This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Show and Tell: Visually Explainable Deep Neural Nets via Spatially-Aware Concept Bottleneck Models

Itay Benou[1,2] Tammy Riklin Raviv[1,2]
[1]Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev
[2]Data Science Research Center, Ben-Gurion University of the Negev

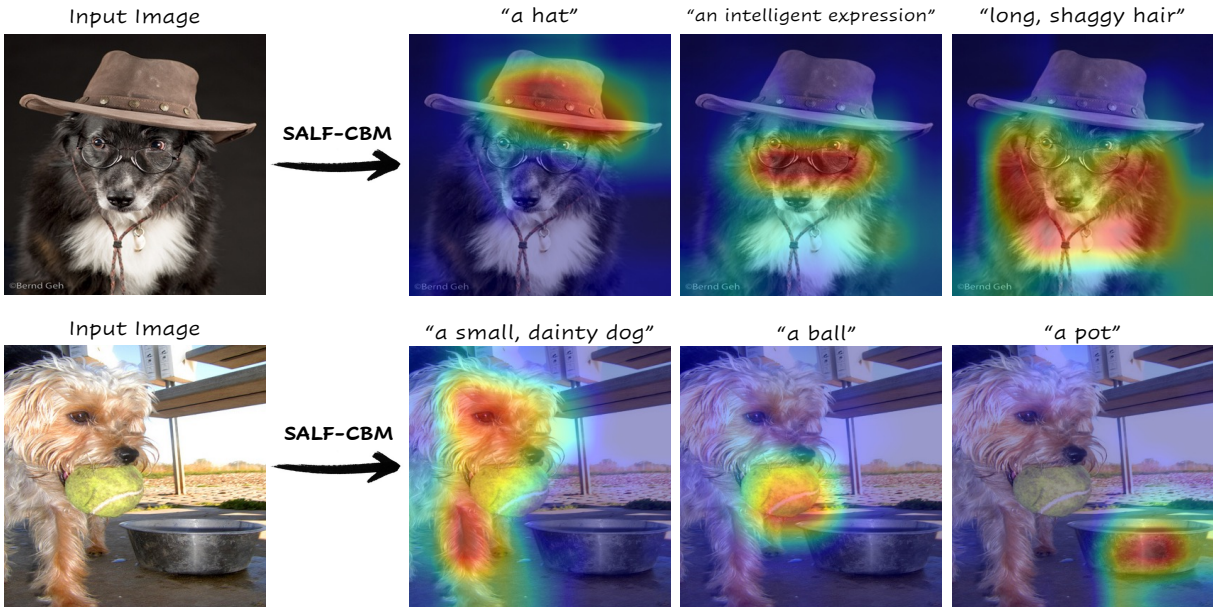


Figure 1. **Concept maps generated by our SALF-CBM**. Inspired by human visual interpretation, our method first decomposes input images into spatially-localized structures, associated with familiar concepts, independent of a specific task. Explainability of task-specific outputs is obtained by training a final task layer on-top of these maps.

Abstract

Modern deep neural networks have now reached human-level performance across a variety of tasks. However, unlike humans they lack the ability to explain their decisions by ***showing where*** *and* ***telling what*** *concepts guided them. In this work, we present a unified framework for transforming any vision neural network into a spatially and conceptually interpretable model. We introduce a spatially-aware concept bottleneck layer that projects "black-box" features of pre-trained backbone models into interpretable concept maps, without requiring human labels. By training a classification layer over this bottleneck, we obtain a self-explaining model that articulates which concepts most influenced its prediction, along with heatmaps that ground them in the input image. Accordingly, we name this method "Spatially-Aware and Label-Free Concept Bottleneck Model" (SALF-CBM). Our results show that the proposed SALF-CBM:* ***(1)*** *Outperforms non-spatial CBM methods, as well as the original backbone, on a variety of classification tasks;* ***(2)*** *Produces high-quality spatial explanations, outperforming widely used heatmap-based methods on a zero-shot segmentation task;* ***(3)*** *Facilitates model exploration and debugging, enabling users to query specific image regions and refine the model's decisions by locally editing its concept maps.*

1. Introduction

Humans often rationalize visually-based assessments or conclusions by describing *what* they have seen and *where* they have seen it, using both semantic concepts and their

spatial locations. For example, an image of a *dog* wearing *glasses* and a *hat*, as shown at the top of Figure 1, is likely to be interpreted as playful or funny due to the unexpected spatial composition of concepts. Notably, this mechanism operates independently of a specific task; even when looking for a dog in the bottom image of Figure 1, one may notice the tennis ball in its mouth and the pot next to it.

Similarly, the ability to explain AI models using spatially-aware concepts is crucial for elucidating their decision-making processes. Such an approach enables the introduction of quality control mechanisms, i.e., understanding the underlying causes of a model's behavior and adjusting it when necessary. These capabilities are essential for ensuring the safe and transparent integration of deep neural networks into critical fields like medical imaging and autonomous driving, as required by the AI Act recently passed by the European Parliament [1].

Most current explainable AI (XAI) methods, however, provide either spatial or concept-based explanations. Spatial approaches, generally referred to as attribution methods, produce heatmaps that highlight the image regions most contributing to the model's output. These heatmaps are generated either by propagating gradients through the model with respect to its input [5, 7, 25, 26, 28–31], or by using attribution-propagation methods [2, 4, 6, 9, 20, 32, 37] that distribute "relevance" (i.e., the contribution of a neuron to the output) backwards through the network, layer by layer. While these methods can visualize the model's spatial attention, in the absence of semantic descriptions, their output can be ambiguous [10, 16].

