-tuning (SFT), without requiring architectural changes or custom objectives.[1]

2 Background and Related Work

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

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

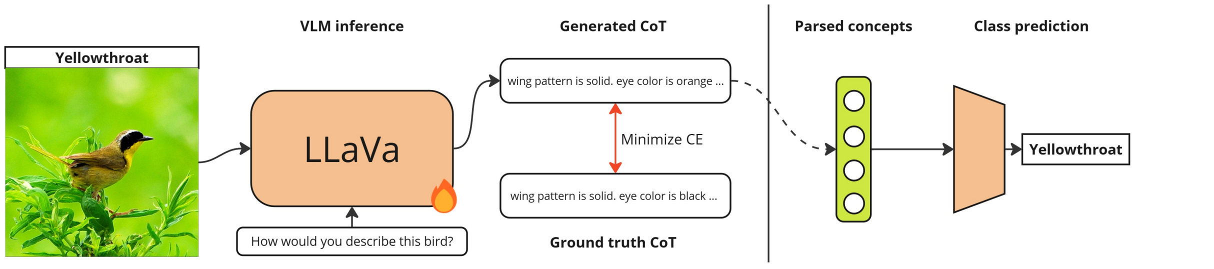

Figure 1: Pipeline of Concept Bottleneck within Chain-of-Though, CB-CoT

$g : \mathbb{R}^D \to \mathbb{R}^k$ maps inputs to concept space and $f : \mathbb{R}^K \to \mathbb{R}$ maps concepts to predictions, requiring supervised concept labels during training.

Havasi et al. (2022) enhance CBMs through residual connections and side-channel models, addressing concept inter-dependencies and relaxing the Markov assumption on concepts while maintaining the advantages of hard CBMs over soft variants. 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, whereas we rely on (V)LMs as building blocks in our pipeline.

CB-LLMs (Sun et al., 2024) integrate CBMs into language processing domain, implementing concept bottlenecks for text classification and generation. 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. While they focus exclusively on text processing, our approach extends to multiple modalities.

In (Oikarinen et al., 2023; Yang et al., 2023; Qu and Yatskar, 2024), LLMs and VLMs are used to generate concepts sets and annotations. Our approach differs in that we instead rely on concept-annotated datasets and demonstrate that a VLM's generation can itself serve as a CBM. In fact, their work is orthogonal to ours: concept annotations for our method could be generated using their approaches.

3 Methodology

We present a simple, yet effective method for implementing CBMs with VLMs without architectural modifications. Our approach, CB-CoT, builds on LLaVa 7B 1.5 (Liu et al., 2023b,a, 2024), which in turn uses CLIP for image embedding and LLaMa (Touvron et al., 2023) for text generation.

CB-CoT 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 convert these binary annotations into natural language statements. For each concept, we generate a sentence indicating its presence or absence, e.g., "The bird has a yellow throat" for positive cases and "The bird does not have a yellow throat" for negative ones. These sentences are concatenated in random order to form the target output for each training example.

The training process consists of two stages. First, we fine-tune a chain-of-thought (CoT) generator by conditioning the VLM on input images with a fixed prompt (e.g., "Describe this animal"). The model learns to generate natural language descriptions that explicitly mention the presence or absence of each concept, optimized using standard language modeling loss (cross-entropy). The second stage is done after a VLM has been trained: we use its concepts predictions to train a classifier for labels, just as sequential CBMs do. Hyperparameters are listed in the Appendix B

To extract structured concept predictions from the model's free-form text generations, we segment the generated text into individual sentences. Each sentence is then mapped back to a binary concept prediction based on whether it indicates the presence or absence of the corresponding concept.

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

4 Datasets

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

4.1 Synthetic lines dataset

We randomly chose the number of lines to draw on the image, uniformly between 1 and

3. The angle and offseet 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. The task is intentionally simple and serves as a controlled setting to evaluate our method.

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

4.3 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 Experiments and results

5.1 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).

5.2 Benchmarks

We evaluated our CB-CoT model against several state-of-the-art concept bottleneck approaches on the CUB-200-2011 and AwA2 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 results reported by (Kim et al., 2023) to ensure fair comparison.

Table 1 presents both concept prediction accuracy and label prediction accuracy for all models. We compare against ProbCBM (Kim et al., 2023),

Dataset	Model	concepts	labels
CUB	Black-box	–	91.9±0.2
	CBM	95.6±0.1	70.8±0.6
	ProbCBM	95.6±0.1	71.8±0.6
	CEM	95.4±0.1	**75.9±0.2**
	Hard AR CBM	95.7±0.1	75.4±0.1
	Zero-shot LLaVa 1.5	3.0±0.1	5.0±0.2
	CB-CoT (this work)	95.7±0.1	73.4±0.1
AwA2	Black-box	–	89.3±0.0
	CBM	97.5±0.0	87.7±0.4
	ProbCBM	97.5±0.0	88.0±0.2
	CEM	97.9±0.1	**88.4±0.2**
	Zero-shot LLaVa 1.5	1.3±0.1	2.1±0.3
	CB-CoT (this work)	97.6±0.1	87.8±0.2

Table 1: Concept prediction and label prediction accuracies on CUB and AwA2

CEM (Zarlenga et al., 2022) and Hard AR CBM (Havasi et al., 2022). See Appendix C for details about baselines.

Our approach achieves competitive performance, maintaining high concept prediction accuracy while demonstrating strong label prediction performance. The drop in performance with respect to Black-box is expected and consistent with the literature. The black-box can rely on non-interpretable information, which, in practical application, can include non-robust features or shortcuts.

To justify the fine-tuning stage, 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. We found that randomizing the order of sentences for each image in training significantly improved the model's performance.

