Visual-TCAV: Concept-based Attribution and Saliency Maps for Post-hoc Explainability in Image Classification

Antonio De Santis* **Riccardo Campi*** **Matteo Bianchi** **Marco Brambilla**
{antonio.desantis, riccardo.campi, matteo.bianchi, marco.brambilla}@polimi.it
Politecnico di Milano

Reviewed on OpenReview: *https://openreview.net/forum?id=SLhOOW5rhu*

Abstract

Convolutional Neural Networks (CNNs) have shown remarkable performance in image classification. However, interpreting their predictions is challenging due to the size and complexity of these models. State-of-the-art saliency methods generate local explanations highlighting the area in the input image where a class is identified but cannot explain how a concept of interest contributes to the prediction. On the other hand, concept-based methods, such as TCAV, provide insights into how sensitive the network is to a human-defined concept but cannot compute its attribution in a specific prediction nor show its location within the input image. We introduce Visual-TCAV, a novel explainability framework aiming to bridge the gap between these methods by providing both local and global explanations. Visual-TCAV uses Concept Activation Vectors (CAVs) to generate class-agnostic saliency maps that show where the network recognizes a certain concept. Moreover, it can estimate the attribution of these concepts to the output of any class using a generalization of Integrated Gradients. We evaluate the method's faithfulness via a controlled experiment where the ground truth for explanations is known, showing better ground truth alignment than TCAV. Our code is available at `https://github.com/DataSciencePolimi/Visual-TCAV`.

1 Introduction

As the performance of deep learning models has grown significantly over recent years, their complexity has also increased, making it difficult for users to understand how these models make decisions. As a result, they are often referred to as *black-box* models, since only their inputs and outputs are known, while their internal mechanisms are too complex for humans to comprehend. This lack of algorithmic transparency (von Eschenbach, 2021) has been shown to reduce trust in AI-based systems (Lipton, 2016), particularly in critical fields such as healthcare or autonomous driving in which neural networks are becoming increasingly employed (Turay & Vladimirova, 2022; Cai et al., 2020). Additionally, debugging models becomes challenging without comprehending the process they use to make predictions. To this end, the field of Explainable AI (XAI) has made significant progress in developing techniques for explaining black-box models (De Santis et al., 2025). However, determining whether a certain human-understandable concept is recognized by the network and how it influences the prediction remains a significant challenge. In computer vision, widely used explainability approaches use saliency maps to localize where a class is identified in an input image, but they can't explain which high-level features led the model to its prediction. For instance, these methods cannot determine whether a golf ball was recognized by the spherical shape, the dimples, or some other feature. To address this, Kim et al. (2018) introduced TCAV (Testing with Concept Activation Vectors), a method that can discern whether a user-defined concept (e.g., dimples, spherical) correlates positively with a selected class. However, TCAV cannot measure the importance of a concept in a specific prediction or show the locations within the input images where the concepts are recognized. Since global concept scores summarize the model's behavior by averaging over many inputs, they may not necessarily reflect the model's rationale

*Equal contribution.

for each individual instance. This is particularly important when investigating mispredictions, especially in critical domains such as medical imaging or autonomous driving, where analysts need to verify what led the model to a specific prediction and whether it was spatially grounded.

In this article, we introduce a novel explainability framework, namely Visual-TCAV, integrating the core principles of both saliency and concept-based methods to overcome their respective limitations. This framework can be applied to any layer of a CNN whose output is a set of feature maps. We design Visual-TCAV to satisfy three main desiderata for concept-based explanations: (D1) *spatial grounding*, by providing concept maps that localize where the network recognizes concepts of interest; (D2) *per-instance attribution*, by estimating how much each concept contributes to a class output for a given input; and (D3) *aggregatability*, by supporting the aggregation of per-instance attributions across inputs into global, class-level explanations.

2 Related Work

For image classification, early methods primarily provide explanations via saliency maps that highlight the most important regions in the input image to predict a certain class. A model-agnostic approach for generating such visualizations involves studying the model's input-output relationship by creating a set of perturbed versions of the input and analyzing how the output changes with each perturbation. Examples of this paradigm include LIME (Ribeiro et al., 2016), which uses random perturbations, and SHAP (Lundberg & Lee, 2017), which estimates the importance of each pixel using Shapley values. A different approach that instead tries to access the internal workings of the model was originally proposed by Simonyan et al. (2014) and consists of generating saliency maps based on the gradients of the model output w.r.t. the input images. This idea led many researchers (Springenberg et al., 2014; Smilkov et al., 2017) to investigate how to exploit gradients to produce more accurate saliency maps. Among them, Selvaraju et al. (2017) proposed Grad-CAM, a generalization of CAM (Zhou et al., 2016) which extracts the gradients of the logits (i.e., raw pre-softmax predictions) w.r.t. the feature maps and uses a Global Average Pooling (GAP) operation to transform these gradients into class-specific weights for each feature map. It then performs a weighted sum of these feature maps to produce a class localization map. However, Sundararajan et al. (2017) demonstrated that gradients can saturate, leading to an inaccurate assessment of feature importance. To address this issue, they introduced Integrated Gradients (IG), an axiomatic attribution method that works by integrating the gradients along a path from a baseline (e.g., a black image) to the actual input image, providing fine-grained saliency maps via per-pixel attribution.

Saliency methods provide per-instance, pixel-level explanations and class localization. They are therefore limited in connecting predictions to high-level semantic concepts and in supporting aggregation across inputs. To overcome these limitations, Kim et al. (2018) proposed TCAV, a method that investigates the correlations between user-defined concepts and the network's predictions using a set of example images representing a concept. For instance, example images of the concept "striped" can be used to determine whether the network is sensitive to this concept for predicting the "zebra" class. This is accomplished by learning a Concept Activation Vector (CAV), which is a vector orthogonal to the decision boundary of a linear classifier trained to differentiate between the feature maps of concept examples and random images. From this, a sensitivity score is computed using the signs of the dot products between the CAV and the gradients of a selected class w.r.t. the feature maps. TCAV has also been used to analyze concept-level biases and can be considered complementary to saliency maps. Indeed, while Grad-CAM and IG apply exclusively to individual predictions, TCAV provides only global explanations. However, TCAV does not show where concepts are identified within the input images, making it challenging to assess whether a high score can truly be attributed to the intended concept. Moreover, TCAV computes the network's sensitivity to a concept, but not the magnitude of its importance, as the score only depends on the signs of the directional derivatives. For instance, the concepts "grass" and "dimples" might have identical TCAV scores for the "golf ball" class, even if one contributes substantially more to the prediction.

TCAV has received attention within the XAI community, leading to various applications (Lucieri et al., 2020; Cai et al., 2019) and extensions (Crabbé & van der Schaar, 2022; Graziani et al., 2018). A notable extension that tries to add localization for a concept of interest is CAVLI (Shukla et al., 2023), which uses LIME to decompose images into superpixels and uses CAVs to assess their similarity to the concepts. In

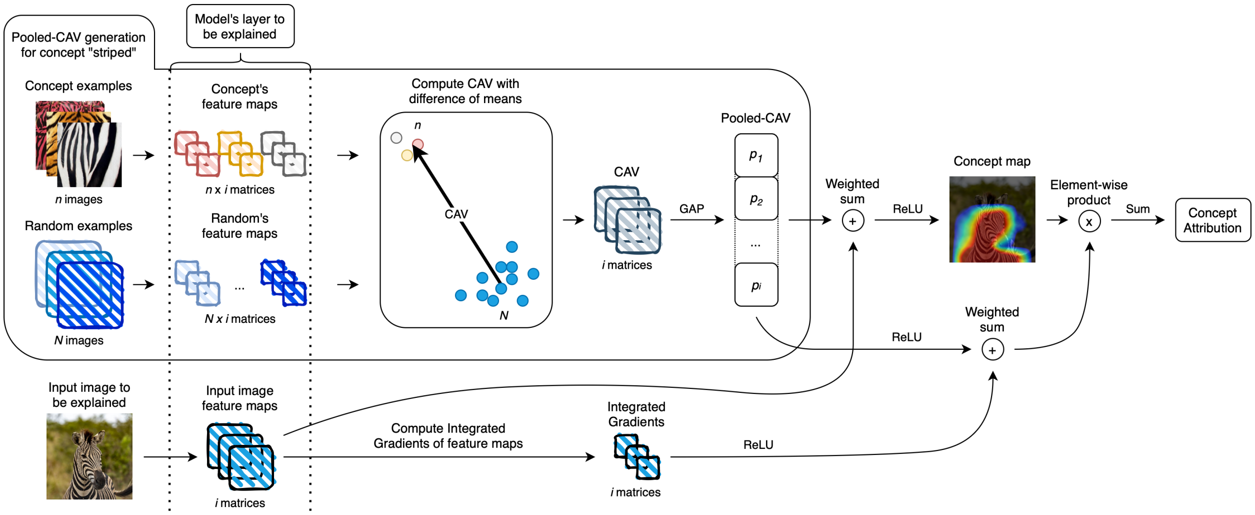


Figure 1: The Visual-TCAV pipeline for generating local explanations. A pooled-CAV is computed using the feature maps of user-defined concept examples and random images. A concept map is then produced through a weighted sum of the pooled-CAV and the image's feature maps. Finally, a concept attribution is obtained by extracting the IG attributions of the neurons that the concept activates using the pooled-CAV and the concept map, which is used as a spatial mask.

parallel, several unsupervised approaches have also been proposed (Ghorbani et al., 2019; Zhang et al., 2021; Fel et al., 2023b; Akula et al., 2020) to discover concepts automatically. These methods typically involve cropping images of a class to generate patches, which are then re-inserted into the network, and activations are clustered, resulting in a set of extracted concepts. More recently, Sparse Autoencoders (SAEs) have also shown promising results in discovering interpretable features across both vision (De Santis et al., 2026; Fel et al., 2023a) and language domains (Bricken et al., 2023; Bereska & Gavves, 2024). However, it is often the case that concepts of interest are not being extracted even though they are important for the task (Sharkey et al., 2025). For the scope of this paper, we will focus only on supervised scenarios in which the concept to study is manually defined by the user.

3 Visual-TCAV

This section presents Visual-TCAV, whose methodology is outlined in Fig. 1. Local explanations can be generated for any layer and consist of two key components: (i) the *concept map*, a saliency map that serves as a visual representation of where the network recognized the concept in the input image and (ii) the *concept attribution*, a numerical value that estimates the importance of the concept for a selected class. To derive global explanations, the process is replicated across multiple input images and the concept attributions are averaged to quantify how the concept influences the network's decisions across a wide range of inputs.

3.1 CAV Generation and Spatial Pooling

Following the original TCAV framework, the initial step of our method consists of computing a Concept Activation Vector (CAV) from a set of example images representing a user-defined concept and a set of negative examples (e.g., random images). Specifically, we use the *Difference of Means* method proposed by Martin & Weller (2019) to compute the CAV, as they demonstrated that this approach produces more consistent CAVs than logistic classifiers or SVMs. As the name suggests, this method uses the arithmetic mean to determine the centroids of both the concept's activations and the activations of random images. A CAV is then computed directly as the difference between the concept centroid and the random centroid. This is also mathematically equivalent to the approach proposed by Pahde et al. (2025). Since we are interested in identifying which feature maps are activated by the concept, irrespective of its spatial location within the

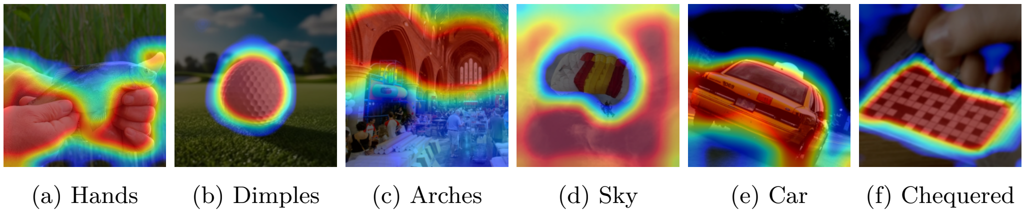

(a) Hands (b) Dimples (c) Arches (d) Sky (e) Car (f) Chequered

Figure 2: Examples of class-independent concept maps for various input images and concepts.

example images, we apply a GAP operation on the obtained CAV. The result is a vector of scalar values whose length is equal to the number of feature maps of the layer under consideration. Each element of this vector is associated with a feature map, and its raw value approximates the degree of correlation between that feature map and the concept. Moving forward, we refer to this vector as the *pooled-CAV*.