Concept Bottleneck Models (CBMs) [17, 18, 22, 33, 36, 40], on the other hand, are an increasingly popular method for obtaining concept-based explanations. Unlike attribution methods, CBMs provide *ante-hoc* explanations—i.e., their explainability mechanism is embedded into the model itself. Current CBMs work by introducing a non-spatial bottleneck layer that maps model features to an interpretable concept space, followed by training a final output layer over these concepts. This design ensures that CBMs are highly interpretable, as their predictions are directly based on the concepts used to explain them. However, existing CBMs provide global concept-based explanations without localizing them in the image. Moreover, the interpretable bottleneck layer often comes at the expense of the final task accuracy, which limits their applicability.

In this work, we present a *spatially-aware* CBM that combines concept-based explanations with the ability to visually ground them in the input image. In contrast to traditional CBMs, we preserve the spatial information of features and project them into a spatial concept space. This is achieved in a label-free manner by leveraging the capability of CLIP [23] to produce local image embeddings using visual prompts [27]. Accordingly, we name our method *"spatially-aware and label-free CBM"* (SALF-CBM). The main contributions of our work are as follows: **(1) Novel unified framework:** we present the first label-free CBM that provides both concept-based (global) and heatmap-based (local) explanations. **(2) Classification results:** SALF-CBM outperforms non-spatial CBMs on several classification tasks, and can even achieve better classification results than the original (non-CBM) backbone model. **(3) Heatmap quality:** our method produces high-quality heatmaps that can be used for zero-shot segmentation. We demonstrate their advantage over widely-used heatmap-based methods in both qualitative and quantitative evaluations. **(4) Explain anything:** SALF-CBM facilitates interactive model exploration and debugging, enabling users to inquire about concepts identified in specific image regions, and to adjust the model's final prediction by locally refining its concept maps. **(5) Applicability:** Our method is model-agnostic and can be applied to both CNNs and transformer architectures, while not introducing any additional learnable parameters compared to non-spatial CBMs.

2. Related Work

XAI methods for computer vision can be categorized by two axes: local (heatmap-based) vs. global (concept-based) approaches, and *post-hoc* vs. *ante-hoc* explanations. In this section, we overview existing methods along these lines.

Heatmap-based explainability. This refers to a family of *post-hoc* explainability techniques, often called attribution methods, that visualize the parts of the input image that contribute most to the model's output. *Gradient-based methods* generate explainable heatmaps by backpropagating gradients with respect to the input of each layer. Some of these methods, such as FullGrad [30], are class-agnostic as they produce roughly identical results regardless of the output class [29, 31], while others, such as GradCAM [25], generate class-dependent heatmaps [8, 28]. This property is essential when the true class is ambiguous. While widely used, their main drawback is high sensitivity to gradient noise, which may render their outcomes impractical [3]. To address this issue, some Class Activation Maps (CAM) methods [39], such as ScoreCAM [34], produce gradient-free explanation maps.

Attribution propagation methods decompose the output of a model into the contributions of its layers by propagating "relevance" in a recursive manner, without exclusively relying on gradients. Common attribution propagation methods, such as Layer-wise Relevance Propagation (LRP) [6], are primarily applicable to Convolutional Neural Networks (CNNs) [20, 26, 37]. Later approaches have been adapted to accommodate vision transformers (ViTs) [11], exploiting their built-in self-attention mechanism [2, 9, 23, 32]. We note that, unlike our SALF-CBM, both gradient-based and attribution propagation methods do not provide concept-

based explanations. Additionally, since these are *post-hoc* techniques, they do not enable test-time user intervention.

Concept-based explainability. An alternative way of explaining vision models is by using human-interpretable concepts. Various methods provide such explanations in a *post-hoc* manner. For example, Testing Concept Activation Vectors (TCAV) [15] measures the importance of user-defined concepts to the model's prediction by training a linear classifier to distinguish between concepts in their activation space. However, this requires labeling images with their corresponding concepts in advance. ACE [13] extends this idea by applying multi-resolution segmentation to images from the same class, followed by clustering similar segments into concepts to compute their TCAV scores. Similarly, Invertible Concept Embeddings (ICE) [38] and Concept Recursive Activation Factorization (CRAFT) [12] provide concept-based explanations using matrix factorization of feature maps. CRAFT also generates attribution maps that localize concepts in the input image. These methods, however, are mostly applicable to CNN architectures [13], which use non-negative activations [12, 38], and therefore cannot be directly applied to other types of models. Additionally, since they provide *post-hoc* explanations, they do not enable test-time user intervention.

