CLIP-Free, Label Free, Unsupervised Concept Bottleneck Models

Supplementary Material

1. Limitations

As with any research work, our study has its own limitations that should be transparent and acknowledged. In particular, we identify a primary limitation of our method concerning wrong semantic associations of class names in CBMs, due to polysemy. This issue is closely tied to the choice of concept set used. When using the 20K most common words in English as our concept set, we observe that class names get associated to wrong semantically-related concepts. For example, the bird "drake" is linked to artist-related concepts (Rihanna, Robbie, lyric). This occurs because the bird "drake" is less familiar to the text encoder than the artist "drake". In fact, a google search with the word "drake" yields directly the artist rather than the bird. As another example, the top-detected concepts for the prediction "african grey" are incorrect semantic associations with the word "african" (ethiopian, tanzania) and the word "grey" (purple, blue) that do not pertain to the bird itself. We also observe similar cases that may raise ethical concerns (e.g., the animal "cock" leads to associations with male reproductive terms). However, it is worth noting the following:

1. Meaningful concepts such as "duck" (first example) are still detected among the top concepts.
2. The incorrect semantic associations contribute only a negligible portion of the total logit, accounting for approximately 0.01% of the overall prediction score.
3. This issue is considerably less severe when using alternative concept sets, such as the LF-CBM concept set tailored for ImageNet.
4. This issue also appears in CLIP-based CBMs and hence not unique to our approach.

2. Ablation Studies on the Text Encoder

In Table 1, we present ablation studies using other text encoders from the Sentence-BERT library [43] with the ResNet50 visual classifier. We observe that the choice of the text encoder has minimal effect on the performance. This is because even lower-performing text encoders are capable of understanding class names.

Text Encoder	Top-1 (%)
DistilRoberta	75.73
MPNet-Base	75.78
MPNet-Base-MultiQA	75.76
MiniLM	**75.80**

Table 1. Ablation studies on other text encoders

3. Ablation Studies of the MLP

3.1. Role and Impact of the MLP

We first evaluate the role and impact of the MLP to verify its contribution and that it learns meaningful transformations, and does not collapse into a trivial function. We perform the following ablation studies: **1) Mean Ablation:** For an image, we replace the input features to the MLP with a constant mean of the features calculated across the full ImageNet validation set. **2) Random Features:** For an image, we replace the input features to the MLP with random values sampled from a normal distribution with a mean and standard deviation equal to that of the features calculated across the full ImageNet validation set. **3) Random Weight Ablation:** We randomize the weights of the MLP projection. **4) Shuffled Ablation:** For an image, we replace the input features to the MLP with input features of another random image in the validation dataset.

For each ablation experiment, we compute the ImageNet validation accuracy. We expect the accuracy to drop across all ablations. As shown in Table 2, the accuracy nearly drops to zero in every case, confirming the effectiveness of our MLP.

3.2. MLP Design

Next, we perform an ablation study over the number of layers and output dimension scaling factor (`dim_out_factor`) of the MLP. `dim_out_factor` specifies how much to scale the previous layer's dimensionality. For example, in a ViT-B/16 model the hidden dimension is 768; setting `dim_out_factor = 2` therefore expands the projection from 768 to $2 \times 768 = 1536$. Results are shown in Table 3 for a ResNet50 model. We observe that the 2 layers and a output dimension scaling factor of 2 provides the best results.

4. Concept Interventions and other Explainability Metrics

4.1. Concept Interventions

We report concept intervention results on our ZS-CBMs. Interventions on CBMs are an effective tool to mitigate biases, debug models and fix their reasoning by explicitly intervening in the concepts of the bottleneck layer to control predictions. The Waterbirds-100 dataset [44] is a standard dataset used in previous works [42] to conduct CBM intervention experiments. It is a binary classification dataset of two classes: waterbirds and landbirds. The training images of waterbirds are on water backgrounds, and training images

Model		Mean Feature ↓	Random Features ↓	Shuffled Features ↓	Random Weights ↓
ResNet101v2	Ours	81.49	81.49	81.49	81.49
	Ablated	**0.10**	**0.11**	**1.70**	**0.11**
ConvNeXt-Base	Ours	83.88	83.88	83.88	83.88
	Ablated	**0.10**	**0.11**	**1.79**	**0.10**
BeiT-L/16	Ours	87.22	87.22	87.22	87.22
	Ablated	**0.10**	**0.11**	**1.87**	**0.11**
DINOv2-B	Ours	84.40	84.40	84.40	84.40
	Ablated	**0.10**	**0.13**	**1.76**	**0.09**

Table 2. Ablation studies of the MLP.

