

# CLIP-Free, Label Free, Unsupervised Concept Bottleneck Models

Fawaz Sammani<sup>1,2,3</sup>, Jonas Fischer<sup>1</sup>, Nikos Deligiannis<sup>2,3</sup>

<sup>1</sup> Max-Planck-Institut für Informatik, Saarbrücken, Germany

<sup>2</sup> ETRO Department, Vrije Universiteit Brussel, Belgium

<sup>3</sup> imec, Kapeldreef 75, B-3001 Leuven, Belgium, Belgium

fawaz.sammani@vub.be, jonas.fischer@mpi-inf.mpg.de, ndeligia@etrovub.be

## Abstract

*Concept Bottleneck Models (CBMs) map dense feature representations into human-interpretable concepts which are then combined linearly to make a prediction. However, modern CBMs rely on the CLIP model to obtain image-concept annotations, and it remains unclear how to design CBMs without the CLIP bottleneck. Methods that do not use CLIP instead require manual, labor intensive annotation to associate feature representations with concepts. Furthermore, all CBMs necessitate training a linear classifier to map the extracted concepts to class labels. In this work, we lift all three limitations simultaneously by proposing a method that converts any frozen visual classifier into a CBM without requiring image-concept labels (label-free), without relying on the CLIP model (CLIP-free), and by deriving the linear classifier in an unsupervised manner. Our method is formulated by aligning the original classifier’s distribution (over discrete class indices) with its corresponding vision-language counterpart distribution derived from textual class names, while preserving the classifier’s performance. The approach requires no ground-truth image–class annotations, and is highly data-efficient and preserves the classifiers reasoning process. Applied and tested on over 40 visual classifiers, our resulting unsupervised, label-free and CLIP-free CBM ( $U-F^2$ -CBM) sets a new state of the art, surpassing even supervised CLIP-based CBMs. We also show that our method can be used for zero-shot image captioning, outperforming existing methods based on CLIP, and achieving state-of-art.*

## 1. Introduction

Visual classifiers predict a class as a linear combination of dense, high-dimensional visual feature vectors that are difficult to interpret by humans. Concept Bottleneck Models (CBMs) [19] address this challenge by mapping these feature vectors into a set of human-interpretable concepts, each associated with an activation score (referred to as a concept activation). Predictions are then made as a linear combi-

nation of these concept activations. Initial CBMs required image–concept annotations to train the bottleneck layer that maps dense features to concepts. Modern CBMs [5, 34, 58] overcome this limitation by leveraging the CLIP model [41] or vision–language grounding models [46] to provide image-concept annotations. Since CLIP models map image and text into a shared embedding space, these approaches can query images against an entire pool of predefined textual concepts and use cosine similarity scores to find a matching annotation. These approaches are commonly referred to as *label-free CBMs*.

In many real-world settings, a high-performing, task-specific *legacy* model already exists and often achieves strong results on the target task [61]. When CLIP is used to generate image-concept annotations, the resulting CBM is anchored to CLIP’s embedding space; the legacy model must be interpreted through CLIP’s notion of similarity, rather than through its own learned representation. In this sense, the approach is not fundamentally different from replacing the legacy model with a CLIP-based CBM. Furthermore, this dependence can transfer CLIP’s biases and behavior into the legacy model (e.g., importing the CLIP typographic bias [10] into a DINO model). A natural question that then arises is: *how to develop a label-free CBM for such legacy specialist models without the CLIP constraint?* Retraining such a specialist on a large image–text corpus following the CLIP approach is impractical, in terms of computational cost and need of a huge amount of image-text data. Furthermore, obtaining ground-truth image-concept annotations through manual human-labor is time consuming and expensive. Finally, retraining this legacy model further alters its original decision-making process and distribution, which is typically not desired.

In this work, we lift these limitations by first proposing a method dubbed as TextUnlock, which aligns a distribution of a frozen visual classifier to its corresponding vision-language counterpart, without relying on CLIP. TextUnlock has four important properties: First, it is *efficient*; it is inexpensive to train and can be performed on any standard hardware, regard-

less of the size of the original classifier. The number of data points is also significantly reduced compared to CLIP-based approaches. Secondly, it is *label-free*, no labels are required to achieve this formulation. Thirdly, TextUnlock is *trained to preserve* the original distribution and reasoning process of the classifier, and does not compromise the classifier’s original performance (average of 0.2 points drop in accuracy). Finally, our method is applicable to *any* vision architecture, whether convolutional-based, transformer-based or hybrid. After the original classifier’s distribution is aligned to its vision-language counterpart distribution with TextUnlock, we simply query the transformed classifier’s image features against a set of predefined text concepts to obtain concept activations for our CBM, and then derive the concept-to-class classifier directly from the transformed classifier’s text source, without requiring any additional training and operating in an unsupervised manner (i.e., we do not train a linear probe to associate the concept activations to class labels).

In summary, our contributions are as follows: **(i)** We propose a method to convert any frozen visual classifier into a CBM without relying on the CLIP constraint (CLIP-free), neither on annotated image-concept data (label-free), and for the first time, also deriving the concept-to-class classifier in a fully unsupervised manner. **(ii)** We demonstrate the effectiveness of our method with 40 different architectures, along with extensive ablation studies and intervention results, and further show how our method can also be used to perform zero-shot image captioning. **(iii)** We set new state-of-the-art results on CBMs, outperforming existing works including supervised CLIP-based CBMs, despite being trained only on ImageNet-1K.