In contrast, *Concept-Bottleneck Models (CBMs)* is a family of *ante-hoc* interpretable models whose explainability mechanism is an integral part of the model itself. CBMs operate by introducing a concept-bottleneck layer into pre-trained models, before the final prediction layer. The goal of this bottleneck is to project features into an interpretable concept space, where each neuron corresponds to a single concept. Unlike *post-hoc* methods, the output of CBMs is directly based on interpretable concepts, making them easily explainable and allowing user intervention by modifying concept neurons activations. In the original CBM work [17], the concept bottleneck layer was trained using manual concept annotations, limiting its ability to scale to large datasets. Recently, Post-Hoc CBM (P-CBM) [36] and Label-Free CBM (LF-CBM) [22] addressed this issue by leveraging CLIP to assign concept scores for training images, thus not requiring concept annotations. LF-CBM also presented an automatic process for creating a list of task-relevant concepts using GPT-3. While showing good interpretability results, both P-CBM and LF-CBM present a performance drop on the final classification task compared to the original (non-CBM) model. Additionally, unlike our SALF-CBM, these methods are limited to global concept explanations, and are unable to localize these concepts within the image.

3. Method

Given a pre-trained backbone model, we transform it into an explainable SALF-CBM as illustrated in Figure 2a:

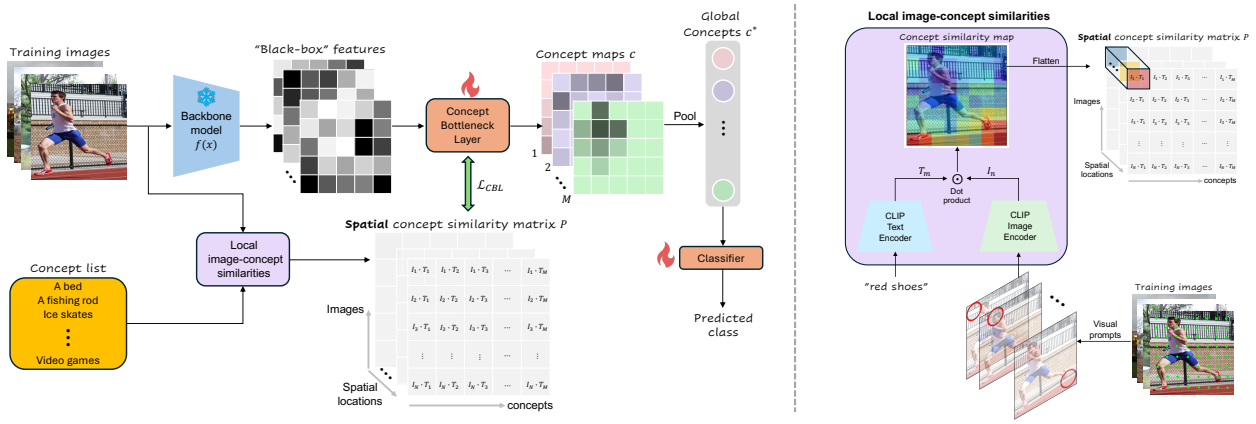
Step 1: Automatically generate a list of task-relevant visual concepts; **Step 2:** Using CLIP, compute a spatial concept similarity matrix that quantifies what concepts appear at different locations in the training images; **Step 3:** Train a spatially-aware Concept Bottleneck Layer (CBL) that projects the backbone's "black-box" features into interpretable concept maps. **Step 4:** Train a sparse linear layer over the pooled concept maps to obtain the model's final prediction. We describe each step in the following sections.

3.1. Concept list generation

Let $\mathcal{X}$ denote an image classification dataset with N training images $\{x_1, \ldots, x_N\}$ and L possible classes. We aim to generate a list of visual concepts $\mathcal{T}$ that is most relevant to the target classes, without relying on human experts. For this purpose, we follow the automatic procedure described in [22]. First, an initial concept list is obtained by prompting GPT as follows: "List the most important features for recognizing something as a {class}"; "List the things most commonly seen around a {class}"; and "Give superclasses for the word {class}", for each class in the dataset. Concepts that are too long, too similar to one of the classes or to another concept, or do not appear in the training data - are then discarded. The resulting filtered list of M concepts is denoted by $\mathcal{T} = \{t_1, \ldots, t_M\}$. See [22] for full details.

3.2. Local image-concept similarities

Vision-language models such as CLIP have been widely used for obtaining *global* image descriptions, as in non-spatial CBMs [22, 36]. Here, we aim to extend this approach to *locally* describe different image regions using visual concepts. Inspired by [27], we leverage CLIP's visual prompting property by drawing a red circle around specific image regions, which enables CLIP to focus on these areas while preserving global context. We apply this property to our training set as illustrated in Figure 2b. Formally, let $x_n \in \mathcal{X}$ denote a training image with spatial dimensions $H \times W$. We create a uniform grid of $\tilde{H} \times \tilde{W}$ locations in the image with integer strides d_H and d_W, i.e., $d_H = \lfloor \frac{H}{\tilde{H}-1} \rfloor$ and $d_W = \lfloor \frac{W}{\tilde{W}-1} \rfloor$. For each x_n, a set of $\tilde{H} \cdot \tilde{W}$ augmented images is then obtained by drawing a red circle with radius r around each location in the grid. We denote by $x_n^{(h,w)}$ the image x_n with a red circle located at (h, w). Next, we compute a *local similarity score* between a visual concept $t_m \in \mathcal{T}$ and the image at location (h, w) as follows: $P[n, m, h, w] = \frac{I_n^{(h,w)} \cdot T_m}{\|I_n^{(h,w)}\| \|T_m\|}$, where $I_n^{(h,w)} = E_I(x_n^{(h,w)})$ and $T_m = E_T(t_m)$ denote the CLIP embeddings of the augmented image and the concept, respectively. As demonstrated in [27], this score measures how well the concept t_m describes the image region highlighted by the red circle. By computing this score for every spatial location (h, w) in the grid, we obtain a concept sim-



(a) Training process scheme.