Layers	**dim_out_factor**	**Top-1 (%)**
1	1	72.48
1	2	74.01
2	1	75.41
2	2	75.80

Table 3. Performance comparison of the MLP for different layer and dimension configurations.

of landbirds are on land backgrounds. However, the validation images do not have that correlation, where waterbird images appear on land backgrounds and landbird images on water backgrounds. The model is assumed to learn the water–land background correlation to perform this classification task. By building a CBM, we can correct this bias by intervening in concepts in the CB layer. We curated a validation dataset of waterbirds/landbirds directly from the ImageNet validation set, including 140 validation images (70 for each class). We create our ZS-CBM using the two class prompts:“an image of a waterbird" (for the waterbird class) and “an image of a landbird" (for the landbird class), and using the same concept set from [42] which includes a collection of bird-related concepts and a collection of land-related concepts. The ZS-CBM achieves a low accuracy, as shown in Table 4, which indicates the water-land bias the model performs for classification. To correct this, we intervene in the concepts in the CB layer, following the setup from [42]. **Intervention R (Int. R):** We zero-out activations of any bird concepts from the bottleneck layer, and expect the accuracy to drop. The larger the drop, the better. **Intervention K (Int. K):** We keep activations of bird concepts as they are, but scale down the activations of all remaining concepts by multiplying them with a factor of 0.1, and expect the accuracy to increase. More increase is better. Results are presented in Table 4 for some models, and demonstrate the success of our intervention experiments.

We also conduct concept intervention experiments with a more challenging multi-class setup. Unlike the Waterbirds dataset where bias-correlation issues are assumed, we make no such assumption here. We instead evaluate whether intervening on class-related concepts in the bottleneck layer and zeroing out their activations, impairs model accuracy. A drop in accuracy indicates that these concepts are important for prediction. We select a subset of 10 classes from ImageNet following [15]. We use the original ImageNet validation images for those classes, rather than the validation set from [15], because the latter contains original ImageNet training examples that our model has already seen. We employ an LLM to generate 5 highly-relevant concepts for each class, achieving in total 50 concepts. The classes and concepts we used are provided in Section 16 of the supplementary material. We then measure CBM accuracy before and after intervention. Results are provided in Table 5. For each image, we zero out the activations of its class-related concepts and report the **Intervention R (Int. R)** metric. As shown in Table 5, this intervention reduces accuracy by approximately 20% on average, underscoring the concepts importance to the CBM.

Model	Orig. CBM	Int. (R)↓	Int. (K)↑
BeiT-B/16	54.29	41.43 **(-12.86)**	58.57 **(+4.28)**
ConvNeXtV2$_{pt}$@384	53.57	42.14 **(-11.43)**	59.29 **(+5.72)**
ConvNeXt_B$_{pt}$	53.57	42.86 **(-10.71)**	58.57 **(+5.00)**
DiNOv2	52.86	43.57 **(-9.29)**	59.29 **(+6.43)**
BeiT-L/16	52.86	44.29 **(-8.57)**	58.57 **(+5.71)**

Table 4. CBM Interventions on the standard Waterbirds dataset

Model	Orig. CBM	Int. (R)↓
ResNet50	96.80	76.00 **(-20.8)**
ResNet101	97.00	76.80 **(-20.2)**
DINOv2-B	98.60	78.40 **(-20.2)**
BeiT-B/16	98.60	78.80 **(-19.8)**
ConvNeXtV2$_{pt}$@384	99.40	79.40 **(-20.0)**