## 2. Related Work

Concept Bottleneck Models (CBMs) are inherently interpretable models that predict by first mapping inputs to human-understandable concepts and then combining those concepts linearly to make the final decision, originally requiring concept annotations for each image [34]. Recently, Label-Free CBMs (LF-CBMs) have been introduced, which leverage CLIP to provide image-concept annotations. CLIP is used to generate ground-truth image–concept similarity scores, which serve as supervision for training the bottleneck layer of a downstream classifier. Several further works [5, 58] adopted this pipeline. Another work uses vision-language grounding models [46] to obtain image-concept annotations. Another line of work [18, 37, 42, 57] builds CBMs directly on top of CLIP, making them specifically tailored to CLIP-only models and thereby eliminating the need for a separate bottleneck training stage. In these methods, image features are queried against a predefined set of concepts in the CLIP embedding space, and the resulting cosine similarities are used directly as concept activations. As a result, all the above methods either (1) are limited to CLIP

models, or (2) depend on CLIP or vision–language models to obtain the image-concept annotations. On the other hand, we propose a fully CLIP-free CBM; our method does not rely on any external vision-language model to generate the image-concept annotations, nor on human annotation, and can be applied to any legacy model without the CLIP constraint. Finally, all CBMs works to date require training a linear probe to map concept activations to class predictions. In contrast, we demonstrate that our method can be readily used to derive the linear probe in a fully unsupervised manner.

**Transforming Visual Features to Text:** Text-to-Concept (T2C) [33] trains a linear layer to map image features of any classifier into the CLIP vision encoder space, such that they can be interpreted via text using the CLIP text encoder. This method 1) relies on the CLIP approach and/or its supervision, 2) it interprets the classifier in the CLIP space, which in practice amounts to interpreting the CLIP model itself rather than the classifier, and 3) it alters the classifier’s distribution by entirely discarding its output class distribution. In contrast, our method is CLIP-free, can be applied to any pretrained classifier without requiring any annotated data, it interprets the classifier in its own space and explicitly preserves the classifier’s distribution.

## 3. Method

We first elaborate on our proposed TextUnlock method in Section 3.1, which will then allow us to design our proposed U-F<sup>2</sup>-CBM, which we discuss in Section 3.2. U stands for unsupervised; L<sup>2</sup> for CLIP-free and Label-free (hence ‘double free’).

### 3.1. TextUnlock

A visual classifier assigns an image to a specific category from a predefined set of discrete class labels. For example, in ImageNet-1K [8], this set contains 1,000 class labels. Originally, these discrete class labels correspond to class names in text format. For example, in ImageNet-trained models, the discrete label 1 corresponds to the class *goldfish*. These classes are typically discretized to facilitate training with cross-entropy. However, when the textual class names are used, they provide an advantage. Specifically, when the textual class names are embedded into vector representations (e.g., using a word embedding model or text encoder), they provide semantic information. Specifically, these embeddings reside in a continuous space where nearby vectors capture related semantic associations. For the “goldfish” example, neighboring vectors might include terms such as “freshwater”, “fins” and “orange”. Such associations can be viewed as high-level conceptual attributes that characterize the class. Our method learns to map images into this text embedding space using *only* class names, thus linking both the class name *and its surrounding semantic associations*

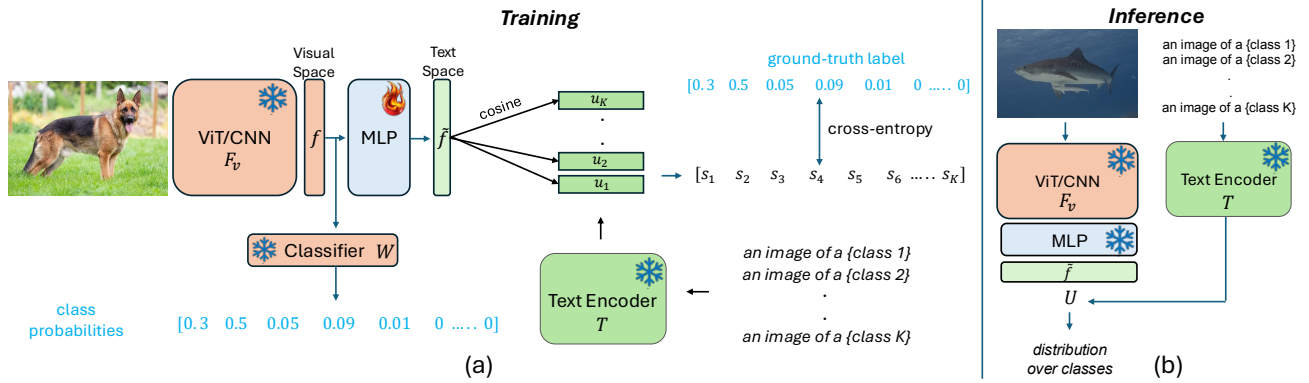


Figure 1. **Overview of our proposed TextUnlock.** (a) The process of training the MLP mapping between vision and text space. (b) The process of inference with the adapted visual classifier. The text encoder acts as weight generator for a linear classifier. \* indicates that the module is frozen, while  $\text{🔥}$  indicates trainable.

with the image. This process thus naturally supports unseen words that are not part of the class names (e.g., concepts in the CBM). We accomplish this through a trainable multi-layer perceptron (MLP) that projects the visual features into the text embedding space, and is explicitly trained to match its distribution with the original classifier’s class distribution. This is done while keeping both the visual and textual encoders frozen. By using solely the class names without any supplementary information, we can learn a semantically meaningful image-text space. This allows us to query the visual classifier with text queries beyond the class names (e.g., concepts) and obtain concept activations for CBMs.

Consider an image  $I$  and a visual classifier  $F$  composed of a visual feature extractor  $F_v$  and a linear classifier  $W$ . Note that  $F$  can be of any architecture.  $F_v$  embeds  $I$  into an  $n$ -dimensional feature vector  $f \in \mathbb{R}^n$ . That is,  $f = F_v(I)$ . The linear classifier  $W \in \mathbb{R}^{n \times K}$  takes  $f$  as input and outputs a probability distribution  $o$  for the image across  $K$  classes. That is,  $o = \text{softmax}(f.W) \in \mathbb{R}^K$ . For ImageNet-1K,  $K = 1000$ . Consider also any off-the-shelf text embedding model  $T$  which takes in an input text  $l$  and embeds it into a  $m$ -dimensional vector representation  $u \in \mathbb{R}^m$ . That is,  $u = T(l)$ . Note that  $u$  and  $f$  are not in the same space and can have a different number of dimensions, so we cannot query  $f$  with the text  $l$ .