3.2 Concept Map

Using the pooled-CAV, we derive a concept map $\boldsymbol{M}^c$ that spatially localizes a concept c within the input image by applying Eq. (1). Specifically, we compute a weighted sum of the feature maps $\mathbf{F}$ from the selected layer, where each channel i is weighted by the corresponding element p_i^c of the pooled-CAV. Subsequently, we apply a ReLU function to retain only the regions positively correlated with the concept.

$$\boldsymbol{M}^c = ReLU\left(\sum_i p_i^c \cdot \mathbf{F}_{:,:,i}\right) \tag{1}$$

To enable meaningful comparisons of concept activation across different input images, we apply a custom min-max normalization to the concept map based on a predefined range, as shown in Eq. (2). The normalization range boundaries (ℓ^c and u^c) are derived from the example images provided by the user to represent the concept. Specifically, the upper bound u^c is computed by first generating concept maps from positive example images $\mathcal{D}_{pos}$, then calculating the median of their contraharmonic means (*chmean*), which serve to heuristically extract values of high activation within each concept map. The lower bound ℓ^c is computed using the same method but with the concept maps of negative example images $\mathcal{D}_{neg}$ instead. The normalized concept map $\hat{\boldsymbol{M}}^c$ is then obtained by clipping any value outside this predefined range and then scaling it to $[0, 1]$. The notation $[\boldsymbol{A}]_a^b$ denotes clipping each element of $\boldsymbol{A}$ within the interval $[a, b]$, i.e., $[\boldsymbol{A}]_a^b = min(max(\boldsymbol{A}, a), b)$, where the min and max are applied element-wise. An ablation study on different normalization approaches is provided in Appendix H.

$$\hat{\boldsymbol{M}}^c = \frac{[\boldsymbol{M}^c]_{\ell^c}^{u^c} - \ell^c}{u^c - \ell^c}, \quad \text{where} \quad \begin{cases} u^c = median\left(chmean_{i \in \mathcal{D}_{pos}}(\mathbf{M}^c_{:,:,i})\right) \\ \ell^c = median\left(chmean_{i \in \mathcal{D}_{neg}}(\mathbf{M}^c_{:,:,i})\right) \end{cases} \tag{2}$$

By overlaying the normalized concept map on the input image, we generate a class-independent visualization that highlights the region of the image where the network recognized the concept (examples are shown in Fig. 2). This allows us to know, for any input image, the concept's location and its degree of activation w.r.t. an ideal concept defined by the user. Additionally, concept maps act as a qualitative validation for whether the CAV represents the intended concept, without requiring activation maximization techniques (Mordvintsev et al., 2015) or sorting images based on their similarity to the CAV.

3.3 Concept Attribution

We aim to gain insights into the network's decision-making process by estimating the importance of a given concept for a target prediction. Concretely, for a chosen layer and target class, we first compute attribution scores for the layer's activations with respect to the target class logit, and then use the pooled-CAV to weight those attributions according to how strongly each feature map is associated with the concept.

Integrated Gradients in feature-map space. We compute feature maps attributions using Integrated Gradients (IG), but applied to the activation space of a chosen layer rather than to the input pixels. Let $\mathbf{F}$ denote the feature maps at the chosen layer for a given input image, and let $\mathbf{F}_0$ denote the baseline feature maps (in our case, an all-zero tensor with the same shape as $\mathbf{F}$). We denote by $z_k(\mathbf{F})$ the logit (pre-softmax score) of class k obtained by running the model forward from the chosen layer while setting that layer's feature maps to $\mathbf{F}$. We consider a straight-line path in the feature maps space from $\mathbf{F}_0$ to $\mathbf{F}$, and compute the integrated gradients of the target class logit with respect to the feature maps along this path. As shown in Eq. (3), we use the IG definition adapted from Sundararajan et al. (2017) to operate in feature map space. Since IG satisfies the completeness axiom, the attributions sum to the difference between the target class's logit and the baseline's logit for that class, regardless of which layer is used as input. Given a target class k, we refer to attributions w.r.t. feature maps of a specific input image as $\mathbf{IG}^k$.

$$\mathbf{IG}^k = (\mathbf{F} - \mathbf{F}_0) \odot \int_{\alpha=0}^{1} \frac{\partial\, z_k(\mathbf{F}_0 + \alpha(\mathbf{F} - \mathbf{F}_0))}{\partial \mathbf{F}}\, d\alpha. \tag{3}$$

In practice, this integral is approximated numerically by evaluating gradients at a finite number of steps along the interpolation path using the Riemann trapezoidal rule.

Normalizing attribution magnitude. Since the scale of these attributions is the same as the raw logits, making interpretation difficult, we normalize the attributions so that their sum is between 0 and 1. We retain only the positive attributions, as we focus on evidence supporting the target class[1]. We then compute the difference between the logits of the input and those of the baseline, followed by a ReLU and $[0, 1]$ scaling as shown in Eqs. (4a) and (4b), where $\boldsymbol{z}(\mathbf{F})$ denotes the vector of logits obtained by running the model forward from the chosen layer while setting that layer's feature maps to $\mathbf{F}$.

$$\Delta \boldsymbol{z} = ReLU\left(\boldsymbol{z}(\mathbf{F}) - \boldsymbol{z}(\mathbf{F}_0)\right) \tag{4a}$$

$$\Delta \hat{\boldsymbol{z}} = \frac{\Delta \boldsymbol{z}}{max(\Delta \boldsymbol{z})} \tag{4b}$$

The attributions for the target class (k) are then scaled so that their sum equals the normalized difference $\Delta \hat{z}_k$ for that class, as shown in Eq. (5a).

$$\hat{\mathbf{IG}}^k = \frac{ReLU(\mathbf{IG}^k)}{\sum ReLU(\mathbf{IG}^k)} \cdot \Delta \hat{z}_k \tag{5a}$$

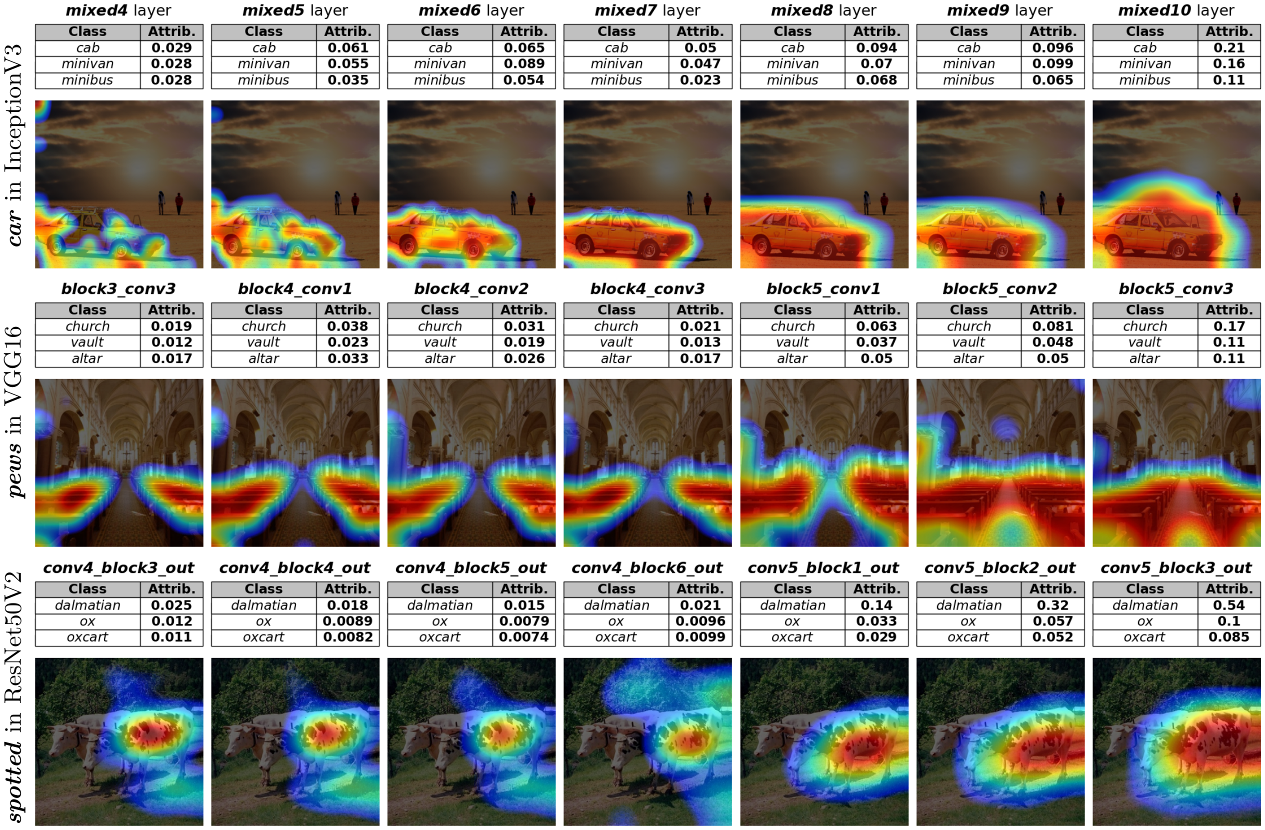
$$\hat{\boldsymbol{p}}^c = \frac{ReLU(\boldsymbol{p}^c)}{max(\boldsymbol{p}^c)} \tag{5b}$$

Weighting attributions by concept alignment. To estimate the attribution of a concept c for a target class k, we utilize the pooled-CAV as a per-channel vector of weights to perform a weighted sum of the normalized attributions $\hat{\mathbf{IG}}^k$ toward that target class. Before weighting, we apply a ReLU and max normalization to the pooled-CAV, as shown in Eq. (5b), to progressively down-weight feature maps that are less aligned with the concept, and emphasize those that are strongly concept-related. We refer to the normalized pooled-CAV as $\hat{\boldsymbol{p}}^c$. The rationale behind using the ReLU here is to keep only feature maps whose activation is positively aligned with the concept.

Combining attribution, concept weighting, and spatial masking. As shown in Eq. (6), we compute the concept attribution $a^{c,k}$ by first aggregating the normalized attributions $\hat{\mathbf{IG}}^k$ across feature maps using the normalized pooled-CAV weights $\hat{p}_i^c$, obtaining a single spatial attribution map for the target class. We then multiply this map element-wise by the normalized concept map $\hat{\boldsymbol{M}}^c$, which acts as a spatial mask. Finally, we sum over all spatial locations to obtain a scalar score that quantifies the attribution of concept-aligned activations to the target class prediction.

$$a^{c,k} = \sum \Big(\hat{\boldsymbol{M}}^c \odot \sum_i (\hat{p}_i^c \cdot \hat{\mathbf{IG}}^k_{:,:,i}) \Big) \tag{6}$$

[1] For binary classifiers with a single output head, we treat the negative class score as $-z$. For the chosen target class (either z or $-z$), we keep only positive IG values and normalize by their sum as in Eq. (5a).



mixed4 layer

Class	Attrib.
cab	**0.029**
minivan	**0.028**
minibus	**0.028**

mixed5 layer

Class	Attrib.
cab	**0.061**
minivan	**0.055**
minibus	**0.035**

mixed6 layer

Class	Attrib.
cab	**0.065**
minivan	**0.089**
minibus	**0.054**

mixed7 layer

Class	Attrib.
cab	**0.05**
minivan	**0.047**
minibus	**0.023**

mixed8 layer

Class	Attrib.
cab	**0.094**
minivan	**0.07**
minibus	**0.068**

mixed9 layer

Class	Attrib.
cab	**0.096**
minivan	**0.099**
minibus	**0.065**

mixed10 layer

Class	Attrib.
cab	**0.21**
minivan	**0.16**
minibus	**0.11**

block3_conv3

Class	Attrib.
church	**0.019**
vault	**0.012**
altar	**0.017**

block4_conv1

Class	Attrib.
church	**0.038**
vault	**0.023**
altar	**0.033**

block4_conv2

Class	Attrib.
church	**0.031**
vault	**0.019**
altar	**0.026**

block4_conv3

Class	Attrib.
church	**0.021**
vault	**0.013**
altar	**0.017**

block5_conv1

Class	Attrib.
church	**0.063**
vault	**0.037**
altar	**0.05**

block5_conv2

Class	Attrib.
church	**0.081**
vault	**0.048**
altar	**0.05**

block5_conv3

Class	Attrib.
church	**0.17**
vault	**0.11**
altar	**0.11**

conv4_block3_out

Class	Attrib.
dalmatian	**0.025**
ox	**0.012**
oxcart	**0.011**

conv4_block4_out

Class	Attrib.
dalmatian	**0.018**
ox	**0.0089**
oxcart	**0.0082**

conv4_block5_out

Class	Attrib.
dalmatian	**0.015**
ox	**0.0079**
oxcart	**0.0074**

conv4_block6_out

Class	Attrib.
dalmatian	**0.021**
ox	**0.0096**
oxcart	**0.0099**

conv5_block1_out

Class	Attrib.
dalmatian	**0.14**
ox	**0.033**
oxcart	**0.029**

conv5_block2_out

Class	Attrib.
dalmatian	**0.32**
ox	**0.057**
oxcart	**0.052**

conv5_block3_out

Class	Attrib.
dalmatian	**0.54**
ox	**0.1**
oxcart	**0.085**

Figure 3: Layer-wise local explanations for various concepts and architectures. We compute the attribution of each concept for the top three predicted classes, ranked by class probability, and the last seven layers.

Interpretation and global aggregation. The concept attribution is a per-concept metric, meaning that two concepts can have significantly different attributions even if they are recognized in the same location of the input image, resulting in similar concept maps. Since concepts may naturally overlap in the same spatial regions, attributions are not meant to form a spatial partition of the image. This is natural in CNNs, where multiple channels may activate in the same region while detecting different patterns. Furthermore, since concepts can be correlated or hierarchically related (e.g., “car” and “wheel”), concept attributions are not assumed to be additive, and attributions should not be interpreted as a decomposition of the prediction that can be summed across concepts.