(b) Local image-concept similarities block.

Figure 2. **SALF-CBM training: (a)** Given a pre-trained backbone model, we: (i) Generate task-relevant concepts; (ii) Describe training images using local image-concept similarities; (iii) Train a spatially-aware concept bottleneck to project features into interpretable concept maps; (iv) Train a sparse classification layer over these maps. **(b)** Local image-concept similarities computation using visual prompting.

ilarity map for the entire image x_n, as shown in Figure 2b. A complete *spatial concept similarity matrix* P is then constructed by calculating these local similarity scores for all concepts $t_m \in \mathcal{T}$ and all images $x_n \in \mathcal{X}$ in the training set. This matrix is computed once, prior to training, and is later used for learning the spatially-aware concept bottleneck layer. We note that the grid resolution $\tilde{H} \times \tilde{W}$ and the circle radius r are hyper-parameters, where $\tilde{H}$ and $\tilde{W}$ control the coarseness of the concept similarity map, and r defines the receptive field around each location. We optimize these hyper-parameters per-dataset (see supplementary).

3.3. Training the concept-bottleneck layer

We aim to learn a bottleneck layer g that linearly projects "black-box" feature maps $f(x)$ of a pre-trained backbone model into interpretable concept maps. Rather than spatially pooling the backbone's features as in conventional CBMs, we retain their spatial information, and resize them to fit the grid's dimensions ($\tilde{H} \times \tilde{W}$) using a bilinear interpolation. We then use a single 1×1 convolution layer with M output channels to produce the desired concept maps, i.e., $c(x) = g(f(x)) \in \mathbb{R}^{M \times \tilde{H} \times \tilde{W}}$. We denote the full list of concept maps for all training images $x_n \in \mathcal{X}$ by $C[n, m, h, w] = [c(x_1), \ldots, c(x_N)]$. In order to obtain concept maps that match the image-concept similarities in P, we train our bottleneck layer using an extended version of the cubic cosine similarity loss from [22] as follows:

$$\mathcal{L}_{CBL} = -\sum_{m=1}^{M}\sum_{h,w} sim\left(q[m,h,w], p[m,h,w]\right) \qquad (1)$$

where $q[m,h,w]$ denotes $C[:,m,h,w]$, $p[m,h,w]$ denotes $P[:,m,h,w]$ and $sim(\cdot,\cdot)$ denotes the cubic cosine similarity function $sim\left(q,p\right) = \frac{\bar{q}\cdot\bar{p}}{\|\bar{q}\|\|\bar{p}\|}$. Here, $\bar{q}$ and $\bar{p}$ are normalized to have zero-mean and raised elementwise to the power of three to emphasize strong concept-image matches. We note that our spatial bottleneck layer requires the same number of parameters as the fully-connected bottleneck typically used in non-spatial CBMs [17, 22, 36]. Furthermore, our bottleneck layer accommodates both CNN and vision transformer architectures: For CNN backbones, feature maps are used directly as inputs to the bottleneck, while for ViTs, the patch tokens are reshaped back into their original spatial arrangement. See details in the supplementary.

3.4. Training the final classification layer

Once the concept bottleneck layer is trained, we spatially pool its output concept maps $c(x)$ to obtain *global* concept activations $c^*(x)$, each corresponds to a single visual concept t_m. We aim to explain each output class of our model using a small set of interpretable concepts. We therefore train a sparse linear layer on top of $c^*(x)$ to obtain the final classification scores $z = Wc^* + b$ and the predicted class $\hat{y} = \arg\max(z)$. Here, W and b denote the classification weights and bias term, respectively. This layer is trained in a fully-supervised manner with the following loss function, using the GLM-SAGA optimizer [35]:

$$\sum_{n=1}^{N} \mathcal{L}_{ce}\left(Wc^*(x_n) + b, y_n\right) + \lambda\mathcal{R}(W) \qquad (2)$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss, y_n is the class label of training image x_n, λ is the regularization strength and $\mathcal{R}(W) = (1-\alpha)\frac{1}{2}\|W\|_F + \alpha\|W\|_{1,1}$ is the elastic net regularization term, where $\|W\|_F$ is the Forbenius norm and $\|W\|_{1,1}$ is the elementwise matrix norm.

3.5. Test-time explainability

Once trained, SALF-CBM provides explanations at multiple levels of granularity as an integral part of its forward pass, without any external tools or additional computations.

Global decision rules. As described in Section 3.4, SALF-CBM's final prediction is a linear combination of a sparse set of concept activations. Therefore, one can gain an intuitive understanding of the model's decision rules simply by examining which concepts $t_m \in \mathcal{T}$ are connected to a specific class $l \in \{1, \ldots, L\}$ by non-zero weights. For instance, in Figure 3, we show Sankey diagrams visualizing the class weights of SALF-CBM trained on ImageNet, for two different classes: "crate" and "toy store".