We propose to learn a lightweight MLP mapping function that projects the visual features  $f$  into the text embedding space of  $T$ , resulting in a new vector  $\tilde{f}$ . That is,  $\tilde{f} = \text{MLP}(f)$ , where  $\tilde{f} \in \mathbb{R}^m$ . Note that the visual encoder  $F_v$ , the linear classifier  $W$ , and the text encoder  $T$  are all frozen; only the MLP is trainable, making our method *parameter-efficient*. We then take the textual class names of the  $K$  classes, and convert each into a text prompt  $l^p$ , represented as: “an image of a {class}” where {class} represents the class name in text format. This results in  $K$  textual prompts, each of which is encoded with  $T$ :  $u_i = T(l_i^p)$ ,

$\forall i = 1, \dots, K$ . Stacking all the encoded prompts, we get a matrix  $U \in \mathbb{R}^{K \times m}$ . Here,  $U$  acts as weights of the classification layer for our approach. We then calculate the cosine similarity<sup>1</sup> between each  $u_i$  and the visual features  $f$ :  $s_i = \tilde{f} \cdot u_i$ . Equivalently, this can be performed as a single vector-matrix multiplication:  $S = \tilde{f} \cdot U^T$ , where  $S \in \mathbb{R}^K$  represents the cosine similarity scores between the visual features and every text prompt  $l_i^p$  representing a class. In other words,  $S$  represents the classification logits of our approach. The most straightforward way to training the MLP is to leverage the ground-truth labels from the dataset, aligning  $S$  with the ground-truth distribution. However, this approach violates two key desiderata: (1) it necessitates annotated data, and (2) re-training the legacy classifier alters its original decision distribution  $o$ , thereby changing the reasoning process of the classifier (i.e., how it maps visual features to class probabilities and makes predictions). Notably, the original soft probability distribution  $o$  is a function of the linear classifier  $W$ , so  $W$  cannot be ignored. We instead propose to align  $S$  to the original decision distribution  $o$  through cross-entropy loss. For a single sample, the loss is

$$L = - \sum_{i=1}^K o_i \log \left( \frac{e^{s_i}}{\sum_{j=1}^K e^{s_j}} \right). \quad (1)$$

This task can be viewed as a knowledge distillation problem, except that we do not distill the knowledge of a bigger teacher model to a smaller student model, but instead **distill the distribution of the original model to its counterpart vision-language distribution**. This loss is equivalent to the KL divergence loss between  $o$  and the predicted distribution, since the additional entropy term  $H(o)$  that appears in the KL divergence loss is a constant that does not depend on the MLP parameters. Eq. 1 shows that our approach does not

<sup>1</sup> in the rest of this paper, we will omit the unit norm in cosine similarity to reduce clutter, and represent it with the dot product.

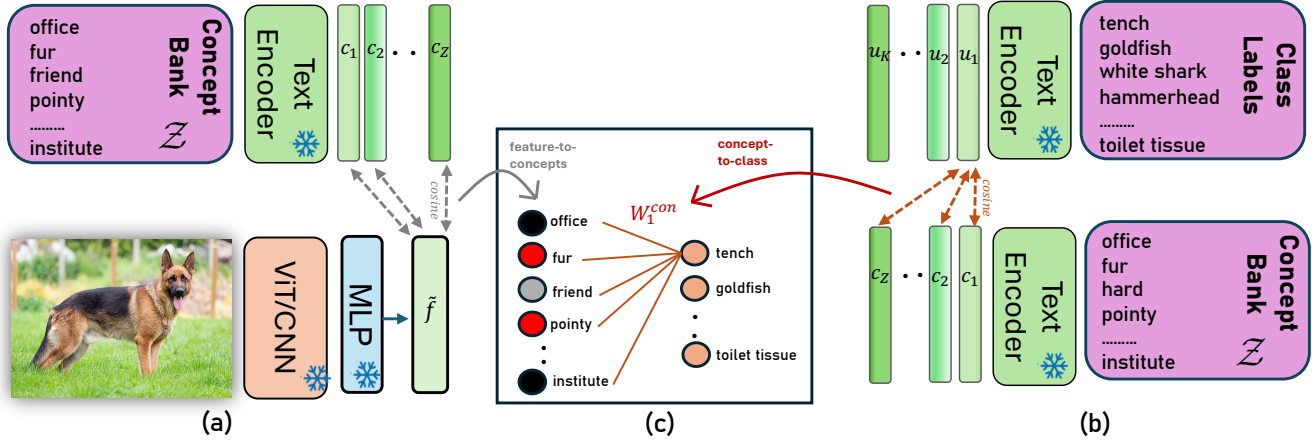


Figure 2. For any pretrained classifier, we first perform (a) concept discovery, followed by (b) building the concepts-to-class classifier in an unsupervised manner, which results in (c) our final U-F<sup>2</sup>-CBM. Note that the concept bank only needs to be encoded once.

require any labels for training, and steers the MLP to preserve the classifier’s distribution, since it is explicitly trained for that purpose. Note also that this loss function naturally encodes the classifier’s relationship across all classes.