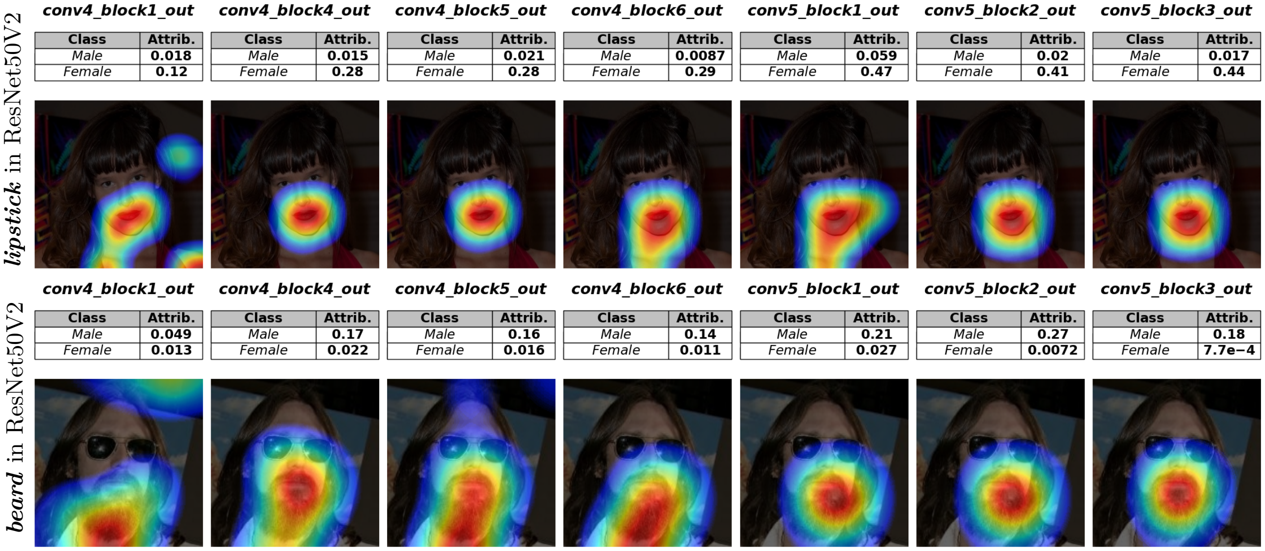
Finally, to obtain a global explanation, we can average the concept attribution across multiple input images to measure the overall importance of a concept for a selected class. For instance, we can calculate a global attribution of the “striped” concept for the “zebra” target class by averaging the attribution of “striped” across a large number (e.g., 200) of images containing zebras.

4 Results

This section presents the results of applying Visual-TCAV to a range of widely used CNNs. We show that our method (i) can reveal how concepts contribute to mispredictions, (ii) provides insights into where concepts are detected within the network, and (iii) aligns with ground truth in controlled experiments.

4.1 Experimental Setup

Our experiments are conducted on ResNet50V2 (He et al., 2016), InceptionV3 (Szegedy et al., 2016), and VGG16 (Simonyan & Zisserman, 2015), all pre-trained on the ImageNet (Deng et al., 2009) dataset, as well as



Class	Attrib.
Male	0.018
Female	0.12

Class	Attrib.
Male	0.015
Female	0.28

Class	Attrib.
Male	0.021
Female	0.28

Class	Attrib.
Male	0.0087
Female	0.29

Class	Attrib.
Male	0.059
Female	0.47

Class	Attrib.
Male	0.02
Female	0.41

Class	Attrib.
Male	0.017
Female	0.44

Class	Attrib.
Male	0.049
Female	0.013

Class	Attrib.
Male	0.17
Female	0.022

Class	Attrib.
Male	0.16
Female	0.016

Class	Attrib.
Male	0.14
Female	0.011

Class	Attrib.
Male	0.21
Female	0.027

Class	Attrib.
Male	0.27
Female	0.0072

Class	Attrib.
Male	0.18
Female	7.7e−4

Figure 4: Layer-wise local explanations for a ResNet50V2 (He et al., 2016) model trained on the CelebA (Liu et al., 2015) dataset for celebrity gender classification. The concepts tested are "lipstick" and "beard", both significantly contributing to the prediction.

a ResNet50V2 model trained from scratch (i.e., without ImageNet weights) on the CelebA (Liu et al., 2015) dataset for celebrity gender classification. The tested concepts were sourced from the Describable Textures Dataset (DTD) (Cimpoi et al., 2014), obtained from popular image search engines, or generated with Stable Diffusion (Rombach et al., 2022) (more on this in Appendix E). For concepts collected via search engines, we use 30 example images per concept, which can be considered a reasonable amount, as shown in the stability experiment provided in Appendix G. For DTD concepts, we use all 120 available images, and for generated concepts, we use 200 images. For ImageNet negative examples, we follow the approach recommended by Martin & Weller (2019), selecting 500 random images. In the case of CelebA, images containing the concept are used as positive examples, while those without it serve as negative examples. Regarding the computation of the integrated gradients, we used 300 steps, which are usually enough to approximate the integral within a 5% error margin (Sundararajan et al., 2017). Experiments are conducted on an Intel i7 13700k with an RTX 4060Ti 16GB and 32 GB of RAM, using TensorFlow 2.15.1, CUDA 12.2, and Python 3.11.5. With this setup, local explanations for 7 layers take 1 to 2 minutes, while global ones with 200 class images and 7 layers take from 5 to 30 minutes depending on the model. For global explanations, the computation time remains nearly constant regardless of the number of concepts processed simultaneously.

4.2 Local Explanations

While concept maps are class-independent, the attribution of each concept depends on the class considered. In our examples, we examine the top three predicted classes and apply Visual-TCAV to a subset of the CNN layers. The rationale behind these layer-wise explanations is that they enable us to visualize where specific concepts are learned within the network. For instance, the "car" concept in Fig. 3 is not well recognized in earlier layers, corroborating previous findings that higher-level features are typically extracted deeper into the network (Zeiler & Fergus, 2014; Olah et al., 2017; Bau et al., 2017; Bianchi et al., 2024). On the other hand, concept maps in earlier layers are more fine-grained (see the "pews" concept in Fig. 3) due to their neurons having smaller receptive fields. Additionally, we observe a substantial increase in attributions in deeper layers, which is aligned with other studies (Morcos et al., 2018; Amjad et al., 2022) showing deeper neurons exhibit higher class selectivity due to their proximity to the output. However, this is slightly less pronounced on the CelebA dataset (refer to Fig. 4), where the "lipstick" and "beard" concepts exhibit class selectivity even in earlier layers, which may be due to the lower task complexity. The utility of these explanations lies mainly in their ability to show not only which image regions the network is focusing on,

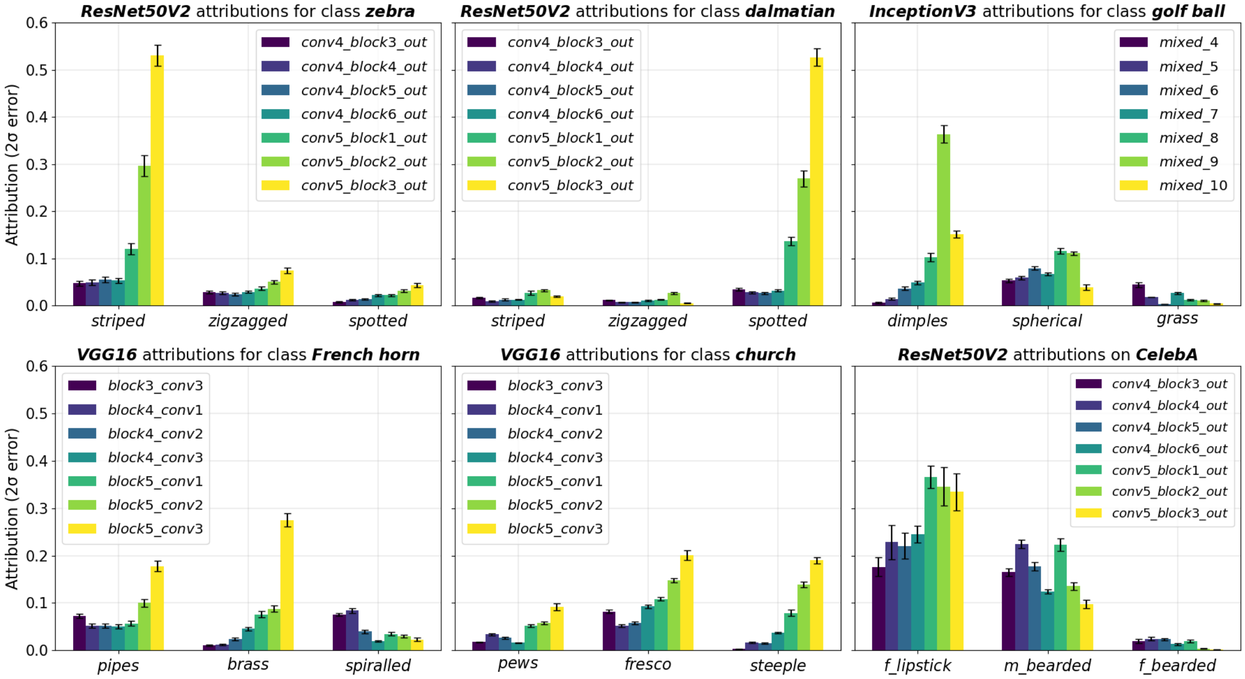


Figure 5: Global explanations for various concepts, classes, and networks. Each bar chart shows the attributions of three concepts for a specific class across the last seven layers of each network. In the CelebA dataset, "m_" denotes attribution toward the "male" class, while "f_" indicates attribution toward the "female" class. Each concept's attribution is averaged over 200 input images.

but also what the model recognizes in those regions and how much this contributes to each class's output. This can be especially useful for revealing the model's reasoning behind mispredictions. For instance, the third image in Fig. 3 is an "ox" wrongly classified as "dalmatian" for which we observe that the network's decision is largely influenced by the "spotted" concept, which accounts for more than half the logit value of the "dalmatian" class.

4.3 Global Explanations

The concept attribution is a per-concept metric of importance; thereby, we can derive global explanations by aggregating this attribution across multiple input images of a selected class. In our experiments, we utilize 200 images per class for each global explanation. For concepts that are inherently part of the class (e.g., "striped" for "zebra", or "dimples" for "golf ball"), we can directly use any image representing that class. On the other hand, for concepts that appear sporadically, we only use images where the concept is present. For instance, we only use images of church interiors for "pews" and "fresco" concepts, and images of church exteriors for the "steeple" concept. This ensures that the explanations are independent of the frequency of the concept's appearance in the class images.

Results of global explanations for various concepts are provided in Fig. 5. The attributions match intuitive expectations, considering, for instance, the importance of "striped" for "zebra" or "spotted" for "dalmatian". However, such high attribution may suggest the model over-relies on certain concepts, using them as shortcuts for class discrimination. This may lead the network to misclassify out-of-distribution cases, such as the "ox" being incorrectly labeled as a dalmatian in Fig. 3. This may also pose issues for concepts like "lipstick" in gender classification, potentially leading to biased predictions. Furthermore, while the final layer typically has the highest attribution, which is expected for class discriminative concepts, there are instances, such as "dimples" and "spherical" for "golf ball", where concepts recognized in the earlier layers have a greater impact on the network's prediction. Empirically, we found such behavior to be dependent not only on the

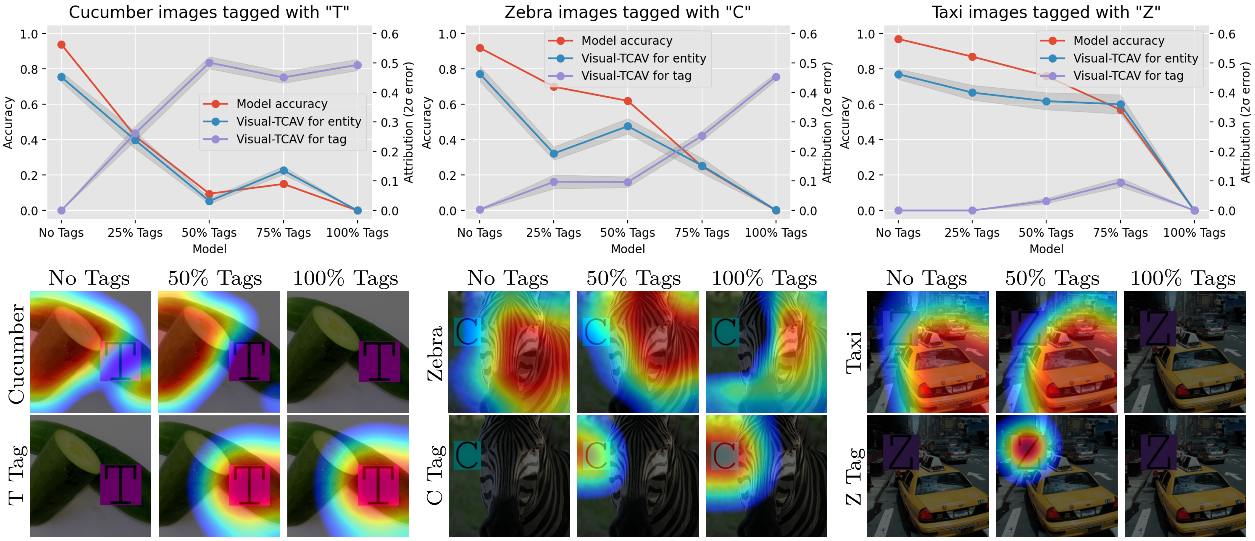


Figure 6: The results of the validation experiment. The upper section of the figure shows the models' accuracies and the Visual-TCAV attributions for both entities and tags across all models. The lower section provides examples of concept maps for the no tags model, 50% tags model, and 100% tags model.

concept but also on the model architecture. More examples of local and global explanations can be found in Appendices C and D.