Table 5. CBM Interventions in a multi-class setup

4.2. Concept Discovery Accuracy

We use the discrete-based Mutual Information (MI) metric[2] to calculate the MI between concepts detected by our method, and those detected by CLIP. Specifically, given the concept bank, we use the CLIP Vision Encoder to encode an image, and the CLIP Text Encoder to encode the concept bank. We record the set of top-K detected concepts most similar to the image. We then encode the same image with our TextUnlocked visual classifier (e.g., DINOv2)(including the MLP) and encode the same concept bank with our Text Encoder T. We record the set of top-K detected concepts most similar to the image. We then calculate the MI between the detected concepts from CLIP (CLIP ViT-B/16 model) and our method. Results are shown in Table 6 and demonstrate high mutual information which indicates that our method successfully detects key concepts in the dataset by only learning from the limited vocabulary of class labels.

Model	NMI
DINOv2	0.681
EfficientNetv2-M	0.704
ViT-B/16v2	0.711

Table 6. Normalized Mutual Information (NMI) with CLIP ViT-B/16 for different models. NMI ranges from 0 (no mutual information) to 1 (perfect agreement).

5. Transformed Classifier Results on Other Datasets

We conduct extra experiments on 3 datasets. Places365 (domain-specific to scenes), DTD (domain-specific to texture and fine-grained), and EuroSAT (domain-specific to satellite images). For each dataset, we show the top-1 accuracy of the classifier's original performance and the top-1 accuracy of our transformed classifier. Results are shown in Table 7.

Dataset	Model	Original (%)	Transformed (%)
Places365	ResNet50	54.77	53.90
	DenseNet161	56.13	55.54
EuroSAT	ResNet50	93.95	94.23
	WideResNet101	93.78	93.88
	ViT-B/16	93.67	93.53
DTD	ResNet50	69.47	69.26
	WideResNet101	68.62	68.14
	ViT-B/16	69.95	69.68

Table 7. Results of our transformed classifier on Places365, EuroSAT, and DTD datasets

[2]https://github.com/fawazsammani/clip-interpret-mutual-knowledge

6. Prompt Variations

We evaluate the robustness of our transformed classifier to variations in text prompts. Using the transformed ViT-B/16, we provide a diverse set of prompts and measure the Top-1 accuracy on ImageNet. Results are shown in Table 8. We order the prompts from highest to lowest scoring. We find that our transformed classifier is robust to text prompts. The worst prompt degrades the baseline accuracy by only 0.36 points

Prompt	Top-1 Accuracy (%)
an image of a {}	80.70
a photo of one {}.	80.67
a photo of a {}.	80.66
a close-up photo of a {}.	80.63
a black and white photo of a {}	80.63
a close-up photo of the {}.	80.62
a cropped photo of a {}.	80.61
a good photo of a {}.	80.61
a dark photo of a {}.	80.61
a bright photo of a {}.	80.60
a bright photo of the {}.	80.59
a bad photo of a {}.	80.57
a blurry photo of a {}.	80.57
a pixelated photo of a {}.	80.55
a photo of many {}.	80.54
a low-resolution photo of the {}	80.54
a low-resolution photo of a {}.	80.52
a photo of my {}.	80.45
a jpeg-corrupted photo of a {}.	80.34

Table 8. Effect of different text prompts on Top-1 accuracy using ViT-B/16.

7. Other Concept Sets

Early works such as label-free CBMs use LLMs to automatically generate a set of concepts for a target class. In recent works such as DN-CBM [42] and DCBM [18], it was however shown that general, pre-defined concept sets outperform the LLM-generated ones, which introduce many spurious correlations and hallucinated associations. In Table 9, we compare general pre-defined and LLM-generated concept sets. Across different types of architectures, general, pre-defined concept sets outperform the LLM-generated ones by several percent. Please note that while earlier works such as LF-CBMs employ LLM-generated concept sets (which underperform compared to the general, pre-defined concept set), all results reported in our main manuscript—including those from prior works—are presented using the general, pre-defined concept set to ensure a fair comparison.