We provide an illustration of the TextUnlock approach in Figure 1, and a PyTorch-like pseudocode of it in Listing 1. It is important to note that we only use the class name to formulate the textual prompt  $I^p$ , and no other supplementary information such as class descriptions, concepts or hierarchies (see Section 14 in the Appendix for more information).

```

# text_feats: textual features of class names from a frozen sentence encoder
, shape (num_classes, text_dim)
# classifier: linear classifier weights of a frozen vision_encoder, shape (
visual_dim, num_classes)
# mlp: trainable MLP from visual_dim -> text_dim
# images: batch of B images, shape (N, 3, height, width)

visual_feats = vision_encoder(images) # (N, visual_dim)
logits = visual_feats @ classifier # (N, num_classes)
original_dist = softmax(logits, dim=-1) # (N, num_classes)

mapped_feats = mlp(visual_feats) # (N, text_dim)
mapped_feats = l2_norm(mapped_feats) # (N, text_dim)
text_feats = l2_norm(text_feats) # (N, text_dim)

pred_logits = mapped_feats @ text_feats.T # (N, num_classes)
pred_dist = softmax(pred_logits, dim=-1) # (N, num_classes)

# cross entropy with original model's soft distribution
loss = -(original_dist * log(pred_dist)).sum(dim=1).mean()
loss.backward() # only mapper parameters are updated

```

Listing 1. PyTorch-like pseudocode for TextUnlock

After training, the projected visual features and the text encoder features lie in the same space. We can therefore query the visual features with any text by finding the alignment score between the embedded text and the projected visual features. In the case of image classification, the text queries remain the class prompts, and encoding them with the text encoder  $T$  is equivalent to generating the weights of a linear classifier for the classification task formulated as  $\text{argmax}(\tilde{f}.U^T)$ , see Figure 1(b).

### 3.2. U-F<sup>2</sup>-CBMs

Once the classifier’s distribution is matched to its corresponding vision–language counterpart distribution via TextUnlock, we proceed to formulate the proposed U-F<sup>2</sup>-CBMs. Note that at this stage, all model components (including the MLP) are frozen and no further training is performed. A CBMs consist of two steps: (1) **concept discovery**, followed by (2) **concept-to-class prediction**. In step (1), the dense output features of a visual encoder are first mapped to textual concepts (e.g., words or short descriptions of objects) each with a score that represents the concept activation to the image. In step (2), a linear classifier  $W^{con}$  is trained on top of these concept activations to predict the class.

**Concept Discovery:** We remind readers from Section 3 that  $U \in \mathbb{R}^{K \times m}$  is the output of the text encoder  $T$  for the class prompts, which represent the classification weights of the newly formulated classifier. In CBMs, we are given a large set of textual concepts, denoted as  $\mathcal{Z}$ , and with cardinality  $|\mathcal{Z}| = Z$ . Following other works [42] and without loss of generality, we use the  $Z = 20K$  most common words in English [11] as our concept set. These are general concepts that are sufficiently expressive and represent world knowledge and are not tailored towards any specific dataset. We ablate on other concept sets in Appendix Section 7 for the interested reader. To ensure that the concepts are meaningful, we apply a rigorous filtering procedure to the concept set. Specifically, we remove any terms that exactly match the target class name, as well as any constituent words that form the class name (for example, eliminating “tiger” and “shark” when the class name is “tiger shark”). In addition, we exclude 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. Details of this procedure can be found in Section 13 of the Appendix. This systematic filtering

Model	Top-1	Orig.	$\Delta$
ResNet50	75.80	76.13	-0.33
ResNet50 <sub>v2</sub>	80.14	80.34	-0.20
ResNet101	77.19	77.37	-0.18
WideResNet101 <sub>v2</sub>	82.21	82.34	-0.13
ResNeXt101-64x4d	83.13	83.25	-0.12
DenseNet161	77.04	77.14	-0.10
EfficientNetv2-M	84.95	85.11	-0.16
ConvNeXtV2-B	84.56	84.73	-0.17
ConvNeXtV2-B <sub>pt</sub> @384	87.34	87.50	-0.16
ViT-B/16	80.70	81.07	-0.37
ViT-B/16 <sub>v2</sub>	84.94	85.30	-0.36
ViT-L/32	76.72	76.97	-0.25
ViT-L/16	79.56	79.66	-0.10
Swin-Base	83.22	83.58	-0.36
Swinv2-Base	83.72	84.11	-0.39
BeiT-L/16	87.22	87.34	-0.12
DINOv2-B	84.40	84.22	+0.18

Table 1. **TextUnlock-ed visual classifiers for CNNs and Transformers maintain classification performance.** Comparison of our re-formulated classifiers for several models (remaining models in Appendix). Top-1 indicates our results of the new formulation, and Orig. denotes the original Top-1 accuracy.  $\Delta$  represents their difference ( $\Delta = \text{Top-1} - \text{Orig}$ ).

guarantees that the resulting concept set is free of terms that are overly similar or directly derived from the target classes. With this filtering procedure, since no concept appears in the training data, our concept discovery set becomes entirely zero-shot. This will also demonstrate the MLP’s ability to generalize to the semantic space surrounding the class name.

We use the same text encoder  $T$  that generates the linear classifier  $U$  to generate concept embeddings, by feeding each concept  $z_i \in \mathcal{Z}$  to the text encoder  $T$  to generate a concept embedding  $c_i$ . That is,  $c_i = T(z_i), \forall i = 1, \dots, Z$ . By performing this for all  $Z$  concepts, we obtain a concept embedding matrix  $C \in \mathbb{R}^{Z \times m}$ . For an image  $I$ , we extract its visual features  $f$  and use the MLP to map them to  $\tilde{f}$  which now lies in the text embedding space. That is,  $\tilde{f} = \text{MLP}(f)$ , and  $\tilde{f} \in \mathbb{R}^m$ . Since  $C$  and the mapped visual features  $\tilde{f}$  are now in the same space, we can query  $\tilde{f}$  to find which concepts it responds to. That is, we perform concept discovery using the cosine similarity between  $\tilde{f}$  and each row-vector in  $C$ . The concept activations are obtained by  $\tilde{f} \cdot C \in \mathbb{R}^Z$  and represent the activation score for each of the  $Z$  concepts. Concepts that the model identifies in an image will produce a high activation score. We provide an illustration in Figure 2(a).