4.4 Quantitative Evaluation with Ground Truth

We conducted a validation experiment to evaluate the faithfulness of Visual-TCAV. In this experiment, we train a series of convolutional networks in a controlled setting, where the ground truth for explanations is known, and assess whether the Visual-TCAV attributions match this ground truth. For this purpose, we create a dataset of three classes – cucumber, taxi, and zebra – which are the same classes used in the TCAV validation experiment. We then create multiple versions of this dataset by altering a percentage of the images with a tag, represented by a letter enclosed in a randomly sized square, added in a random location of the image. Specifically, zebra images are tagged with a "Z" in a purple square, taxi images with a "T" in a magenta square, and cucumber images with a "C" in a cyan square. From these tagged images, we create five datasets: one of images without tags, and four others with 25%, 50%, 75%, and 100% of tagged images, respectively. Each dataset is then used to train a different model, each including six convolutional layers and a GAP layer. Depending on the dataset used for training, each model may learn to recognize either the entities (i.e., cucumbers, taxis, and zebras), the tags, or both, and will decide which ones to give more importance. To obtain an approximated ground truth assessing which concept – entity or tag – is more important, we ask the models to classify a set of 200 incorrectly tagged test images per class. In this test set, taxis are tagged with the "Z", cucumbers are tagged with the "T", and zebras are tagged with the "C". If the network accuracy remains high, it indicates that the entity is more important than the tag, and thus, we expect its attribution to be higher. On the other hand, if the performance deteriorates on these wrongly tagged images, it indicates that the tag is more important than the entity. Hence, the tag's attribution should be higher. We obtain the CAVs for entities using images of each class as concept examples and the other two classes with random tags as negative examples. For tags, we use random images containing that tag as concept examples and images of cucumbers, taxis, and zebras containing the other two tags as negative examples. We use the same incorrectly tagged test set to compute the concept attributions for both entities and tags across the last convolutional layer of all models.

The results are shown in Fig. 6 and, as expected, an increase in the percentage of tagged images correlates with a decrease in accuracy for all classes. In particular, for the "cucumber" class, the accuracy declines much faster than other classes, with most images being incorrectly classified as taxis. This suggests that

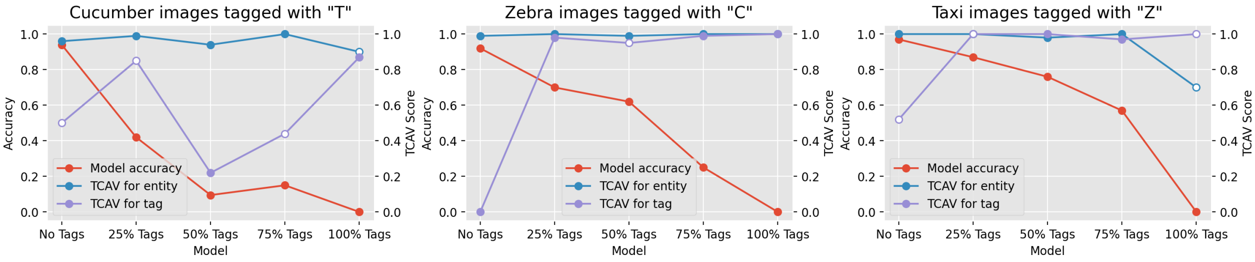


Figure 7: TCAV scores for tags and entities across models trained on a different percentage of tagged images. Results that did not reach statistical significance (p-value < 0.05) according to the TCAV test are indicated with a white dot.

even the models trained on a small fraction of tagged images tend to overfit on the "T" tag. The concept attributions for both the "cucumber" entity and the "T" tag closely mirror this ground truth. The "zebra" entity and the "C" tag are also consistent with the ground truth: the attributions for "zebra" show a positive correlation with accuracy, whereas the attributions for the "C" tag demonstrate a clear inverse correlation. Notably, the networks did not pay much attention to the "Z" tag, focusing instead on the absence of the other two tags to classify zebras. Indeed, the model trained with 100% of images tagged classifies any image without a "C" or a "T" tag as "zebra", regardless of whether the "Z" tag is present or not. This is confirmed by our method, which assigns an attribution of nearly zero to both the "Z" tag and the "taxi" entity for the aforementioned model. We also tested saliency methods, such as Grad-CAM and IG, to further validate these findings. These methods do not highlight the "Z" tag either, but rather the entire image, in search of the "zebra" class (see Appendix B). For all models excluding the one trained with 100% of tags, the accuracy for "taxi" remains high, implying that these models are indeed capable of recognizing the "taxi" entity. The concept attribution for the "taxi" entity aligns with this observation. In the lower part of Fig. 6, we provide examples of concept maps for the models trained with a different percentage of tagged images. As expected, the model trained without tags can recognize the entities but not the tags, the model trained with 50% tagged images can recognize both, and the model trained with 100% tagged images struggles to recognize the entities but effectively identifies the "T" and "C" tags.

4.4.1 Comparative Analysis with TCAV

The primary difference between our concept attribution and the TCAV score is that our method considers the magnitude of the gradients, not just their direction. This allows us to measure the concept's impact on the predictions, beyond just the network's sensitivity to it. To demonstrate this, we compute the TCAV scores for tags and entities across each validation model (see Fig. 7). For the model trained without tags, the TCAV scores align with ground truth, showing high sensitivity to entities and no or negative sensitivity to tags (where 0.5 indicates no sensitivity and 0 signifies negative sensitivity). However, for the other models, TCAV struggles to capture variations in concept importance as defined by ground truth. These models yield high TCAV scores for both entities and tags, despite the decline in accuracy, indicating that their contributions to predictions differ significantly. This limitation arises because TCAV relies solely on the signs of gradients, making it incapable of distinguishing the relative importance of these concepts. Therefore, while the TCAV score may be valuable in explaining whether the network learned to correlate a certain concept with a class, our concept attribution provides a more comprehensive explanation by quantifying the extent to which a given concept influences the network's decisions. A qualitative and quantitative comparison with saliency methods (Grad-CAM and Integrated Gradients) is provided in Appendix F.

5 Conclusion

In this article, we presented Visual-TCAV, a novel method that provides concept-based local and global explanations for image classification models by estimating the attribution of user-defined concepts to the network's predictions. Visual-TCAV also generates saliency maps to show where concepts are identified by the network, providing users with visual evidence that CAVs align with their intended concepts. Our method's effectiveness was demonstrated in widely used CNNs and a ground truth experiment, where it successfully identified the most important concept in each examined model.

5.1 Limitations and Future Work

One limitation is that collecting example images requires some effort and domain knowledge and may also introduce some subjectivity. To address this, future work could explore generating CAVs directly from text using methods such as Text-To-Concept (Moayeri et al., 2023), which aligns the activations of multimodal models like CLIP (Radford et al., 2021) with vision-only models. Following recent work on synthetic CAVs (Campi et al., 2025), another promising direction is fine-tuning text-to-image generative models with a few real examples using DreamBooth (Ruiz et al., 2023) to generate images better aligned with the analyst's intended concept for CAV training. A further limitation is that, since concept attributions are not additive, Visual-TCAV does not provide a completeness guarantee over an arbitrary user-defined concept set, and therefore cannot verify that the tested concepts are exhaustive of what the model has learned.

As future work, we plan to investigate the applicability of our method to other tasks like regression as well as other architectures like Vision Transformers (ViTs) (Dosovitskiy et al., 2021). For ViTs in particular, a natural adaptation could operate on patch tokens to obtain concept maps. However, token mixing through self-attention implies that patch tokens can encode evidence originating from other regions, making spatial grounding less reliable compared to CNN feature maps (Jeanneret et al., 2025; Darcet et al., 2024). Since addressing this would require non-trivial design adaptations, we leave a systematic ViT evaluation to a dedicated future study. It may also be interesting to study interconnections between concepts to determine not only the concept attribution toward a class but also toward other concepts in deeper layers.

References

Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2594–2601, Apr. 2020. doi: 10.1609/aaai.v34i03.5643. URL https://ojs.aaai.org/index.php/AAAI/article/view/5643.

Rana Ali Amjad, Kairen Liu, and Bernhard C. Geiger. Understanding neural networks and individual neuron importance via information-ordered cumulative ablation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):7842–7852, 2022. doi: 10.1109/TNNLS.2021.3088685.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3319–3327, 2017. doi: 10.1109/CVPR.2017.354.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ePUVetPKu6. Survey Certification, Expert Certification.

Matteo Bianchi, Antonio De Santis, Andrea Tocchetti, and Marco Brambilla. Interpretable network visualizations: A human-in-the-loop approach for post-hoc explainability of cnn-based image classification. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 3715–3723. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/411. URL https://doi.org/10.24963/ijcai.2024/411. Main Track.

Trenton Bricken, Adly Templeton, Ben Chen, Jack Lindsey, Adam Jermyn, Shan Carter, Tom Henighan, Adam Pearce, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. https://transformer-circuits.pub/2023/monosemantic-features, 2023. Transformer Circuits Thread, Anthropic.

Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pp. 258–262. Association for Computing Machinery, 2019. ISBN 9781450362726. doi: 10.1145/3301275.3302289. URL https://doi.org/10.1145/3301275.3302289.

Lei Cai, Jingyang Gao, and Di Zhao. A review of the application of deep learning in medical image classification and segmentation. *Annals of Translational Medicine*, 8(11), 2020. ISSN 2305-5847. URL https://atm.amegroups.com/article/view/36944.

Riccardo Campi, Santiago Borrego, Antonio De Santis, Matteo Bianchi, Andrea Tocchetti, and Marco Brambilla. Towards synthetic concept activation vectors via generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2745–2753, June 2025.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Jonathan Crabbé and Mihaela van der Schaar. Concept activation regions: A generalized framework for concept-based explanations. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 2590–2607. Curran Associates, Inc., 2022.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

Antonio De Santis, Riccardo Campi, Matteo Bianchi, Andrea Tocchetti, and Marco Brambilla. 2 - foundational approaches to post-hoc explainability for image classification. In William Lawless, Ranjeev Mittu, Donald Sofge, and Marco Brambilla (eds.), *Bi-directionality in Human-AI Collaborative Systems*, pp. 23–54. Academic Press, 2025. ISBN 978-0-443-40553-2. doi: https://doi.org/10.1016/B978-0-44-340553-2.00008-3. URL https://www.sciencedirect.com/science/article/pii/B9780443405532000083.

Antonio De Santis, Schrasing Tong, Marco Brambilla, and Lalana Kagal. Learning concept bottleneck models from mechanistic explanations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=gdEWoxhb70.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

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*, 2021.

Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023a. Curran Associates Inc.

Thomas Fel, Agustin Picard, Louis Béthune, 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 (CVPR)*, pp. 2711–2721, June 2023b.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Mara Graziani, Vincent Andrearczyk, and Henning Müller. Regression concept vectors for bidirectional explanations in histopathology. In Danail Stoyanov, Zeike Taylor, Seyed Mostafa Kia, Ipek Oguz, Mauricio Reyes, Anne Martel, Lena Maier-Hein, Andre F. Marquand, Edouard Duchesnay, Tommy Löfstedt, Bennett Landman, M. Jorge Cardoso, Carlos A. Silva, Sergio Pereira, and Raphael Meier (eds.), *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pp. 124–132, Cham, 2018. Springer International Publishing. ISBN 978-3-030-02628-8.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 630–645, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

Guillaume Jeanneret, Loic Simon, and Frederic Jurie. Disentangling Visual Transformers: Patch-Level Interpretability for Image Classification . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2670–2680, Los Alamitos, CA, USA, June 2025. IEEE Computer Society. doi: 10.1109/CVPRW67362.2025.00252. URL https://doi.ieeecomputersociety.org/10.1109/CVPRW67362.2025.00252.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

Zachary Lipton. The mythos of model interpretability. *Communications of the ACM*, 61, 10 2016. doi: 10.1145/3233231.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 07 2020. doi: 10.1109/IJCNN48605.2020.9206946.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Tyler Martin and Adrian Weller. *Interpretable Machine Learning*. M.Phil. diss., Dept. of Engineering, University of Cambridge, August 2019. URL https://www.mlmi.eng.cam.ac.uk/files/tam_final_reduced.pdf.

Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Ari S. Morcos, David G.T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1iuQjxCZ.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015. URL https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Frederik Pahde, Maximilian Dreyer, Moritz Weckbecker, Leander Weber, Christopher J. Anders, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. Navigating neural space: Revisiting concept activation vectors to overcome directional divergence. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Q95MaWfF4e.

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 Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Fatih Porikli, Sandra Skaff, Alireza Entezari, Jianyuan Min, Daisuke Iwai, Amela Sadagic, Carlos Scheidegger, and Tobias Isenberg (eds.), *Advances in Visual Computing*, pp. 234–244, Cham, 2016. Springer International Publishing. ISBN 978-3-319-50835-1.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, August 2016. doi: 10.1145/2939672.2939778. URL http://dx.doi.org/10.1145/2939672.2939778.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042.

N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.02155. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.02155.

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 *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017. doi: 10.1109/iccv.2017.74. URL http://dx.doi.org/10.1109/ICCV.2017.74.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, Eric J Michaud, Stephen Casper, Max Tegmark, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Thomas McGrath. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=91H76m9Z94. Survey Certification.

Pushkar Shukla, Sushil Bharati, and Matthew Turk. Cavli - using image associations to produce local concept-based explanations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3750–3755, 2023. doi: 10.1109/CVPRW59228.2023.00387.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL http://arxiv.org/abs/1706.03825.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3319–3328. JMLR.org, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Tolga Turay and Tanya Vladimirova. Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey. *IEEE Access*, 10:14076–14119, 2022. doi: 10.1109/ACCESS.2022.3147495.

Warren von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34, 12 2021. doi: 10.1007/s13347-021-00477-0.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A. Ehinger, and Benjamin I. P. Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11682–11690, May 2021. doi: 10.1609/aaai.v35i13.17389. URL https://ojs.aaai.org/index.php/AAAI/article/view/17389.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization . In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, Los Alamitos, CA, USA, June 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.319. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.319.