Model	Predefined	LLM-generated
Swinv2-Base	82.60	80.34
ConvNext-Base	82.80	80.24
ConvNeXtV2-B_{pt}@384	86.40	84.03
ViT-B/16	79.30	76.37

Table 9. Comparison of Predefined and LLM-generated concept sets on CBMs

8. Implementation Details

For the text encoder, we use the all-MiniLM-L12-v1[3] model available on the Sentence Transformers library [43]. This text encoder was trained on a large and diverse dataset of over 1 billion training text pairs. It contains a dimensionality of $m = 384$ and has a maximum sequence length of 256. Our MLP projector is composed of 3 layers, the first projects the visual feature dimensions n to $n \times 2$ and is followed by a Layer Normalization [2], a GELU activation function [14] and Dropout [47] with a drop probability of 0.5. The second layer projects the $n \times 2$ dimensions to $n \times 2$ and is followed by a Layer Normalization and a GELU activation function. The final linear layer projects the $n \times 2$ dimensions to m (the dimensions of the text encoder). We train the MLP projector with a batch size of 256 using the ADAM optimizer [17] with a learning rate of 1e-4 that decays using a cosine schedule [28] over the total number of epochs. We follow the original image sizes that the classifier was trained on. For the training images, we apply the standard image transformations that all classifiers were trained on which include a Random Resized Crop and a Random Horizontal Flip. For the validation images, we follow exactly the transformations that the classifier was evaluated on, which include resizing the image followed by a Center Crop to the image size that the classifier expects. Each model is trained on a single NVIDIA GeForce RTX 2080 Ti GPU.

9. Compositional Image Captioning

Compositional Image Captioning is an old image captioning paradigm termed [20], later revived with deep learning methods [29]. In compositional captioning, a set of image-grounded concepts (such as attributes, objects and verbs) are first detected, and a language model is then used to compose them into a natural sounding sentence. With the current advancements of Large Language Models (LLMs) and their powerful capabilities, we use an LLM as a compositioner. Specifically, we detect the top concepts and verbs to the image using the concept discovery method introduced in Section 3.2 and shown in Figure 2(a), and feed them, along with their similarity scores, to an LLM. For the concepts, we use the same concept set as in Section 3.2. We also add a list of the most common verbs [7] in English to the pool. This allows us to cover all possible words and interactions. We prompt the LLM to utilize the provided information to compose a sentence, given a limited set of in-context examples from the COCO captioning training set (in our experiments, we use 6 examples). This allows us to generate sentences adhering to a specific style and structure of the in-context examples. For this experiment, we used GPT4o-mini [35] as our LLM, as it is fast and cost-efficient. In the prompt, we explicitly instruct the LLM to refrain from reasoning or generating content based on its own knowledge or assumptions, and that all its outputs must be strictly grounded in the provided concepts, verbs, and score importance. We use the following prompt:

```
I will give you several attributes
and verbs that are included in an image,
each with a score.  The score reflects
how important (or how grounded) the
attrribute/verb is to the image,
and higher means more important and
grounded.  Your job is to formulate a
caption that describes the images by
looking at the attributes/verbs with
their associated scores.  You should
not reason or generate anything that is
based on your own knowledge or guess.
Everything you say has to be grounded
in the attrbiutes/verbs and score
importances.  Please use the following
the structure, style, and pattern of
the following examples.  Example 1:  A
woman wearing a net on her head cutting
a cake.  Example 2:  A child holding
a flowered umbrella and petting a yak.
Example 3:  A young boy standing in
front of a computer keyboard.  Example
4:  a boy wearing headphones using one
computer in a long row of computers.
Example 5:  A kitchen with a stove,
microwave and refrigerator.  Example
6:  A chef carrying a large pan inside
of a kitchen.  Here are the attributes
and scores:  {detected concepts with
scores}, and these are the verbs and
scores:  {detected verbs with scores}.
```

Results are shown in Table 10.

While the results on CiDEr and SPICE are incremental compared to the results in Table 4, the n-gram metrics (B4, M and R-L) are boosted, which verifies our hypothesis that the low scores of B4 and M were attributed to the specific caption style and structure of the respective dataset (COCO).