**Concept-to-Class Prediction:** The classifier  $W^{con}$  takes the concept activations produced from the concept discov-

ery stage, and outputs a distribution  $S_{cn}$  over classes. We build  $W^{con}$  in an unsupervised manner; here *unsupervised* indicates that no training is required to map concept activations to classes. Recall that both  $U$  and  $C$  are outputs of the text encoder  $T$ , and they are already in the same space. Therefore, we can build the weights of the classifier  $W^{con}$  with a text-to-text search between the concepts and the class name. Specifically, we calculate the cosine similarity between the concept embeddings  $C$  and the classification matrix  $U$  to obtain the new weights for  $W^{con}$ . That is, we perform  $C \cdot U^T \in \mathbb{R}^{Z \times K}$ . Therefore, the weights of  $W^{con}$  represent how similar the class name is to each of the concepts. This process is shown in Figure 2(b). While the operation is performed entirely in the text feature space, the text encoder serves as the classifier weight generator. Because the MLP learns to map image representations to the text-encoder space, it is effectively mapping images into the classifier weight space. In total, the output distribution  $S_{cn}$  of the CBM is obtained by feeding the identified concept activations in the image obtained from the concept discovery stage, to  $W^{con}$ . That is,

$$S_{cn} = \underbrace{(\tilde{f} \cdot C^T)}_{\text{concept discovery}} \cdot \underbrace{(C \cdot U^T)}_{\text{concept-to-class}} = \tilde{f} \cdot \underbrace{C^T C}_{\text{gram matrix}} \cdot U^T. \quad (2)$$

From Eq. 2 we make an interesting observation. Our formulation involves scaling the linear feature-based classifier  $U$  by the gram matrix of concepts ( $C^T C \in \mathbb{R}^{m \times m}$ ). The gram matrix represents a feature correlation matrix measuring how different dimensions of the feature space relate to each other. Notably, if the gram matrix is the identity ( $C^T C = I$ ), we get back our original feature-based classifier given by  $\tilde{f} \cdot U^T$ . Therefore, to convert any classifier to a CBM, we plug in the gram matrix in-between, making it a convenient way to directly switch to an inherently interpretable model. Eq. 2 also shows that we do not change the linear classifier  $U$ , we only scale it by the gram matrix of concepts. This means our CBMs preserve the basic reasoning process of the original classifier. By this, we obtain CBMs that discover concepts and build  $W^{con}$  in an unsupervised manner for any classifier (Figure 2(c)). An additional unique property is the construction of CBMs *at inference time*, allowing any concept set to be selected to build a CBM on-the-fly. This makes our method highly flexible to the concept set.

## 4. Experiments

We first provide results of the the classifier with its corresponding vision-language distribution aligned using our method TextUnlock. We use the most challenging CBM benchmark of ImageNet-1K dataset. There is also widespread publicly available visual classifiers trained and evaluated on ImageNet. In Section 5 of the Appendix, we also report results on other datasets including Places365 [62],

Supervised CBMs					
Method	Model	Top-1	Method	Model	Top-1
LF-CBM	CLIP ResNet50	67.5	LF-CBM	CLIP ViT-B/16	75.4
LaBo	CLIP ResNet50	68.9	LaBo	CLIP ViT-B/16	78.9
CDM	CLIP ResNet50	72.2	CDM	CLIP ViT-B/16	79.3
DCLIP	CLIP ResNet50	59.6	DCLIP	CLIP ViT-B/16	68.0
DN-CBM	CLIP ResNet50	72.9	DN-CBM	CLIP ViT-B/16	79.5
DCBM-SAM2	CLIP ViT-L/14	77.9	DCBM-RCNN	CLIP ViT-L/14	77.8
Unsupervised, Label-Free, CLIP-Free CBMs (U-F <sup>2</sup> -CBM)					
Method	Model	Top-1	Method	Model	Top-1
U-F <sup>2</sup> -CBM	ResNet50	73.9	U-F <sup>2</sup> -CBM	ViT-B/32	73.3
U-F <sup>2</sup> -CBM	ResNet50 <sub>v2</sub>	<u>78.1</u>	U-F <sup>2</sup> -CBM	ViT-B/16	79.3
U-F <sup>2</sup> -CBM	ResNet101	75.3	U-F <sup>2</sup> -CBM	ViT-B/16 <sub>v2</sub>	83.2
U-F <sup>2</sup> -CBM	ResNet101 <sub>v2</sub>	79.9	U-F <sup>2</sup> -CBM	Swin-Base	82.2
U-F <sup>2</sup> -CBM	WideResNet50	76.9	U-F <sup>2</sup> -CBM	Swinv2-Base	82.6
U-F <sup>2</sup> -CBM	WideResNet50 <sub>v2</sub>	79.2	U-F <sup>2</sup> -CBM	ViT-B/16 <sub>pt</sub>	<u>81.5</u>
U-F <sup>2</sup> -CBM	WideResNet101 <sub>v2</sub>	81.0	U-F <sup>2</sup> -CBM	BeiT-B/16	83.0
U-F <sup>2</sup> -CBM	DenseNet121	69.9	U-F <sup>2</sup> -CBM	DINOv2-B	82.6
U-F <sup>2</sup> -CBM	DenseNet161	75.2	U-F <sup>2</sup> -CBM	ConvNeXt-B <sub>pt</sub>	84.0
U-F <sup>2</sup> -CBM	EfficientNetv2-S	83.0	U-F <sup>2</sup> -CBM	ConvNeXtV2-B <sub>pt</sub>	84.9
U-F <sup>2</sup> -CBM	EfficientNetv2-M	83.9	U-F <sup>2</sup> -CBM	BeiT-L/16	86.2
U-F <sup>2</sup> -CBM	ConvNeXt-Small	81.9	U-F <sup>2</sup> -CBM	ViT-L/16 <sub>v2</sub>	<u>86.3</u>
U-F <sup>2</sup> -CBM	ConvNeXt-Base	82.8	U-F <sup>2</sup> -CBM	ConvNeXtV2-B <sub>pt</sub> @384	<b>86.4</b>

Table 2. **Our U-F<sup>2</sup>-CBMs outperform CLIP-based counterparts.** Accuracy of Supervised and U-F<sup>2</sup>-CBMs (ours) on ImageNet validation set. Similar backbones are color-coded. Best within backbone family is underlined, overall best is **bold**.