5.3 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 2 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).

More specifically, following (Koh et al., 2020; Havasi et al., 2022; Kim et al., 2023), we perform interventions on semantically grouped concepts rather than individual concept predictions. For ex-

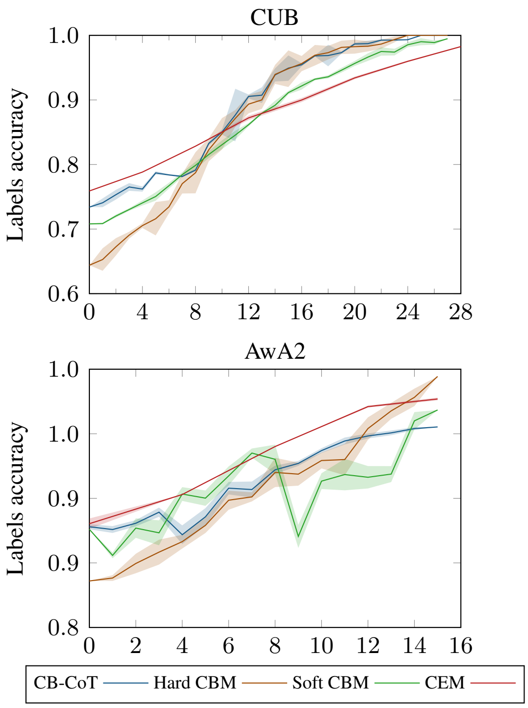

Figure 2: Change in target prediction accuracy after intervening on concept groups

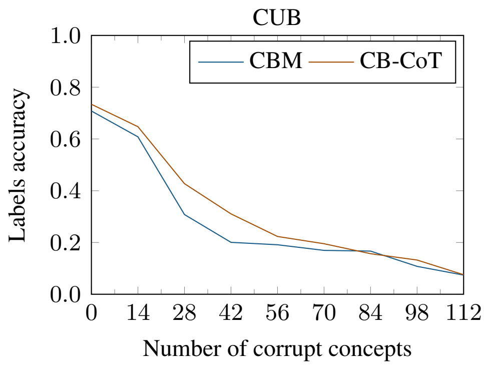

Figure 3: Decrease in target prediction accuracy as more and more concepts are replaced with random noise

ample, instead of intervening on specific predictions like "wing_color::red" or "wing_color::blue" separately, we intervene on the entire "wing color" group of concepts together. We randomly select a number of these concept groups for intervention in each trial. Figure 2 demonstrates how prediction accuracy changes with the number of intervened groups, with error bars showing the standard deviation across 5 random samples of group selections. As shown in the plots, CB-CoT's classification accuracy consistently improves as we increase the number of intervened concept groups. In fact, our model's intervention behavior is fairly close to that of hard sequential CBM across both CUB and AwA2 datasets.

5.4 Concept leakage analysis

To investigate potential concept leakage in our model, we trained our models on corrupted concept sets, where individual concepts were replaced with random Bernoulli noise. These concepts were then used to produce sentences describing concepts, with the same procedure as for the original data. Figure 3 illustrates how classification accuracy declines for both the standard CBM and our CB-CoT as the number of corrupted concepts increases.

In theory, a model might compensate for missing concept information by encoding it elsewhere. However, as we empirically confirm here, the reliance on (V)LMs and the absence of joint training make this unlikely. The observed drop in performance suggests that both models primarily depend on the intended concept information for their predictions, rather than bypassing the concept bottleneck through unintended leakage.

This finding complements our intervention analysis, in which we demonstrate that an increase in the number of intervened concept groups leads to improved performance. Together, these results suggest that CB-CoT exhibits minimal concept leakage. If significant information leakage was present, we would expect resistance to concept corruption (Figure 3) or limited benefit from concept interventions. Instead, we observe both strong sensitivity to concepts quality and substantial benefits from interventions, suggesting minimal concept leakage.

6 Conclusion

We have demonstrated that CBMs can be effectively implemented with a combination of VLM and LLM, without requiring architectural modifications or custom loss functions. Our approach, CB-CoT, leverages a simple fine-tuning process to enable (V)LMs to predict interpretable intermediate concepts, preserving high classification accuracy while allowing human intervention and correction. We show that CB-CoT closely matches or outperforms traditional CBMs while simplifying implementation. We further validate its effectiveness by analyzing intervention impact and concept leakage, confirming that our method maintains the core advantages of CBMs, such as robustness to corrections and reliance on interpretable representations.

7 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 CB-CoT but were not explored here. While our experiments and overall architecture do not make substantial information leakage likely, a more thorough investigation would be needed to confirm this.

References

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

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.

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

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

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

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.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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.

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 chains of thought

```
<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 Training details and hyperparameters

We fine-tune LLaVa 7B 1.5 using LoRA (Hu et al., 2022) adapters with $r = 128, \alpha = 256$. We set $weight\_decay = 0.001, lr = 0.0002$ and train until convergence on validation loss with $batch\_size\_per\_device = 8$. We employ cosine scheduling for the learning rate with first 3% iterations spent on warmup. We chose LoRA rank and learning rate based on the scripts from LLaVa repository

C More details on baselines

For CBM, ProbCBM, and CEM implementations, we utilize the results as reported by Kim et al. (2023) on both CUB and AwA2 datasets.

C.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.

C.2 Black-box

For this baseline, we adopted the architecture and hyperparameters from Kim et al. (2023)'s CBM implementation, using InceptionV3 as the backbone network. The key distinction is that this model is trained only with cross-entropy loss on the final class labels, without any intermediate concept supervision.

C.3 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).