[3] https://huggingface.co/sentence-transformers/all-MiniLM-L12-v1

Model	B4	M	R-L	C	S
ZeroCap	2.6	11.5	—	14.6	5.5
ConZIC	1.3	11.5	—	12.8	5.2
Ours					
DenseNet161	4.20	12.5	30.1	17.0	6.6
ResNet50	4.10	12.5	30.1	17.0	6.6
$\text{ResNet50}_{\text{v2}}$	4.50	12.8	30.5	18.4	6.9
$\text{WideResNet50}_{\text{v2}}$	4.20	12.7	30.3	17.7	6.9
ResNet101v2	4.30	12.6	30.1	18.0	6.8
$\text{ConvNeXt-Base}_{\text{v2}}$	4.40	12.8	30.2	18.6	7.1
$\text{ResNet50}_{\text{v2}}$	4.50	12.8	30.5	18.4	6.9
EfficientNetv2-S	4.40	12.7	30.4	18.6	6.9
$\text{ViT-B/16}_{\text{pt}}$	4.50	12.8	30.2	18.7	**7.2**
$\text{ConvNeXtV2-B}_{\text{pt}}$@384	4.40	12.7	30.2	18.7	**7.2**
BeiT-B/16	4.50	12.8	30.3	**18.9**	7.1
DINOv2-Base	**4.60**	**13.0**	**30.7**	18.7	7.1

Table 10. Zero-Shot Compositional Captioning Performance

9.1. Domain-Specific Compositional Captioning

Since the LLM we used for compositional image captioning is generic and can adapt to any style and structure, compositional captioning is especially useful for generating captions tailored to specific domains. We explore alternative concept sets in Compositional Captioning. In the main manuscript, we reported results using the 20,000 most common English words as our concept set. Since the LLM remains fixed and functions as a composer, integrating detected concepts and verbs grounded in the image into a caption, we can seamlessly substitute the concept set with any domain-specific concept set alternative. This allows for the generation of captions tailored to a specific domain. Here, we maintain the same set of verbs but explore the use of concepts specific to the ImageNet dataset. Since ImageNet lacks dedicated captions, we evaluate the domain-specific captioning by anticipating a decline in performance on the COCO captioning dataset. We use the ImageNet-specific concept set from [34] and report zero-shot captioning performance in Table 11. As shown, we observe a decrease in all metrics. This shows that our method can readily produce captions for any domain. Finally, also note that we can control the style of the generations by simply prompting the LLM to compose the concepts and verbs in a specific style (e.g., humorous, positive, negative).

10. Qualitative examples of zero-shot image captioning.

We present qualitative examples of zero-shot image captioning from different classifiers in Figure 1. We choose BeiT-L/16, ConvNeXtv2, and ViT-B/16. This allows us to see how different classifiers “see" the image. From the first example, BeiT-L/16 captures features of both the vegetables and the dog, whereas ConvNeXtV2 captures only the vegetables. In contrast, ViT-B/16 focuses exclusively on the dog and its characteristics.

Method	B4	M	R-L	C	S
ZeroCap	2.6	11.5	—	**14.6**	5.5
ConZIC	1.3	11.5	—	12.8	5.2
Ours					
MobileNetv3-L	3.50	12.7	29.1	11.4	6.1
ResNet50	3.60	12.7	29.3	12.0	6.0
ResNet101v2	3.50	12.9	29.3	12.2	6.3
WideResNet101v2	3.70	12.9	29.6	12.4	6.2
ConvNeXt-Base	3.80	12.8	29.5	12.7	6.2
EfficientNetv2-S	3.70	12.9	29.6	12.9	6.3
ViT-B/16 (pt)	3.80	13.1	29.5	13.2	6.5
BeiT-L/16	**3.90**	**13.2**	**29.6**	13.4	**6.6**

Table 11. Composition Captioning Performance using the ImageNet-specific LF-CBM concept set