Concept-based explanations ("tell what"). We aim to explain individual model decisions by evaluating the contribution of a visual concept $t_m \in \mathcal{T}$ to the the model's output $\hat{y}$ on a given test image x. This is achieved by computing a contribution score $S(x, m, \hat{y} = l) = W[m, l]c^*(x)[m]$, where m is the concept index, $l \in \{1, \ldots, L\}$ is the index of the predicted class and $c^*(x)[m]$ is the global concept activation, normalized by its mean and standard deviation on the training data. Since $W[:, l]$ is sparse, the majority of contribution scores are zero, so the model's prediction can be explained by a small set of k concepts whose absolute contribution scores are the highest. In Figure 3, we illustrate the top-3 concepts with the highest contribution scores for the "toy store" and "crate" classes.

Spatial explanations ("show where"). In addition to global concept-based explanations, our method produces heatmaps which highlight the location of each concept in the input image. Specifically, given the top-k contributing concepts, we upsample their associated concept maps $c(x)[m]$ to the input image dimensions, using a bilinear interpolation. Examples of heatmaps associated with the most contributing concepts for two different output classes are presented in Figure 3. We note that the resolution of the heatmaps can be controlled by adjusting the density of the visual prompting grid, as discussed in Section 3.2.

3.6. Model exploration and debugging

We introduce two interactive features that enable users to explore their model's decision-making process and debug failure cases in an intuitive manner.

Explain Anything. Inspired by the Segment Anything Model (SAM) [19], this feature allows users to actively "prompt" SALF-CBM with inputs such as points, bounding boxes, or free-form masks, to explore what visual concepts were recognized in the specified region-of-interest (ROI). Specifically, given an image x, the computed concept maps $c(x)$ (upsampled to the image dimensions) and a user-provided ROI in the form of a binary mask I, we compute the aggregated activation of each concept within the ROI: $a(x, m \mid I) = \sum I \odot c(x)[m]$, where $\odot$ represents elementwise multiplication. By presenting the top-k concepts with the strongest aggregated activation, we provide a concise overview of the model's perception of the ROI. We note that in addition to user-provided ROIs, our method supports segmentation masks from tools such as SAM to automatically produce objects descriptions.

Local user intervention. We enable users to intervene in the model's final prediction by suggesting counterfactual explanations in specific image regions, i.e., *"how would the model's prediction change if concept $\mathcal{A}$ were more/less activated at location $\mathcal{B}$?"*. Given an image x, the concept map $c(x)[m]$ of a specific concept t_m and the predicted class $\hat{y}$, one can locally edit the concept map according to their judgment and understanding of the task, as follows: $c(x)[m] \leftarrow c(x)[m] + \beta I$, where I is a binary mask of the edited region and β is a correction factor that can be either positive or negative. By tuning concept activations up or down in specific regions and re-running the final classification layer, one can observe how the model adjusts its prediction $\hat{y}$ based on the revised concept maps.

4. Experiments

We thoroughly evaluate the different components of our method. In section 4.1, we test its classification accuracy compared to several baselines, on different large-scale datasets. In section 4.2, we present qualitative and quantitative evaluations of our SALF-CBM's heatmaps in comparison to several other heatmap-based methods. In section 4.3, we validate the concepts learned by SALF-CBM's bottleneck layer by conducting a user study. In section 4.4, we demonstrate how the proposed Explain Anything and user intervention features are used to debug model errors. Additional results are provided in the supplementary.

4.1. Classification accuracy

Experimental setup. We test our method on a diverse set of classification datasets: CUB-200 (fine-grained bird-species classification), Places365 (scene recognition) and ImageNet. We train a SALF-CBM on each of the three datasets, using a appropriate backbone model to allow fair comparisons with competing CBM methods [22, 36]: For CUB-200 we use a ResNet-18 pre-trained on CUB-200, and for both ImageNet and Places365 we use a ResNet-50 pre-trained on ImageNet. For each dataset, we use the same initial concept list and regularization parameters α and λ as in [22], resulting in 370 concepts for CUB-200, 2544 concepts for Places365 and 4741 concepts for ImageNet. For computing local image-concept similarities, we use CLIP ViT-B/16 and a visual prompting grid of 7×7 with $r = 32$ for all experiments. Results with different grid parameters and with a ViT backbone are provided in the supplementary.

Results. Table 1 presents the classification accuracy of our SLAF-CBM compared to several other methods: (1) the

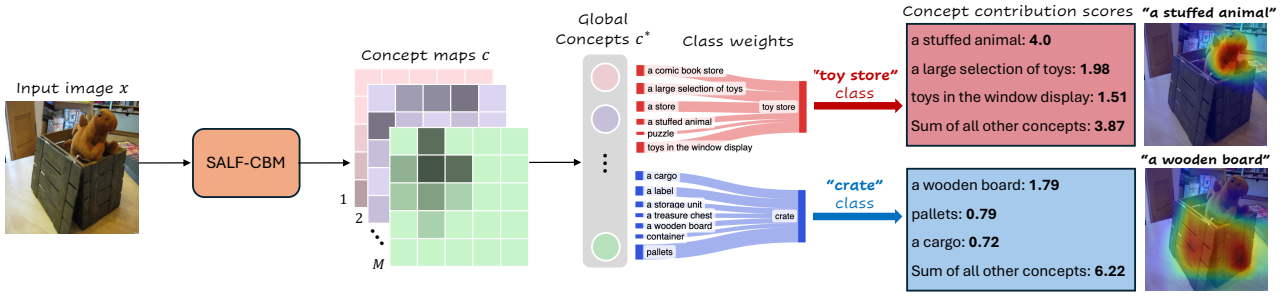


Figure 3. **Test-time explainability.** Global decision rules can be inferred by visualizing the sparse class weights. Individual model decisions are explained *globally* and *locally* by the concept contribution scores and their associated spatial heatmaps.