EuroSAT [13] and DTD [6] showing that our method is also applicable to domain-specific, fine-grained datasets as well as datasets with a small number of classes. We apply TextUnlock on a diverse set of 40 visual classifiers. For CNNs, we consider the following family of models (each with several variants): Residual Networks (ResNets) [12], Wide ResNets [59], ResNeXts [56], ShuffleNetv2 [30], EfficientNetv2 [48], Densely Connected Networks (DenseNets) [16], ConvNeXts [27] and ConvNeXtv2 [54]. For Transformers, we consider the following family of models (each with several variants): Vision Transformers (ViTs) [9], DINOv2 [36], BeiT [4], the hybrid Convolution-Vision Transformer CvT [55], Swin Transformer [26] and Swin Transformer v2 [25]. All models are pretrained on ImageNet-1K from the PyTorch [31] and HuggingFace [53] libraries. Models with the subscript *pt* indicate that the model was pretrained on ImageNet-21k before being finetuned on ImageNet-1K. Models with a subscript *v2* are trained following the updated PyTorch training recipe [51]. BeiT, DINOv2 and ConvNeXtv2 are pretrained in a self-supervised manner before being finetuned on ImageNet-1k. When training with TextUnlock, both the pretrained classifier and text encoder remain frozen, only the MLP is trained on the ImageNet training set following Eq. 1.

Performance is evaluated using the same protocol and dataset splits as the original classifier, specifically the 50,000 validation split of ImageNet-1K. For the text encoder, we use the MiniLM Sentence Encoder [52] as it is fast and efficient. We provide ablation studies on other text encoders in Section 2 of the Appendix. Results for 17 classifiers are presented in Table 1, with the remaining 23 classifiers reported in Section 11 of the Appendix. We report the replicated Top-1 accuracy of the re-formulated classifier with our method in the first column, the original Top-1 accuracy of the classifier in the second column, and the difference between them ( $\Delta$ ) in the last column. As it can be seen, the loss in performance as indicated by  $\Delta$  is minimal, with an average drop in performance of approximately 0.2 points across all models. We also perform ablation studies on the MLP to verify its design, impact, role and that it learns meaningful transformations in Section 3 of the Appendix (see Section 3.1 for the role and impact of the MLP and Section 3.2 for its design hyperparameters). We also evaluate our transformed classifier robustness to prompt variations in Section 6 of the Appendix. For implementation details, we refer to Section 8 of the Appendix.

**U-F<sup>2</sup>-CBM:** We report CBM evaluation results on the Ima-

geNet validation set using the top-1 accuracy in Table 2. We compare against six SOTA methods: LF-CBMs [34], LaBo [57], CDM [37], DCLIP [32], DN-CBMs [42], and DCBM [18], all using the same concept set for fair comparison. All these methods are supervised and use CLIP-based models for computing the concept activations, and many use it as a backbone as well. Our U-F<sup>2</sup>-CBM outperforms all the supervised CBMs, setting a new state-of-the-art performance. Notably, even a simple ResNet-50 classifier trained solely on ImageNet already outperforms the CBM for the significantly more powerful ResNet-50 CLIP model trained on 400M samples (that is, we used 400× less images). Even an EfficientNetv2-S with only 21M parameters can significantly outperform all CLIP models. It outperforms the largest CLIP ViT-L/14 model of 428M parameters by +5.1% points, although being 20× smaller. The best results are obtained by the ConvNeXtv2 model, which achieves a top-1 accuracy of 86.4. All models show close to original accuracy, which means we can transform any classifier to be inherently interpretable without notable performance loss.

**Other Datasets:** We experiment with other datasets and show that our method is applicable even to domain-specific, fine-grained datasets as well as datasets with a small number of classes. We report CBM results on Places365 (domain-specific to scenes, 365 classes), DTD (domain-specific to texture and fine-grained, 47 classes), and EuroSAT (domain-specific to satellite images, 10 classes). When a baseline method is also reported on that dataset, we include it. Otherwise, we use CLIP models as a baseline to act as a feature extractor and to compute concept activations, and train a linear classifier on top of the concept activations, formulating a CLIP supervised baseline. All baselines use the same concept set. Note also that all baselines train a supervised linear classifier, while our method derives the classifier in an unsupervised manner. Results are shown in Table 3. For Places 365, we can see that U-F<sup>2</sup>-CBM for DenseNet161 classifier (using the transformed ImageNet-only trained classifier with TextUnlock) outperforms supervised CLIP-based ResNet and ViTs CBM methods. The same applies for EuroSAT and DTD with our U-F<sup>2</sup>-CBM. Therefore, the experiments show that our method scales to those scenarios as well.

**Concept Interventions:** Another common way to evaluate interpretability of CBMs is concept intervention. This follows metrics from the explainability literature [39]. In this way, we can show the effectiveness of the concepts, how we can mitigate biases, debug models and fix their reasoning by explicitly intervening in the concepts of the bottleneck layer to control predictions. As these metrics serve as complementary evaluations, we refer readers to Section 4 of the Appendix for the results.