Model	Top-1	Orig.	Δ
$\text{ConvNeXtV2-B}_{\text{pt}}$	86.07	86.25	−0.18
BeiT-B/16	84.54	85.06	−0.52
WideResnet50	78.35	78.47	−0.12
$\text{ViT-L/16}_{\text{v2}}$	87.61	88.06	−0.45
EfficientNetv2-S	84.04	84.23	−0.19
ConvNeXt-Small	83.42	83.62	−0.20
ResNeXt101-32x8d	79.10	79.31	−0.21
$\text{ShuffleNetv2}_{\text{x2.0}}$	75.83	76.23	−0.40
$\text{WideResNet50}_{\text{v2}}$	81.17	81.31	−0.14
DenseNet169	75.46	75.60	−0.14
ConvNeXt-Tiny	82.19	82.52	−0.33
$\text{ConvNeXt-B}_{\text{pt}}$	85.27	85.52	−0.25
$\text{ResNet101}_{\text{v2}}$	81.50	81.68	−0.18
ResNeXt50-32x4d	77.44	77.62	−0.18
ConvNeXt-Base	83.88	84.06	−0.18
$\text{ResNeXt50-32x4d}_{\text{v2}}$	80.79	80.88	−0.09
ViT-B/32	75.40	75.91	−0.51
Swin-Small	82.63	83.20	−0.57
Swinv2-Tiny	81.44	82.07	−0.63
CvT-21	80.45	81.27	−0.82
Swinv2-Small	83.32	83.71	−0.39

Table 12. Performance of our reformulated classifiers for additional models

11. Performance on Additional Models

We report performance on additional models that were not included in the main manuscript in Table 12.

BeiT-L/16: broccoli and vegetables, the dog enjoying a puppy's favorite vegetable
ConvNextv2: broccoli caulifol, cabbage or spinach
ViT-B/16: Gordon, who was born in a dog and Labrador puppy

BeiT-L/16: shower curtain, bathing room and toilet.
ConvNextv2: shower curtain toilet seat and toilets, which were made by the bath.
ViT-B/16: shower curtain in the bathroom, and a small bathtub.

BeiT-L/16: motorcycles with the motorized scooters and electric motors
ConvNextv2: motorbike, scooter and motorcycle scrapping
ViT-B/16: motorcycles and motor scooters, motorcycle engines

Figure 1. Qualitative examples of zero-shot image captioning.

12. Process of Zero-Shot Image Captioning

We remind readers of the mapping function, denoted as MLP, that transforms the visual features f into the same space as textual features, producing $\tilde{f}$. A pre-trained language model G is then optimized to generate a sentence that closely aligns with $\tilde{f}$. To preserve the generative power of G, we keep it frozen and apply prefix-tuning [21], which prepends learnable tokens in the embedding space. We follow a test-time training approach to optimize learnable tokens for each test input on-the-fly. Our method builds upon the work of [49].

A high-level overview of this process is illustrated in Figure 2. Using a pre-trained language model G, we prepend randomly initialized learnable tokens, referred to as prefixes, which guide G to produce text that maximizes alignment with visual features. These learnable prefixes function as key-value pairs in each attention block, ensuring that every generated word can attend to them.

For each iteration j, at a timestep ts, we sample the top-Q tokens from the output distribution of G, denoted as G_{out}, which serve as possible continuations for the sentence. These Q candidate sentences are then encoded by a text encoder T, mapping them into the same embedding space as $\tilde{f}$. We compute the cosine similarity between each encoded sentence and $\tilde{f}$, resulting in Q similarity scores. These scores are normalized with softmax and define a target distribution used to train G_{out} via Cross-Entropy loss. The learnable prefixes are updated through backpropagation.

With the updated prefixes, G is run again, and the most probable token is selected as the next word. This process is repeated for a predefined number of timesteps (up to the desired sentence length) or until the $< . >$ token is generated. At the end of each iteration, a full sentence is generated. We conduct this process for 20 iterations, generating 20 sentences in total. The final output is chosen as the sentence with the highest similarity to the visual features $\tilde{f}$.