standard pre-trained backbone model with its original classification layer; (2) the standard backbone model with a sparse classification layer (reported from [22]); (3) post-hoc CBM (P-CBM) [36]; and (4) Label-Free CBM (LF-CBM) [22]. We note that in P-CBM [36], they do not report their results on ImageNet and Places365, and it is unclear how to scale it to those datasets. For fair comparisons, results with sparse and non-sparse classification layers are shown separately. We see that when using a sparse final layer, our SALF-CBM outperforms both P-CBM and LF-CBM on all three datasets. Notably, **our method is the best performing sparse method on the the two larger-scale datasets (Places365 and ImageNet)**, outperforming even the original backbone with a sparse final layer. To demonstrate the high-limit potential of our method, we assess its performance with a non-sparse final layer. Remarkably, the non-sparse SALF-CBM achieves better classification results than original (non-sparse) model on both ImageNet and Places365, even though its predictions are based on interpretable concepts.

These results indicate that SALF-CBM facilitates model interpretability without compromising performance; in fact, it can outperform the original backbone model when using a comparable final layer (i.e., sparse or non-sparse). We also note that the performance gap between the sparse and non-sparse SALF-CBMs is relatively small (less that 1% on ImageNet), indicating that our model effectively captures the full span of possible explanations using a sparse set of concepts.

Model	Sparse	Dataset		
		CUB-200	Places365	ImageNet
Standard	Yes	**75.96%**	38.46%	<u>74.35%</u>
P-CBM [36]	Yes	59.60%	N/A	N/A
LF-CBM [22]	Yes	74.31%	<u>43.68%</u>	71.95%
SALF-CBM	Yes	<u>74.35%</u>	**46.73%**	**75.32%**
Standard	No	**76.70%**	48.56%	76.13%
SALF-CBM	No	76.21%	**49.38%**	**76.26%**

Table 1. **Classification accuracy.** Our method outperforms P-CBM and LF-CBM on all three datasets, and is the highest performing model on ImageNet and Places365. Results are shown separately for sparse and non-sparse final layers. Best results are in bold and 2nd-best are underlined.

4.2. Beyond classification: zero-shot segmentation

Experimental setup. We conduct a quantitative analysis of the heatmaps generated by our method in a zero-shot segmentation task. We follow a standard protocol for evaluating heatmap-based explainability methods [9] on ImageNet-segmentation dataset [14], a subset of the ImageNet validation set containing 4,276 images with ground-truth segmentation masks of the class object. In order for our concept maps to correspond to ImageNet classes, we train a SLAF-CBM with a ResNet-50 backbone on ImageNet, using a concept list of the form "An image of a {class}", where {class} refers to each of the ImageNet classes. According to [9], the resulting heatmaps are binarized to obtain a foreground/background segmentation, and evaluated with respect to the ground-truth masks based on three metrics: mean average precision (mAP) score, mean intersection-over-union (mIOU) and pixel accuracy.

Results. Table 2 presents the zero-shot segmentation results of our method, compared to several widely-used explainability methods: LRP [6], integrated gradients (IG) [31], GradCAM [25], GradCAM++ [8], ScoreCAM [34], and FullGrad [30]. **Notably, our SALF-CBM achieves the best pixel accuracy and mIOU segmentation scores, and the second best mAP.** Specifically, our method demonstrates significant improvements in pixel accuracy and mIOU (+3.9% and +2.52% over the 2nd-best method, respectively), indicating that our heatmaps are consistently better aligned with the ground-truth masks. In Figure 4, we present a qualitative comparison to the baseline methods, for different images from the ImageNet validation set. We observe that LRP [6] and IG [31] typically produce noisy results, and struggle to accurately localize the class object. GradCAM [25], GradCAM++ [8], ScoreCAM [34]