In Figure 3, we present qualitative examples of a selection from the top concepts responsible for the prediction,

Dataset	Method	Model	Acc (%)
Places365	DCLIP	CLIP-ResNet50	37.90
	DCLIP	CLIP-ViT-B/16	40.30
	LF-CBM	CLIP-ResNet50	49.00
	LF-CBM	CLIP-ViT-B/16	50.60
	CDM	CLIP-ResNet50	52.70
	CDM	CLIP-ViT-B/16	52.60
	Ours	ResNet50	51.57
	Ours	DenseNet161	<b>53.42</b>
EuroSAT	Baseline	CLIP-ResNet50	86.27
	Baseline	CLIP-ViT-B/16	88.57
	Ours	ViT-B/16	93.65
	Ours	WideResNet101	94.12
	Ours	ResNet50	<b>94.22</b>
DTD	Baseline	CLIP-ResNet50	57.77
	Baseline	CLIP-ViT-B/16	61.86
	Ours	WideResNet101	66.97
	Ours	ViT-B/16	68.46
	Ours	ResNet50	<b>68.88</b>

Table 3. CBM results on Places365, EuroSAT, and DTD datasets.

along with their weight importance on the x-axis. The weight importance is calculated by multiplying the concept activation with its corresponding weight to the predicted class. We use various concept sets to demonstrate the flexibility of our method to any desired concept set directly at test time (on-the-fly), as this process simply involves encoding the chosen concept set using the text encoder. All examples use the LF-CBM concept set [34]. By observing the first example, the image is predicted as a “scorpion” because it has a lizard-like overall body, it is a venomous, desert animal with claws on its legs, and has large claws on its hand that look like a crab. Interestingly, in the second example, the top-detected concept is a “lift arm on the side.” Although this is not the primary feature defining a dumbbell, it reflects a well-documented bias in the literature [45] regarding the “dumbbell” class. Because most training images for this class show a dumbbell being lifted by an arm, the classifier not only learns to recognize the dumbbell but also associates it with the hand or arm that lifts it. With our method, we can obtain a textual interpretation of the biases that the original classifier learns. We also provide qualitative examples of global class-wise concepts detected in Section 15 of the appendix.

## 5. Zero-Shot Image Captioning

We also show that TextUnlock enables zero-shot image captioning with any pretrained visual classifier beyond CLIP models. Existing zero-shot captioning methods largely rely

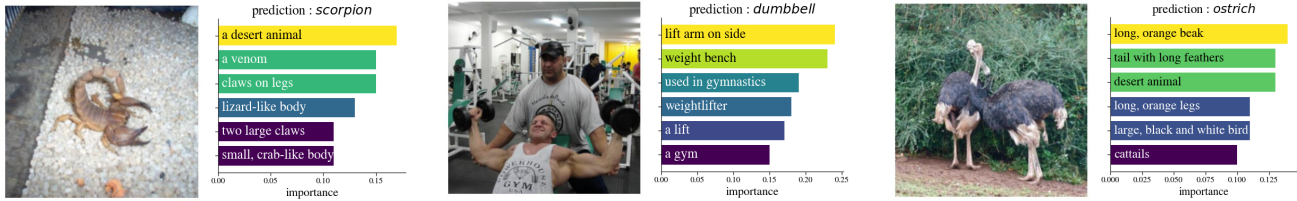


Figure 3. Qualitative examples of our U-F<sup>2</sup>-CBM. We show the top-detected concepts, each with their corresponding importance score.

on CLIP and its shared vision–language space. Thanks to TextUnlock, zero-shot image captioning can now be performed for any pretrained visual classifier. We adapt the method introduced in ZeroCap [49] for our purpose. Specifically, ZeroCap is a test-time approach that learns to produce a caption that maximizes the similarity with the image features. We first project the visual feature vector  $f$  using the MLP to obtain  $\tilde{f}$ . That is,  $\tilde{f} = \text{MLP}(f)$ . Since  $\tilde{f}$  is now in the same space as the text encoder  $T$ , we can measure its association to any encoded text. We utilize an off-the-shelf pretrained language decoder model, denoted as  $G$ , to generate open-ended text. We keep  $G$  frozen to maintain its language generation capabilities and instead use prefix-tuning [21] to guide  $G$  to generate a text that maximizes the similarity with the transformed visual feature vector  $\tilde{f}$ . More details about ZeroCap is in Section 12 of the appendix.

We now evaluate the performance of the zero-shot captions produced on the COCO image captioning dataset [24]. Since we do not train any model on the ground-truth image captions provided by COCO, we use zero-shot image captioning as a benchmark. Note that the COCO dataset differs in distribution than ImageNet, as a single image may contain many objects, interactions between them, and categories not included in ImageNet (*e.g.*, person). Therefore, it also serves as a way to evaluate generalization of our method to other datasets, given that we only used the ImageNet images and class names for training. We present results on the widely used “Karpathy test split” with various vision classifiers. As baselines, we compare our approach against existing methods in zero-shot image captioning, specifically ZeroCap [49] and ConZIC [60], both which use CLIP. For evaluation, we employ standard natural language generation metrics: BLEU-4 (B@4) [38], METEOR (M) [3], ROUGE-L (R-L) [22], CIDEr (C) [50], and SPICE (S) [1]. Results are shown in Table 4. ConvNeXtv2 achieves state-of-the-art performance on CIDEr and SPICE, the two most critical metrics for evaluating image captioning systems. Even with a simple ResNet-50 vision encoder trained on ImageNet-1K (1.2 million images), our approach outperforms the baseline methods on CIDEr and SPICE, despite the latter utilizing the significantly more powerful CLIP vision encoder, trained on 400 million image-text pairs (that is, we used 400× less images compared to CLIP). Qualitative examples of the produced

Model	B4	M	R-L	C	S
ZeroCap	<b>2.6</b>	<b>11.5</b>	—	14.6	5.5
ConZIC	1.3	11.5	—	12.8	5.2
<b>Ours</b>					
ResNet50	1.43	10.2	20.3	15.9	6.2
ResNet50 <sub>v2</sub>	1.47	10.5	20.6	16.8	6.5
WideResNet101 <sub>v2</sub>	1.50	10.4	20.5	16.6	6.4
ResNet101 <sub>v2</sub>	1.48	10.4	20.6	16.7	6.5
ConvNeXt-B <sub>pt</sub>	1.50	10.6	20.8	17.2	6.7
DINOv2-Base	1.50	10.7	21.0	17.3	6.7
ViT-B/16 <sub>v2</sub>	1.50	10.5	20.9	17.3	6.5
BeiT-L/16	1.50	10.6	20.9	17.6	6.9
ViT-B/16 <sub>pt</sub>	1.50	10.7	20.9	17.7	6.9
ConvNeXtV2-B <sub>pt</sub> @384	1.60	10.7	<b>21.1</b>	<b>17.9</b>	<b>6.9</b>
ConvNeXtV2-B <sub>pt</sub> @384 <sup>com</sup>	<b>4.40</b>	<b>12.7</b>	<b>30.2</b>	<b>18.7</b>	<b>7.2</b>