We also add the fluency loss from [49] as well as other token processing operations. We refer readers to [49] for more information. We use the smallest GPT-2 of 124M parameters as G. We also noticed that using a bigger G (e.g., GPT-2 medium) does not enhance performance, indicating that a decoder with basic language generation knowledge is sufficient.

13. Concept Filtering

One of our concept filtering procedures requires us to know the terms corresponding to the parent and subparent classes (e.g., "fish" and "animal" for the class "tiger shark"), other species within the same category, and any synonyms of the target class name. In order to obtain this information, we used an LLM (gpt-4o-mini) with the following prompt:

```
Provide your answer to below as
single-word comma separated.  If you
need to provide a term composed of
2 words, then separate each into a
single word.  For each class, provide
its synonyms and closely related
names (e.g., other species of the
category, its superclasses such as
bird, fish, dog, cat, animal...etc).
Here is an example.  The ImageNet
class is:  tench, tinca tinca.  Answer:
fish,animal,cyprinid,carp,vertebrate.
```

14. Using only class names

We remind readers from Section 3 that we only use the class names to formate the text prompt for the text encoder when training the MLP. In practice, we can go beyond class names by using resources like a class hierarchy from WordNet [23] (the original source where ImageNet was extracted from), or class descriptions extracted from a LLM as in CuPL [40] or VCVD[32]. However, this approach would be considered

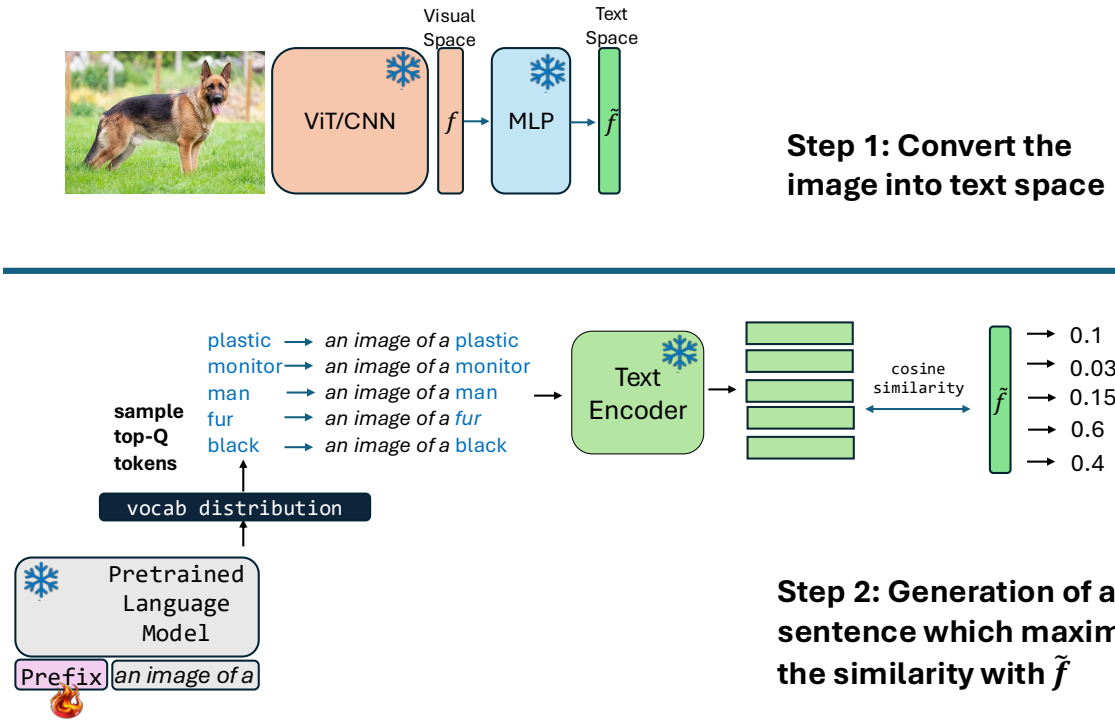

Figure 2. The process used to generate zero-shot captions using any pretrained language decoder (e.g., GPT-2). The process is shown for the first timestep ($ts = 1$) and first iteration ($j = 1$) with a hard prompt set as "an image of a". We apply prefix tuning while keeping the language decoder frozen, generating text that maximizes the similarity with the visual features.