A Appendix Overview

In the appendix, we provide:

B. IG and Grad-CAM for 100% tags model

C. Additional results of Local Explanations

D. Additional results of Global Explanations

E. Example images for generated concepts

F. Comparison with IG and Grad-CAM

G. Experiments on Visual-TCAV stability

H. Ablation study of concept map normalization

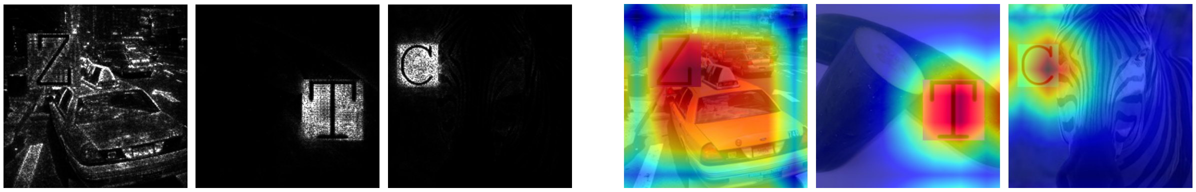
I. IG baseline ablation

J. IG steps ablation

K. C-Insertion and C-Deletion faithfulness experiment

B Saliency methods for 100% tags model

We provide the results obtained by applying IG (Sundararajan et al., 2017) and Grad-CAM (Selvaraju et al., 2017) to the 100% tags model (see Fig. 8). These methods align with Visual-TCAV in showing that this model does not pay attention to the “Z”, but rather to the absence of the “T” and the “C” for predicting the “zebra” class.



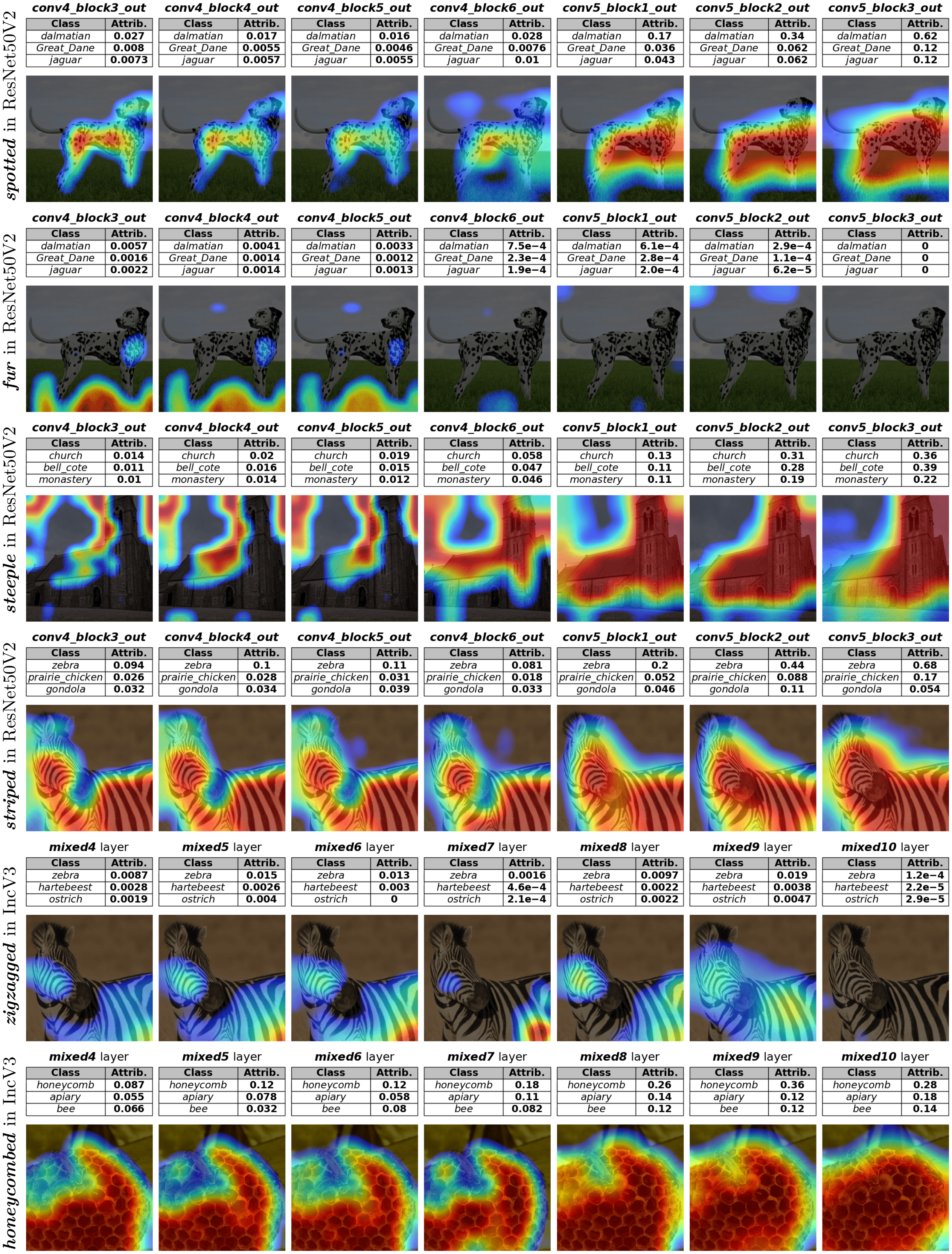
(a) Integrated Gradients

(b) Grad-CAM

Figure 8: IG and Grad-CAM for the model with 100% tags, searching respectively for the classes "zebra", "taxi", and "cucumber". Both methods highlight the "T" for class "taxi" and the "C" for class "cucumber", but fail to recognize the "Z" for class "zebra".

C Additional results of Local Explanations

Continuing from the results presented in the main paper, we further provide additional local explanations for more input images and concepts in Figs. 9 to 11.



spotted in ResNet50V2

conv4_block3_out

Class	Attrib.
dalmatian	**0.027**
Great_Dane	**0.008**
jaguar	**0.0073**

conv4_block4_out

Class	Attrib.
dalmatian	**0.017**
Great_Dane	**0.0055**
jaguar	**0.0057**

conv4_block5_out

Class	Attrib.
dalmatian	**0.016**
Great_Dane	**0.0046**
jaguar	**0.0055**

conv4_block6_out

Class	Attrib.
dalmatian	**0.028**
Great_Dane	**0.0076**
jaguar	**0.01**

conv5_block1_out

Class	Attrib.
dalmatian	**0.17**
Great_Dane	**0.036**
jaguar	**0.043**

conv5_block2_out

Class	Attrib.
dalmatian	**0.34**
Great_Dane	**0.062**
jaguar	**0.062**

conv5_block3_out

Class	Attrib.
dalmatian	**0.62**
Great_Dane	**0.12**
jaguar	**0.12**

fur in ResNet50V2

conv4_block3_out

Class	Attrib.
dalmatian	**0.0057**
Great_Dane	**0.0016**
jaguar	**0.0022**

conv4_block4_out

Class	Attrib.
dalmatian	**0.0041**
Great_Dane	**0.0014**
jaguar	**0.0014**

conv4_block5_out

Class	Attrib.
dalmatian	**0.0033**
Great_Dane	**0.0012**
jaguar	**0.0013**

conv4_block6_out

Class	Attrib.
dalmatian	**7.5e−4**
Great_Dane	**2.3e−4**
jaguar	**1.9e−4**

conv5_block1_out

Class	Attrib.
dalmatian	**6.1e−4**
Great_Dane	**2.8e−4**
jaguar	**2.0e−4**

conv5_block2_out

Class	Attrib.
dalmatian	**2.9e−4**
Great_Dane	**1.1e−4**
jaguar	**6.2e−5**

conv5_block3_out

Class	Attrib.
dalmatian	**0**
Great_Dane	**0**
jaguar	**0**

steeple in ResNet50V2

conv4_block3_out

Class	Attrib.
church	**0.014**
bell_cote	**0.011**
monastery	**0.01**

conv4_block4_out

Class	Attrib.
church	**0.02**
bell_cote	**0.016**
monastery	**0.014**

conv4_block5_out

Class	Attrib.
church	**0.019**
bell_cote	**0.015**
monastery	**0.012**

conv4_block6_out

Class	Attrib.
church	**0.058**
bell_cote	**0.047**
monastery	**0.046**

conv5_block1_out

Class	Attrib.
church	**0.13**
bell_cote	**0.11**
monastery	**0.11**

conv5_block2_out

Class	Attrib.
church	**0.31**
bell_cote	**0.28**
monastery	**0.19**

conv5_block3_out

Class	Attrib.
church	**0.36**
bell_cote	**0.39**
monastery	**0.22**

striped in ResNet50V2

conv4_block3_out

Class	Attrib.
zebra	**0.094**
prairie_chicken	**0.026**
gondola	**0.032**

conv4_block4_out

Class	Attrib.
zebra	**0.1**
prairie_chicken	**0.028**
gondola	**0.034**

conv4_block5_out

Class	Attrib.
zebra	**0.11**
prairie_chicken	**0.031**
gondola	**0.039**

conv4_block6_out

Class	Attrib.
zebra	**0.081**
prairie_chicken	**0.018**
gondola	**0.033**

conv5_block1_out

Class	Attrib.
zebra	**0.2**
prairie_chicken	**0.052**
gondola	**0.046**

conv5_block2_out

Class	Attrib.
zebra	**0.44**
prairie_chicken	**0.088**
gondola	**0.11**

conv5_block3_out

Class	Attrib.
zebra	**0.68**
prairie_chicken	**0.17**
gondola	**0.054**

zigzagged in IncV3

mixed4 layer

Class	Attrib.
zebra	**0.0087**
hartebeest	**0.0028**
ostrich	**0.0019**

mixed5 layer

Class	Attrib.
zebra	**0.015**
hartebeest	**0.0026**
ostrich	**0.004**

mixed6 layer

Class	Attrib.
zebra	**0.013**
hartebeest	**0.003**
ostrich	**0**

mixed7 layer

Class	Attrib.
zebra	**0.0016**
hartebeest	**4.6e−4**
ostrich	**2.1e−4**

mixed8 layer

Class	Attrib.
zebra	**0.0097**
hartebeest	**0.0022**
ostrich	**0.0022**

mixed9 layer

Class	Attrib.
zebra	**0.019**
hartebeest	**0.0038**
ostrich	**0.0047**

mixed10 layer

Class	Attrib.
zebra	**1.2e−4**
hartebeest	**2.2e−5**
ostrich	**2.9e−5**

honeycombed in IncV3

mixed4 layer

Class	Attrib.
honeycomb	**0.087**
apiary	**0.055**
bee	**0.066**

mixed5 layer

Class	Attrib.
honeycomb	**0.12**
apiary	**0.078**
bee	**0.032**

mixed6 layer

Class	Attrib.
honeycomb	**0.12**
apiary	**0.058**
bee	**0.08**

mixed7 layer

Class	Attrib.
honeycomb	**0.18**
apiary	**0.11**
bee	**0.082**

mixed8 layer

Class	Attrib.
honeycomb	**0.26**
apiary	**0.14**
bee	**0.12**

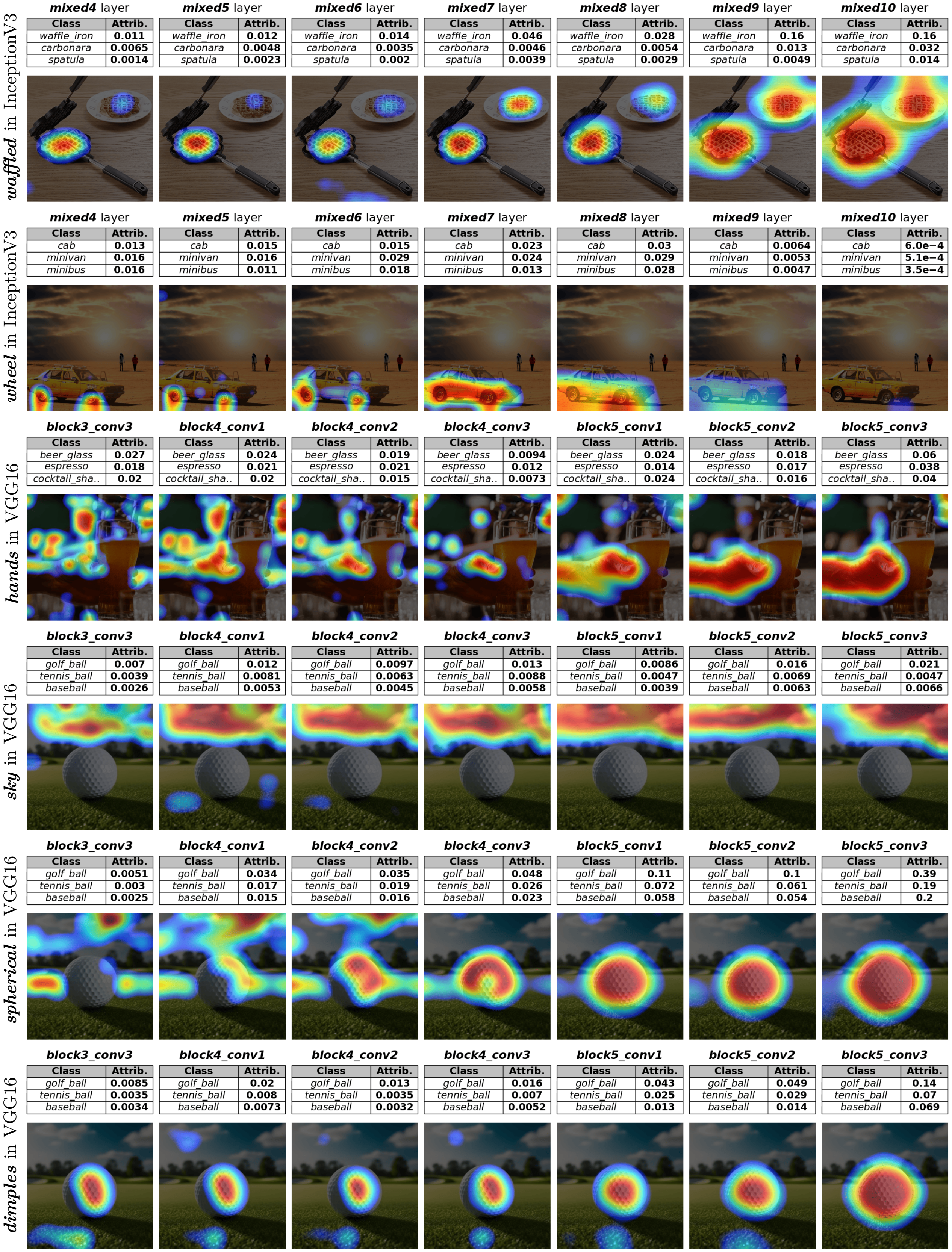
mixed9 layer

Class	Attrib.
honeycomb	**0.36**
apiary	**0.12**
bee	**0.12**

mixed10 layer

Class	Attrib.
honeycomb	**0.28**
apiary	**0.18**
bee	**0.14**

Figure 9: More examples of layer-wise local explanations for various concepts and networks. (Part 1)