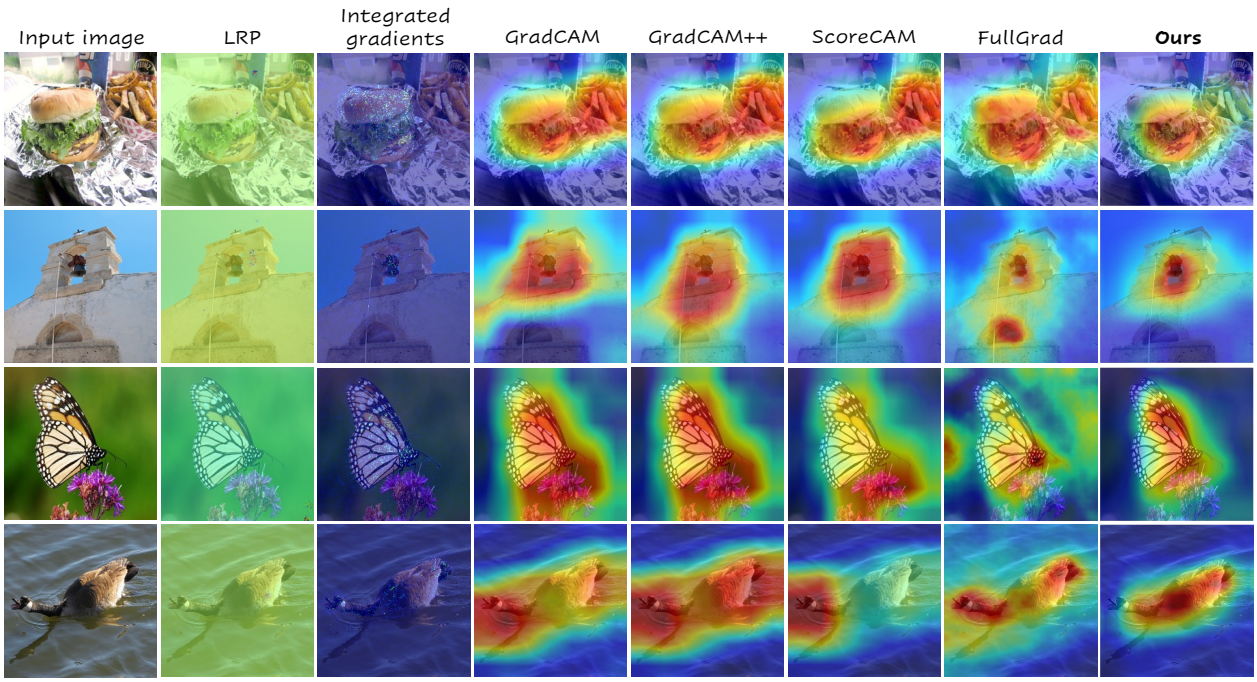


Figure 4. **Qualitative heatmaps comparison.** Explanation map of each method with respect to the ground-truth class (from top to bottom): "Cheeseburger", "Bell-cote", "Monarch butterfly" and "Goose".

and FullGrad [30] manage to highlight the target region, but also include unrelated background areas. Conversely, our method generates heatmaps that accurately captures the class object, thus providing more precise explanations.

Method	Pixel Acc. ↑	mIoU ↑	mAP ↑
LRP [6]	69.52%	36.85%	69.95%
IG [31]	68.49%	46.59%	73.46%
GradCAM [25]	71.34%	53.34%	83.88%
GradCAM++ [8]	71.31%	53.56%	83.93%
ScoreCAM [34]	69.56%	51.44%	81.78%
FullGrad [30]	<u>73.04%</u>	<u>55.78%</u>	**88.35%**
SALF-CBM	**76.94%**	**58.30%**	<u>85.31%</u>

Table 2. **Zero-shot segmentation results.** Our SALF-CBM achieves the highest mIoU and pixel accuracy, and the second highest mAP. Best results are in bold, 2nd-best are underlined.

4.3. Bottleneck interpretability validation

We conduct a user study to quantitatively validate that SALF-CBM's bottleneck neurons in-fact correspond to their designated target concepts.

Experimental setup. Following [24], we evaluate global concept neurons c^* from SLAF-CBM's bottleneck layer compared to output neurons of the baseline ResNet-50 backbone. We assign concept labels to baseline neurons using CLIP-Dissect [21] and rank neurons from both models by interpretability scores using CLIP-Dissect's soft-WPMI metric. We then sample 10 neurons from the top 30% interpretable neurons and 10 from the remaining 70% for each model. For each neuron, we show 25 users the five most activated test images and ask them to rate from 1-5: (a) *semantic consistency*: "Do these images share a common semantic concept?" and (b) *concept accuracy*: "Does [neuron label] describe a common concept shared by these images?".

Results. Figure 5 shows that SALF-CBM achieves significantly better scores in both semantic consistency and concept accuracy, demonstrating improved interpretability compared to the baseline across all neuron interpretability groups. See additional results in the supplementary.

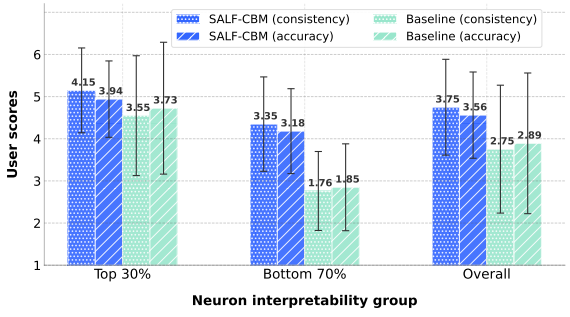


Figure 5. **User study results**.



Figure 6. **Explain Anything**. For each image, we prompted SALF-CBM with two different ROI masks produced by SAM [19] (red and blue regions). Our method provides accurate concept descriptions for each ROI.

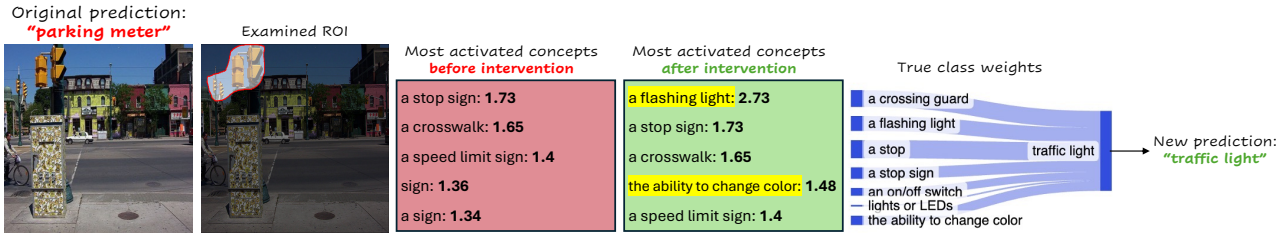


Figure 7. **Model debugging with Explain Anything.** We reveal that the model misclassified the image since it mistakenly identified traffic lights as street signs. Its prediction is corrected by locally editing the relevant concepts maps in the examined ROI.