Table 4. Zero-Shot Image Captioning Performance

zero-shot image captions from different visual classifiers are shown in Section 10 of the Appendix. Note that our results in Table 4 are outperformed by the baseline ZeroCap on the BLEU-4 (B4) and METEOR (M) metrics. However, it is important to note that B4 and M are n-gram overlap-based metrics. They assume that the generated caption follows a specific structure and style. We verify this hypothesis by applying compositional image captioning [20, 29] with in-context learning. With this technique, the last row of Table 4 with the superscript *com* shows that the (B4, M and R-L) are boosted, which verifies our hypothesis about the low scores of B4 and M compared to baseline methods. We refer to Section 9 of the Appendix for more details.

## 6. Conclusion

We introduced a method for transforming any frozen visual classification model into a CBM. We proposed TextUnlock, the core of our method, that aligns the distribution of the original classifier with that of its vision–language counterpart. This allows us to then produce an unsupervised, label-free and CLIP-free (U-F<sup>2</sup>) CBM, which outperforms supervised CLIP-based CBMs across 40 models. We also showed zero-shot image captioning as an additional application. Finally, as with any research work, this study has its own limitations, discussed in Section 1 of the Appendix.

## Acknowledgment

Fawaz Sammani is funded by the Fonds Wetenschappelijk Onderzoek (FWO) (PhD) fellowship strategic basic research 1SH7W24N). N. Deligiannis acknowledges the ERC Consolidator Grant IONIAN (No. 101171240, DOI: 10.3030/101171240). Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them."

## References

- [1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016. 8
- [2] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016. 4
- [3] Satantjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, 2005. 8
- [4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 6
- [5] Itay Benou and Tammy Riklin-Raviv. Show and tell: Visually explainable deep neural nets via spatially-aware concept bottleneck models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30063–30072, 2025. 1, 2
- [6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2013. 6
- [7] Manh Dat. List of all english words (common), 2015. 4
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 6
- [10] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. <https://distill.pub/2021/multimodal-neurons>. 1
- [11] Google. google-10000-english, 2016. 4
- [12] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 6
- [13] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 6
- [14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016. 4
- [15] Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, 2019. 2
- [16] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016. 6
- [17] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015. 4
- [18] Patrick Knab, Katharina Prasse, Sascha Marton, Christian Bartelt, and Margret Keuper. Dcbm: Data-efficient visual concept bottleneck models. 2024. 2, 7, 3
- [19] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5338–5348, 2020. 1
- [20] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. Baby talk: Understanding and generating simple image descriptions. *CVPR 2011*, pages 1601–1608, 2011. 8, 4
- [21] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021. 8, 6
- [22] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. 8
- [23] Dekang Lin. Wordnet: An electronic lexical database. 1998. 6
- [24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 8
- [25] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11999–12009, 2021. 6
- [26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 6

- [27] Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022. 6
- [28] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 4
- [29] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7219–7228, 2018. 8, 4
- [30] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 6
- [31] TorchVision maintainers and contributors. Torchvision: Pytorch’s computer vision library. <https://github.com/pytorch/vision>, 2016. 6
- [32] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *International Conference on Learning Representations*, 2023. 7, 6
- [33] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *International Conference on Machine Learning*, 2023. 2
- [34] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 7, 5
- [35] OpenAI. Gpt-4o mini: advancing cost-efficient intelligence, 2024. 4
- [36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 6
- [37] Konstantinos P. Panousis, Dino Ienco, and Diego Marcos. Sparse linear concept discovery models. *2023 IEEE/CVF International Conference on Computer Vision Workshops (IC-CVW)*, pages 2759–2763, 2023. 2, 7
- [38] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002. 8
- [39] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. 7
- [40] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. *International Conference on Computer Vision (ICCV)*, 2023. 6
- [41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 1
- [42] Sukrut Rao, Sweta Mahajan, Moritz Bohle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision*, 2024. 2, 4, 7, 1, 3
- [43] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. 1, 4
- [44] Shiori Sagawa\*, Pang Wei Koh\*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. 1
- [45] Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. *ArXiv*, abs/1909.12072, 2019. 7
- [46] Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. VLG-CBM: Training concept bottleneck models with vision-language guidance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 2
- [47] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014. 4
- [48] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning (ICML)*, 2021. 6
- [49] Yoav Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zeropcap: Zero-shot image-to-text generation for visual-semantic arithmetic. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17897–17907, 2021. 8, 6
- [50] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2014. 8
- [51] Vasilis Vryniotis. How to train state-of-the-art models using torchvision’s latest primitives, 2021. 6
- [52] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. 6
- [53] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics. 6
- [54] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In-So Kweon, and Saining Xie. Convnext

- v2: Co-designing and scaling convnets with masked autoencoders. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16133–16142, 2023. [6](#)
- [55] Haiping Wu, Bin Xiao, Noel C. F. Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22–31, 2021. [6](#)
- [56] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2016. [6](#)
- [57] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19187–19197, 2022. [2](#), [7](#)
- [58] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. [1](#), [2](#)
- [59] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. [6](#)
- [60] Zequn Zeng, Hao Zhang, Zhengjue Wang, Ruiying Lu, Dongsheng Wang, and Bo Chen. Conzic: Controllable zero-shot image captioning by sampling-based polishing. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23465–23476, 2023. [8](#)
- [61] Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [1](#)
- [62] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. [5](#)