mixed4 layer		***mixed5*** layer		***mixed6*** layer		***mixed7*** layer		***mixed8*** layer		***mixed9*** layer		***mixed10*** layer	
Class	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**
waffle_iron	**0.011**	*waffle_iron*	**0.012**	*waffle_iron*	**0.014**	*waffle_iron*	**0.046**	*waffle_iron*	**0.028**	*waffle_iron*	**0.16**	*waffle_iron*	**0.16**
carbonara	**0.0065**	*carbonara*	**0.0048**	*carbonara*	**0.0035**	*carbonara*	**0.0046**	*carbonara*	**0.0054**	*carbonara*	**0.013**	*carbonara*	**0.032**
spatula	**0.0014**	*spatula*	**0.0023**	*spatula*	**0.002**	*spatula*	**0.0039**	*spatula*	**0.0029**	*spatula*	**0.0049**	*spatula*	**0.014**

mixed4 layer		***mixed5*** layer		***mixed6*** layer		***mixed7*** layer		***mixed8*** layer		***mixed9*** layer		***mixed10*** layer	
Class	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**
cab	**0.013**	*cab*	**0.015**	*cab*	**0.015**	*cab*	**0.023**	*cab*	**0.03**	*cab*	**0.0064**	*cab*	**6.0e−4**
minivan	**0.016**	*minivan*	**0.016**	*minivan*	**0.029**	*minivan*	**0.024**	*minivan*	**0.029**	*minivan*	**0.0053**	*minivan*	**5.1e−4**
minibus	**0.016**	*minibus*	**0.011**	*minibus*	**0.018**	*minibus*	**0.013**	*minibus*	**0.028**	*minibus*	**0.0047**	*minibus*	**3.5e−4**

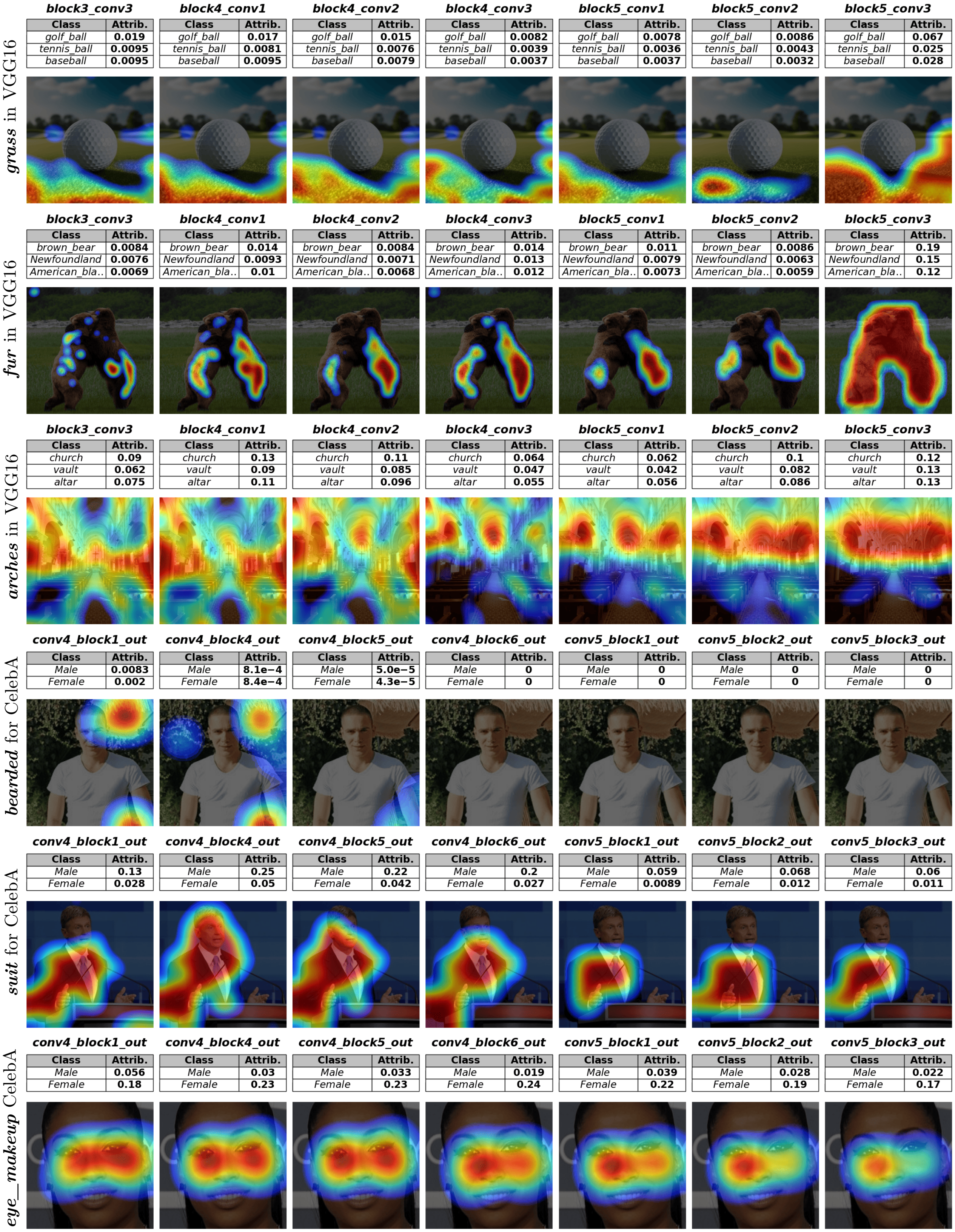
block3_conv3		***block4_conv1***		***block4_conv2***		***block4_conv3***		***block5_conv1***		***block5_conv2***		***block5_conv3***	
Class	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**
beer_glass	**0.027**	*beer_glass*	**0.024**	*beer_glass*	**0.019**	*beer_glass*	**0.0094**	*beer_glass*	**0.024**	*beer_glass*	**0.018**	*beer_glass*	**0.06**
espresso	**0.018**	*espresso*	**0.021**	*espresso*	**0.021**	*espresso*	**0.012**	*espresso*	**0.014**	*espresso*	**0.017**	*espresso*	**0.038**
cocktail_sha..	**0.02**	*cocktail_sha..*	**0.02**	*cocktail_sha..*	**0.015**	*cocktail_sha..*	**0.0073**	*cocktail_sha..*	**0.024**	*cocktail_sha..*	**0.016**	*cocktail_sha..*	**0.04**

block3_conv3		***block4_conv1***		***block4_conv2***		***block4_conv3***		***block5_conv1***		***block5_conv2***		***block5_conv3***	
Class	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**
golf_ball	**0.007**	*golf_ball*	**0.012**	*golf_ball*	**0.0097**	*golf_ball*	**0.013**	*golf_ball*	**0.0086**	*golf_ball*	**0.016**	*golf_ball*	**0.021**
tennis_ball	**0.0039**	*tennis_ball*	**0.0081**	*tennis_ball*	**0.0063**	*tennis_ball*	**0.0088**	*tennis_ball*	**0.0047**	*tennis_ball*	**0.0069**	*tennis_ball*	**0.0047**
baseball	**0.0026**	*baseball*	**0.0053**	*baseball*	**0.0045**	*baseball*	**0.0058**	*baseball*	**0.0039**	*baseball*	**0.0063**	*baseball*	**0.0066**

block3_conv3		***block4_conv1***		***block4_conv2***		***block4_conv3***		***block5_conv1***		***block5_conv2***		***block5_conv3***	
Class	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**
golf_ball	**0.0051**	*golf_ball*	**0.034**	*golf_ball*	**0.035**	*golf_ball*	**0.048**	*golf_ball*	**0.11**	*golf_ball*	**0.1**	*golf_ball*	**0.39**
tennis_ball	**0.003**	*tennis_ball*	**0.017**	*tennis_ball*	**0.019**	*tennis_ball*	**0.026**	*tennis_ball*	**0.072**	*tennis_ball*	**0.061**	*tennis_ball*	**0.19**
baseball	**0.0025**	*baseball*	**0.015**	*baseball*	**0.016**	*baseball*	**0.023**	*baseball*	**0.058**	*baseball*	**0.054**	*baseball*	**0.2**

block3_conv3		***block4_conv1***		***block4_conv2***		***block4_conv3***		***block5_conv1***		***block5_conv2***		***block5_conv3***	
Class	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**	**Class**	**Attrib.**
golf_ball	**0.0085**	*golf_ball*	**0.02**	*golf_ball*	**0.013**	*golf_ball*	**0.016**	*golf_ball*	**0.043**	*golf_ball*	**0.049**	*golf_ball*	**0.14**
tennis_ball	**0.0035**	*tennis_ball*	**0.008**	*tennis_ball*	**0.0035**	*tennis_ball*	**0.007**	*tennis_ball*	**0.025**	*tennis_ball*	**0.029**	*tennis_ball*	**0.07**
baseball	**0.0034**	*baseball*	**0.0073**	*baseball*	**0.0032**	*baseball*	**0.0052**	*baseball*	**0.013**	*baseball*	**0.014**	*baseball*	**0.069**

Figure 10: More examples of layer-wise local explanations for various concepts and networks. (Part 2)



grass in VGG16

block3_conv3		block4_conv1		block4_conv2		block4_conv3		block5_conv1		block5_conv2		block5_conv3	
Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.
golf_ball	0.019	golf_ball	0.017	golf_ball	0.015	golf_ball	0.0082	golf_ball	0.0078	golf_ball	0.0086	golf_ball	0.067
tennis_ball	0.0095	tennis_ball	0.0081	tennis_ball	0.0076	tennis_ball	0.0039	tennis_ball	0.0036	tennis_ball	0.0043	tennis_ball	0.025
baseball	0.0095	baseball	0.0095	baseball	0.0079	baseball	0.0037	baseball	0.0037	baseball	0.0032	baseball	0.028

fur in VGG16

block3_conv3		block4_conv1		block4_conv2		block4_conv3		block5_conv1		block5_conv2		block5_conv3	
Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.
brown_bear	0.0084	brown_bear	0.014	brown_bear	0.0084	brown_bear	0.014	brown_bear	0.011	brown_bear	0.0086	brown_bear	0.19
Newfoundland	0.0076	Newfoundland	0.0093	Newfoundland	0.0071	Newfoundland	0.013	Newfoundland	0.0079	Newfoundland	0.0063	Newfoundland	0.15
American_bla..	0.0069	American_bla..	0.01	American_bla..	0.0068	American_bla..	0.012	American_bla..	0.0073	American_bla..	0.0059	American_bla..	0.12

arches in VGG16

block3_conv3		block4_conv1		block4_conv2		block4_conv3		block5_conv1		block5_conv2		block5_conv3	
Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.
church	0.09	church	0.13	church	0.11	church	0.064	church	0.062	church	0.1	church	0.12
vault	0.062	vault	0.09	vault	0.085	vault	0.047	vault	0.042	vault	0.082	vault	0.13
altar	0.075	altar	0.11	altar	0.096	altar	0.055	altar	0.056	altar	0.086	altar	0.13

bearded for CelebA

conv4_block1_out		conv4_block4_out		conv4_block5_out		conv4_block6_out		conv5_block1_out		conv5_block2_out		conv5_block3_out	
Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.
Male	0.0083	Male	8.1e−4	Male	5.0e−5	Male	0	Male	0	Male	0	Male	0
Female	0.002	Female	8.4e−4	Female	4.3e−5	Female	0	Female	0	Female	0	Female	0

suit for CelebA

conv4_block1_out		conv4_block4_out		conv4_block5_out		conv4_block6_out		conv5_block1_out		conv5_block2_out		conv5_block3_out	
Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.
Male	0.13	Male	0.25	Male	0.22	Male	0.2	Male	0.059	Male	0.068	Male	0.06
Female	0.028	Female	0.05	Female	0.042	Female	0.027	Female	0.0089	Female	0.012	Female	0.011

eye_makeup CelebA

conv4_block1_out		conv4_block4_out		conv4_block5_out		conv4_block6_out		conv5_block1_out		conv5_block2_out		conv5_block3_out	
Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.	Class	Attrib.
Male	0.056	Male	0.03	Male	0.033	Male	0.019	Male	0.039	Male	0.028	Male	0.022
Female	0.18	Female	0.23	Female	0.23	Female	0.24	Female	0.22	Female	0.19	Female	0.17

Figure 11: More examples of layer-wise local explanations for various concepts and networks. (Part 3)

D Additional results of Global Explanations

In Figure 12 we present additional global explanations for various classes and concepts.