4.4. Model exploration and debugging

We first qualitatively validate our *Explain Anything* feature (Section 3.6) on different images from the SAM dataset [19]. For each image, we prompt SALF-CBM (with ResNet-50 backbone pre-trained on ImageNet) with two different ROI masks automatically obtained by SAM [19], highlighted in red and blue. As shown in Figure 6, SALF-CBM generates informative region-specific concept descriptions that accurately correspond to the selected ROIs. For instance, in the child's drawing (left image), the dress (blue mask) and grass area (red mask) are correctly identified as fabric-like material and a field or lawn, respectively.

Next, we demonstrate the usefulness of Explain Anything in diagnosing classification errors, and facilitating targeted corrections using local user intervention. We present a case study from the ImageNet validation set, where our model miscalssified a "traffic light" image as a "parking meter", as shown in Figure 7. Applying Explain Anything to the traffic lights region in the image reveals that the model primarily detected sign-related concepts there. However, as indicated by the class weights visualization, these concepts are not associated with the correct "traffic light" class. This misalignment, along with the presence of street-related features in the image, led the model to incorrectly classify the image as a "parking meter." To rectify that, we locally edit two concepts maps associated with the true class - "a flashing light" and "the ability to change color" - within the selected ROI. Specifically, we increase their activation there by a correction factor of $\beta = 1$. As illustrated in the figure, this mild adjustment promoted these concepts to the top-5 most activated concepts in the ROI, subsequently adjusting the model's output to the correct class.

5. Conclusions

In this work, we presented SALF-CBM, a novel framework for transforming any vision neural network into an explainable model that provides both concept-based and spatial explanations for its predictions. We showed that SALF-CBM enhances model interpretability without compromising performance, outperforming both existing CBMs and the original model across several classification tasks. We demonstrated that it produces high-quality spatial explanations, achieving better zero-shot segmentation results compared to widely used heatmap-based explainability methods.

Additionally, we introduced interactive capabilities for model exploration and debugging, demonstrating their effectiveness in diagnosing and correcting model errors. We believe that such features are particularly valuable for high-stakes applications like medical imaging and autonomous driving. By providing expert practitioners with intuitive tools to understand and adjust model decisions, our approach can boost confidence and support safer deployment in these critical fields. Looking ahead, as new VLMs are developed across various domains, our findings could help inform the design of more powerful interpretability tools for a broad spectrum of AI applications. We plan to explore these directions in future work.

References

[1] Madiaga: Artificial intelligence act (2023). http://www.europarl.europa.eu/RegData/etudes/BRIE/2021/698792/EPRS_BRI(2021)698792_EN.pdf. 2

[2] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020. 2

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018. 2

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10 (7):e0130140, 2015. 2

[5] Sarah Adel Bargal, Andrea Zunino, Vitali Petsiuk, Jianming Zhang, Kate Saenko, Vittorio Murino, and Stan Sclaroff. Guided zoom: Zooming into network evidence to refine fine-grained model decisions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4196–4202, 2021. 2

[6] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, pages 63–71. Springer, 2016. 2, 6, 7

[7] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2956–2964, 2015. 2

[8] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 2, 6, 7

[9] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021. 2, 6

[10] Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *Advances in neural information processing systems*, 35: 2832–2845, 2022. 2

[11] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[12] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2711–2721, 2023. 3

[13] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019. 3

[14] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110:328–348, 2014. 6

[15] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 3

[16] Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: Evaluating the human interpretability of visual explanations. In *European Conference on Computer Vision*, pages 280–298. Springer, 2022. 2

[17] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*, pages 5338–5348. PMLR, 2020. 2, 3, 4

[18] Max Losch, Mario Fritz, and Bernt Schiele. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv preprint arXiv:1907.10882*, 2019. 2

[19] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024. 5, 8

[20] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017. 2

[21] T. Oikarinen and T-W. Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965*, 2022. 7

[22] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3, 4, 5, 6

[23] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning (ICML)*, pages 8748–8763, 2021. 2

[24] Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision*, pages 444–461. Springer, 2024. 7

[25] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE in-*

ternational conference on computer vision, pages 618–626, 2017. 2, 6, 7

[26] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017. 2

[27] A. Shtedritski, C. Rupprecht, and A. Vedaldi. What does clip know about a red circle? visual prompt engineering for VLMs, 2023. 2, 3

[28] Karen Simonyan. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[29] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise, 2017. 2

[30] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. *Advances in neural information processing systems*, 32, 2019. 2, 6, 7

[31] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 2, 6, 7

[32] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019. 2

[33] Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 10962–10971, 2023. 2

[34] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020. 2, 6, 7

[35] Eric Wong, Shibani Santurkar, and Aleksander Mądry. Leveraging sparse linear layers for debuggable deep networks. *arXiv preprint arXiv:2105.04857*, 2021. 4

[36] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022. 2, 3, 4, 5, 6

[37] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 2

[38] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11682–11690, 2021. 3

[39] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2

[40] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2