as "cheating. The original classifier implicitly learns the semantics, hierarchies, relationships and distinctive features of different classes. Explicitly providing additional information would not replicate the classifier faithfully since it would force the classifier to focus on predefined features or those we intend it to learn. Moreover, this would also leak information to downstream tasks such as CBMs and textual decoding of visual features, compromising the fairness of evaluation. For instance, if class descriptions were used in the training, the concepts in CBMs would align with those specified in the training prompts. For these reasons, we refrain from using any other additional information than the class names. We use the class names provided from https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a.

15. Global Class-Wise Qualitative Examples

In Figure 3, we present a probability distribution of global class-wise concepts. These are concepts detected for all images of a specific class, along with their frequency. We consider two semantically similar classes but distinctively different: "hammerhead shark" and a "tiger shark". We highlight in yellow the top concepts in "hammerhead shark" that are not present in "tiger shark". These concepts are "harpoon" and "lobster hammer", both which are distinctive to the head of the hammerhead shark and drive its prediction. Note that for this experiment, we do not apply the concept filtering procedure.

16. Multi-class CBM Intervention Classes and Concepts

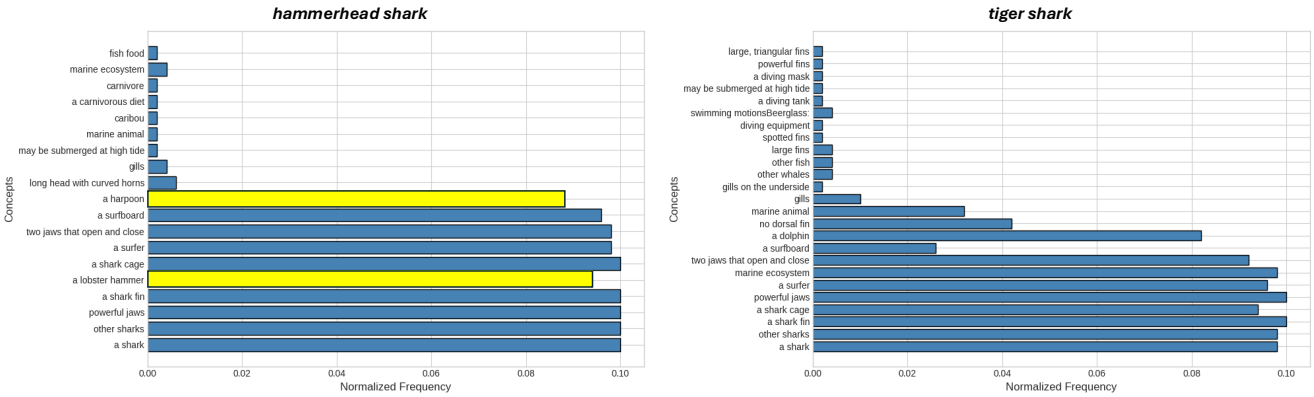

Figure 3. Global class-wise interpretability analysis with our Concept Bottleneck Model. We highlight in yellow the top concepts in "hammerhead shark" that are not present in "tiger shark", and therefore distinctive to "hammerhead shark".

Class	**Concepts**
tench	fish, freshwater, fins, dorsal, olive
english springer	dog, long ears, brown and white, playful, hunting
cassette player	portable, audio, tape, speakers, buttons
chainsaw	sharp, handheld, cutting, metal, wood
church	cross, tower, architecture, sacred, religious
french horn	curved, mouthpiece, musical instrument, orchestral, blow
garbage truck	large vehicle, wheels, clean, high load, lift
gas pump	fueling, hose, metallic, gasoline, handle
golf ball	small, white, round, rubber, dimples
parachute	fabric, fly, air, landing, strings

Table 13. ImageNet classes and their five associated concepts we use in our multi-class CBM intervention experiment.