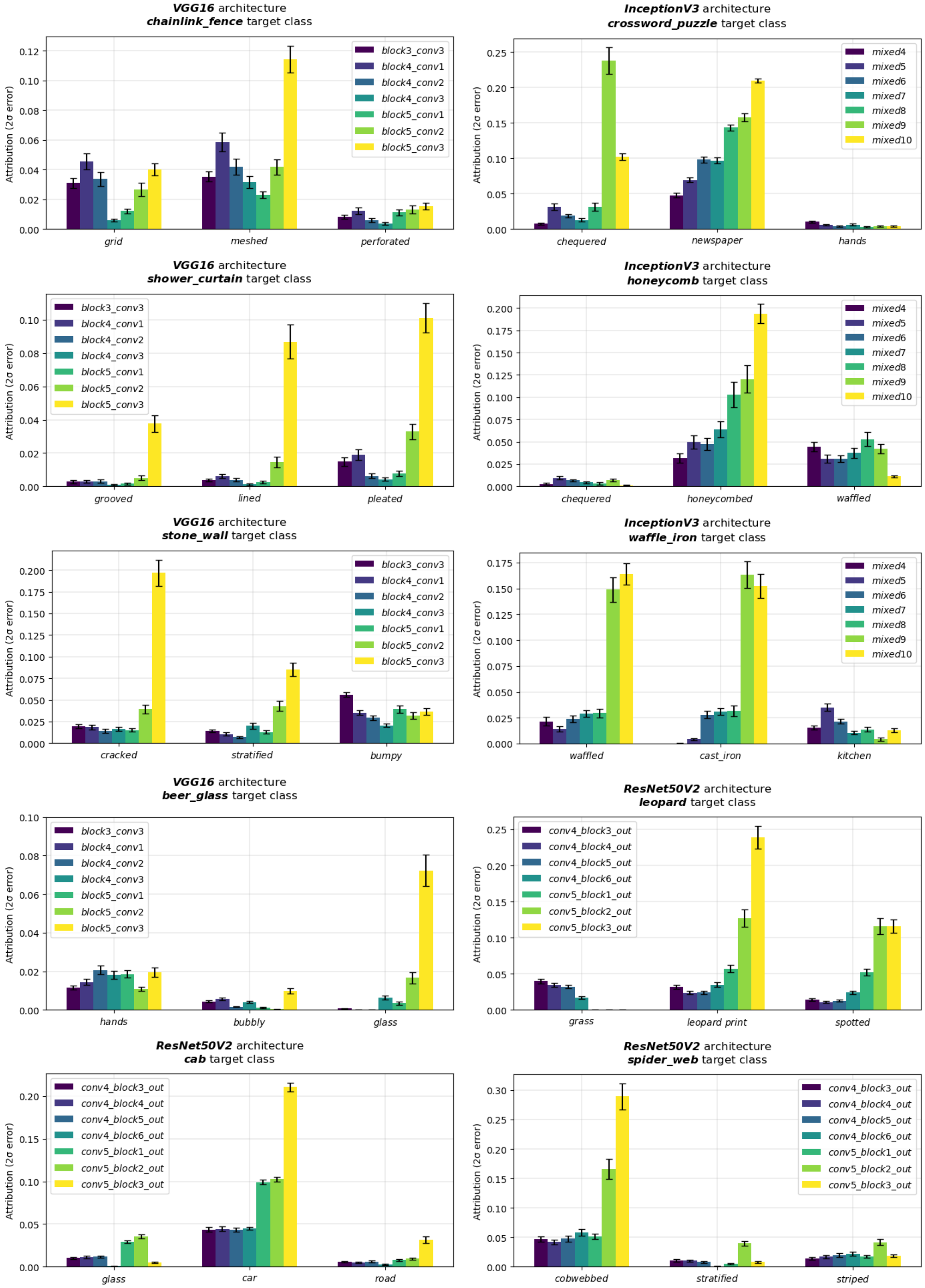
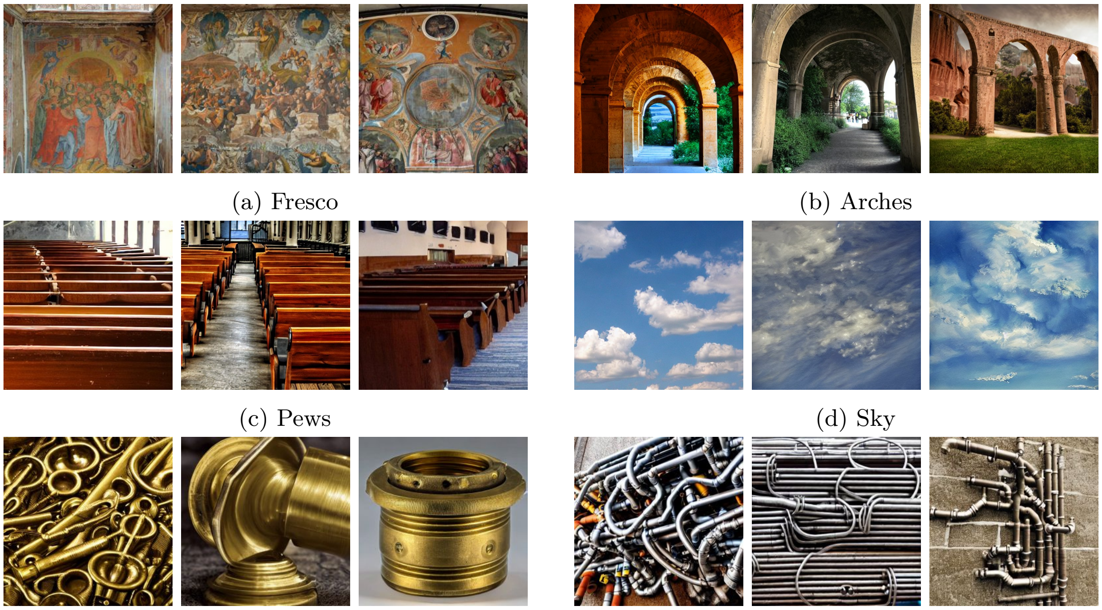


Figure 12: More examples of global explanations for various classes, concepts, and networks.

E Example images for generated concepts

Some of the concepts used in the paper were generated using Stable Diffusion v1.5 (Rombach et al., 2022) with default parameters. In particular, we generated the following concepts: “pews”, “fresco”, “arches”, “sky”, “pipes”, and “brass”. We used just the concept name as a prompt and generated 200 images per concept. A subsequent manual revision was still necessary to eliminate errors and artifacts. In Fig. 13, we provide three example images for each generated concept.



(a) Fresco

(b) Arches

(c) Pews

(d) Sky

(e) Brass

(f) Pipes

Figure 13: We provide three example images for each concept generated with Stable Diffusion v1.5.

F Comparison with IG and Grad-CAM

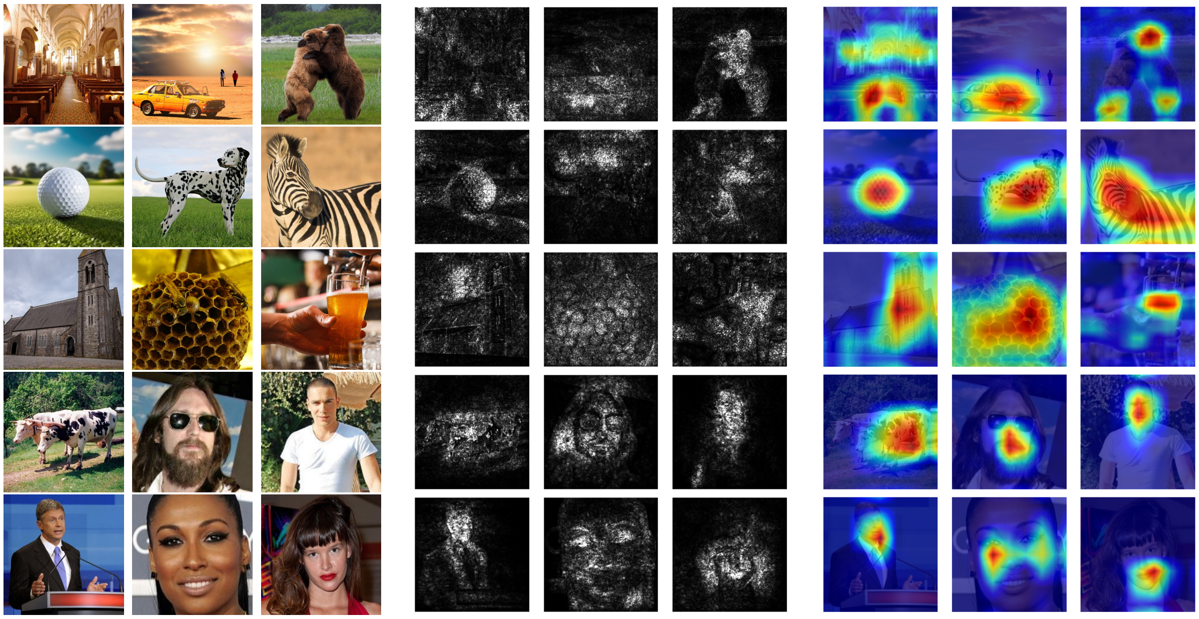
F.1 Qualitative comparison

In Figure 14, we apply IG and Grad-CAM to some of the images analyzed in the paper, with the aim of qualitatively assessing the differences in output and scope between these methods and Visual-TCAV. As discussed in previous sections, saliency methods are among the most widely used approaches for explaining black-box image classifiers, as they enable the visualization of the most relevant regions of an image for a given prediction. However, these methods provide limited insights beyond identifying these regions. Specifically, they do not explain what features or concepts the model is identifying in those regions, how these features contribute to a specific prediction, nor can they provide global explanations.

Considering the “church (interiors)” image, IG reveals that the model primarily focuses on the pews, while Grad-CAM indicates attention to both the pews and the arches, although with lesser emphasis on the latter. These observations align with the results derived from Visual-TCAV (Figs. 3 and 11), which show that the concepts “pews” and “arches” are indeed important for the prediction, with “pews” more than “arches”. Moreover, Visual-TCAV also provides a quantitative analysis of their contributions, explains “what” the network is seeing in the image in terms of the example images in Figs. 13b and 13c, and offers a global explanation by computing the average attribution of these concepts across multiple church images. Furthermore, from Visual-TCAV, we can also derive that the “arches” concept is not class discriminative in this case, as it has a similar importance for the classes “vault” and “altar”.

Often, key concepts may also remain undetected by saliency methods. For example, in the case of the “golf ball” and “bear” images, saliency methods fail to reveal that the “grass” concept is somewhat relevant for the former and that the "fur" concept is highly significant for the latter. Instead, saliency maps mainly highlight the golf ball itself, while Grad-CAM focuses on the bear's head and feet. Additionally, these methods are unable to discern whether the model recognizes the dimples on the golf ball, its spherical shape, or both, since these features visually overlap. In contrast, Visual-TCAV enables such distinctions by attributing these overlapping features to specific concepts with different importance.

Considering the saliency maps for the CelebA dataset, especially those generated by Grad-CAM, they are all very similar, making it difficult to understand whether the model learned to recognize specific attributes such as "beard", "lipstick", or "suit". However, while Visual-TCAV can provide explanations in terms of such concepts, other facial features can be very complex and are also difficult to define as human-understandable concepts, making this task challenging to interpret for any state-of-the-art XAI method.



(a) Original Images

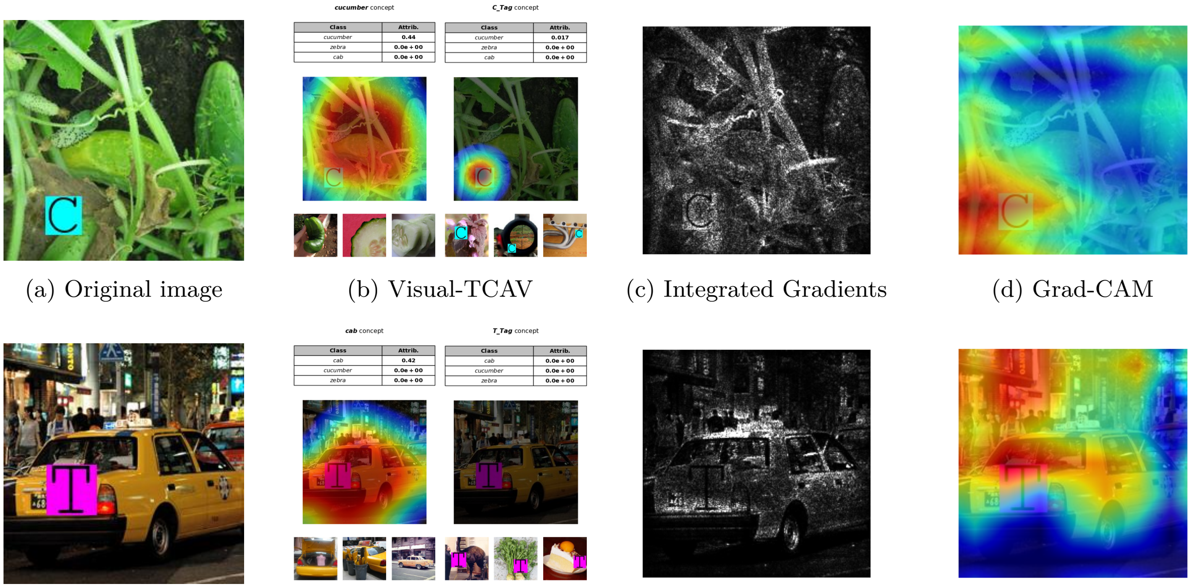
(b) Integrated Gradients

(c) Grad-CAM

Figure 14: IG and Grad-CAM for the images tested in the paper. Explanations of "church" (interiors), "bear", and "golf ball" are from VGG-16. Explanations of "church" (outside), "dalmatian", "ox", and "zebra" are from ResNet50V2. Explanations of "cab", "honeycomb", and "waffle iron" are from InceptionV3. All explanations are computed in the last convolutional layer, considering the predicted class as the target class.

F.2 Quantitative comparison

In this section, we present the results of a quantitative experiment designed to evaluate and compare the effectiveness of Visual-TCAV, IG, and Grad-CAM in communicating which concepts are more important in a given prediction. To perform this experiment, we took the "No Tags" and "100% Tags" models from the experiment in Section 4.4 and considered the classes "cucumber" and "taxi" with their respective "C" and "T" tags, as for these classes and models, we have a clear ground truth for explanations. Indeed, from Fig. 6, we know by construction that for the "No Tags" model, the entity itself (i.e., cucumbers and taxis) is more important than the tag, while for the "100% Tags" model, the tag is more important than the entity. We selected a total of 60 images, equally split across the two classes, and generated explanations using Visual-TCAV, IG, and Grad-CAM for the two models (for a total of 360 unique explanations). Each image-explanation pair was then presented to a Multimodal Large Language Model (GPT-5 in our case), which was asked to rate, based on the explanation, which concept between the entity and the tag appeared to be more important for the model's decision. A possible answer was also "Don't know" in case, based on the explanation, it was too difficult to tell which concept was more important.



(a) Original image (b) Visual-TCAV (c) Integrated Gradients (d) Grad-CAM

(e) Original image (f) Visual-TCAV (g) Integrated Gradients (h) Grad-CAM

Figure 15: Explanations from Visual-TCAV, IG, and Grad-CAM for the model trained without tags. In this model, the approximated ground truth is that the model pays much more attention to the entity than the tag (which was not present in the training set). While with Visual-TCAV the explanation matches this ground truth, this is not clear from IG and Grad-CAM.

For the "No Tags" model, the ground truth label was the entity, while for the "100% Tags" model, the ground truth label was the tag. "Don't know" responses were counted as incorrect. As for the metrics considered, for each method, we computed its accuracy (i.e., fraction of correct judgment over all images) and a statistical significance applying McNemar's test between Visual-TCAV and the two baselines (Grad-CAM and IG). The results in terms of accuracy are shown in Table 1, while the results of the McNemar's tests are shown in Table 2. Fig. 15 shows two examples where saliency maps communicated the wrong concept importance. Overall, Visual-TCAV was able to communicate correctly which concept was more important 98.33% of the time, while saliency methods (Grad-CAM and Integrated Gradients) achieved substantially lower accuracies of 71.67% and 65.83%, respectively. This suggests that explanations based solely on saliency maps may be misleading, which is the same conclusion reached in a similar experiment with human subjects by Kim et al. (2018). Statistical analysis using McNemar's test confirms that these results are statistically significant with p-values $\ll 0.05$. The exact prompts used in this experiment are provided below:

Visual-TCAV: `You are an evaluator judging which visual concept a classifier relied on more strongly for a prediction based on the output of an explainability method.`
```
The first image is the input to the classifier and was predicted as class <class>.

The explainability method used is called Visual-TCAV, and given a concept, it provides a heatmap showing whether the network
recognized that concept and where, and the attribution of the concept towards each class.
The second image is the explanation for the concept <concept1>, and the third is the explanation for the concept <concept2>.
In the explanations, below the heatmap, are shown examples of images containing the concept being analyzed.
Question: Based only on these explanations, which concept appears more responsible for the model's decision?
Answer only with the name of the concept, or if you can't tell from the explanation, then answer "na".
```

IG & Grad-CAM: `You are an evaluator judging which visual concept a classifier relied on more strongly for a prediction based on the output of an explainability method.`
```
The first image is the input to the classifier and was predicted as class <class>.
The second image is the explanation.

The explainability method used is called <method>, and it provides a saliency map highlighting the most important image
region for the prediction.
Question: Based only on the explanation, which feature appears more responsible for the model's decision between <concept1>
and <concept2>?
Answer only with the name of the concept, or if you can't tell from the explanation, then answer "na".
```

Table 1: Comparison of accuracy and number of "don't know" responses across methods. Visual-TCAV achieves consistently higher accuracy than pure saliency approaches on both classes and overall.

Class	Method	Accuracy (%)	Don't know (%)
cucumber	Visual-TCAV	96.67	1.67
cucumber	Grad-CAM	60.00	23.33
cucumber	Integrated Gradients	51.67	0.00
taxi	Visual-TCAV	100.0	0.00
taxi	Grad-CAM	83.33	6.67
taxi	Integrated Gradients	80.00	0.00
Overall	**Visual-TCAV**	**98.33**	**0.83**
Overall	Grad-CAM	71.67	15.00
Overall	Integrated Gradients	65.83	0.00

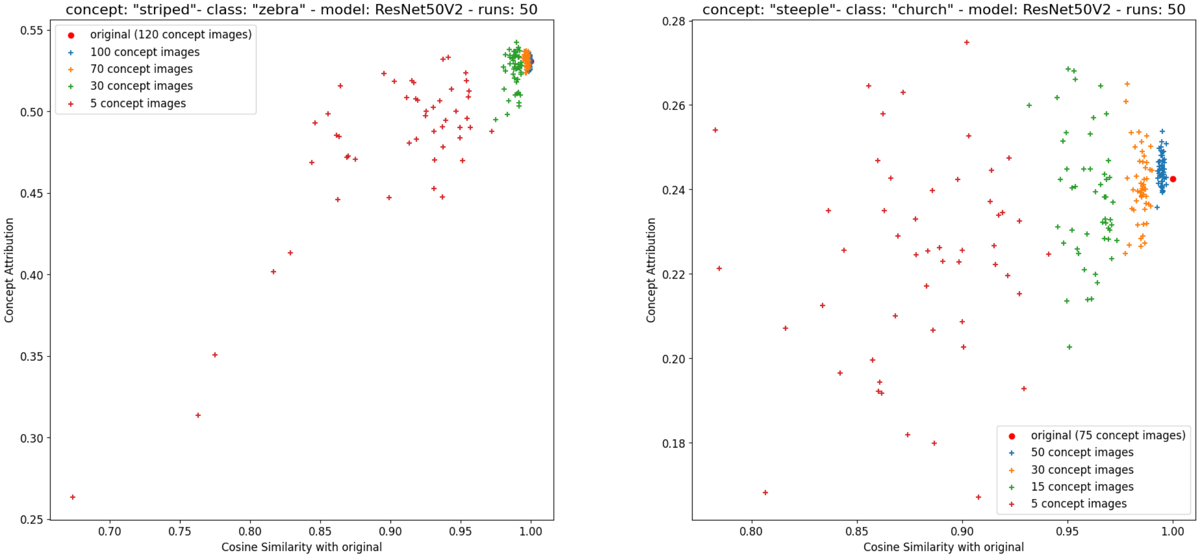
Table 2: McNemar tests comparing Visual-TCAV against Grad-CAM and IG. b = (Visual-TCAV correct, other method wrong) and c = (other method correct, Visual-TCAV wrong).

Class	Comparison	b	c	p-value
cucumber	Visual-TCAV vs Grad-CAM	24	2	1.05×10^{-5}
cucumber	Visual-TCAV vs IG	27	0	1.49×10^{-8}
taxi	Visual-TCAV vs Grad-CAM	10	0	1.95×10^{-3}
taxi	Visual-TCAV vs IG	12	0	4.88×10^{-4}
Overall	Visual-TCAV vs Grad-CAM	34	2	1.94×10^{-8}
Overall	Visual-TCAV vs IG	39	0	3.64×10^{-12}

G Experiments on Visual-TCAV stability

In this section, we experiment on the stability of Visual-TCAV. Specifically, we are interested in evaluating (i) the stability of Visual-TCAV with respect to the number of concept example images used for learning the pooled-CAVs, and (ii) the robustness of concepts when perturbed with random images. To address these points, we perform two separate experiments inspired by the procedure used by Martin & Weller (2019) to assess the stability of the *difference of means* and the TCAV Score. The experiments are all performed using the final layer of the same ResNet50V2 model, ensuring comparability of the results. Regarding the example images, we select two classes, "zebra" and "church", and perform an evaluation on the "striped" and "steeple" concepts, respectively.

For the first experiment, starting from a concept defined through an arbitrary amount of training images (specifically, 120 images for "striped" and 75 images for "steeple"), we evaluate both the similarity of pooled-CAVs and the concept attribution when using a lower number of example images. To do this, we apply random sampling without replacement to the original concept images, and we apply Visual-TCAV to derive an updated pooled-CAV and the global concept attribution across the same 50 test images. We computed these metrics considering four different subsets of the concepts over 50 random runs, ultimately resulting in the plots of Fig. 16. From these plots, we observe that when considering both a texture-like concept, such as "striped", or a less homogeneous concept, such as "steeple", utilizing 30 example images instead of the original amount leads to a contained error for Visual-TCAV's concept attribution (i.e., with a mean absolute error < 0.01 for both concepts). At the same time, the pooled-CAVs obtained with 30 concept images present a cosine similarity very close to the original. In general, these experiments suggest that 30-50 example images for defining a concept could be a reasonable amount. However, future studies on a large variety of concepts and networks are still necessary to draw definitive conclusions. Notably, in these experiments, even just 5 images were able to provide a rough estimate, even though the risk of misrepresenting the concept is slightly higher.

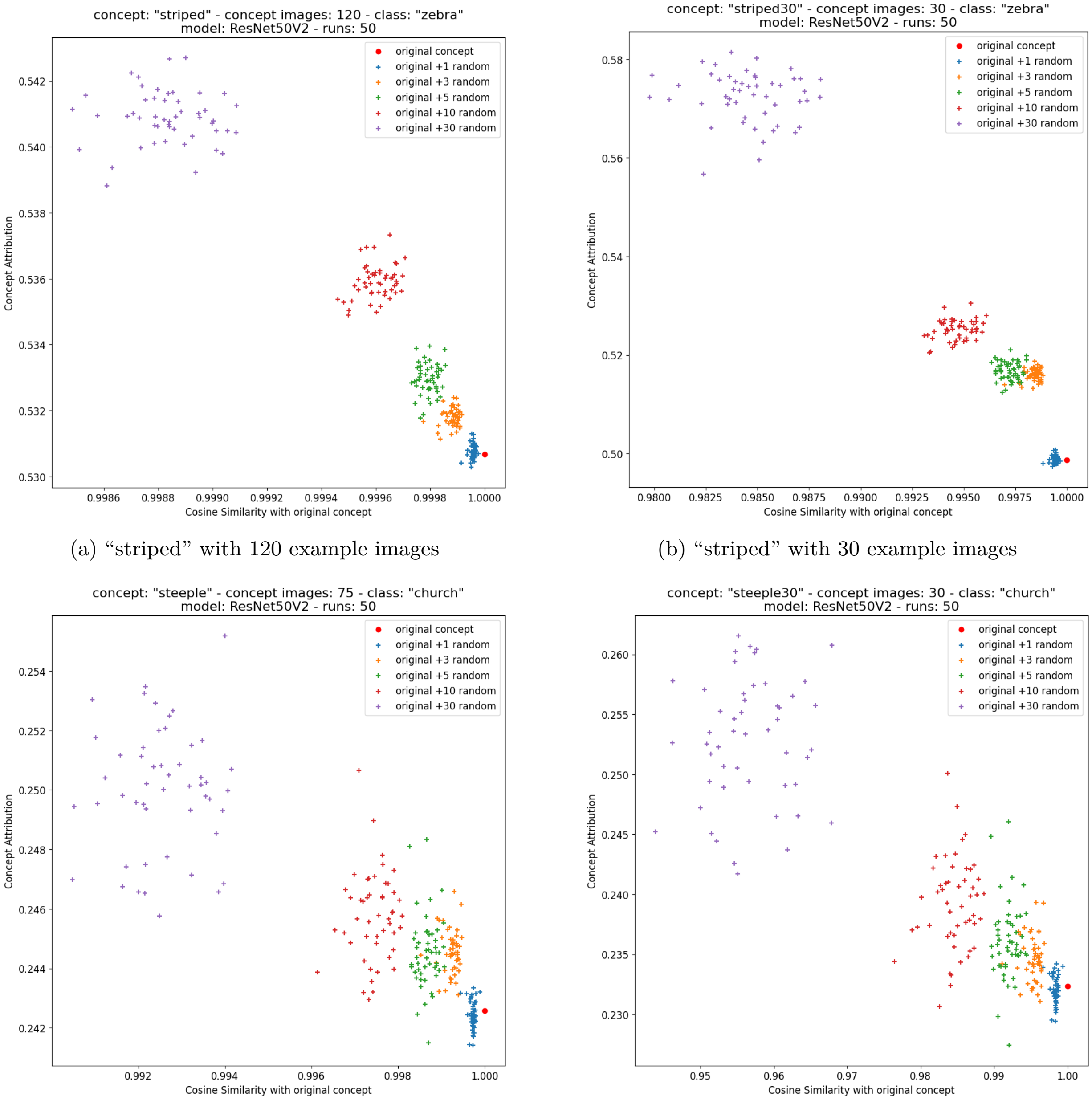


(a) “striped” concept

(b) “steeple” concept

Figure 16: Analysis of results for “striped” and “steeple” concepts computed with a different number of example images. Each point represents the concept attribution and pooled-CAV obtained by removing n example images from a concept set.

We perform a similar experiment to test Visual-TCAV’s stability when the example images contain noise (i.e., random images not correlated with the concept). The purpose is to assess how careful an analyst should be when collecting example images. To this end, we apply random sampling without replacement to select a subset of images from the “random” concept and inject them into the training images. By deriving the updated pooled-CAVs, the global concept attribution, and by repeating this procedure for an increasing number of noisy images, we can compare the deviation from the initial pooled-CAV representing the concept, and the variation of Visual-TCAV’s concept attribution score. We perform this test both considering all the available example images and also with just 30 images. The results are shown in Figs. 17 and 18. From these plots, we observe that adding a few random images to the concept examples does not change the attribution significantly. The worst case is when adding 30 random images to 30 example images, in which we see a slight increase in attribution. This can be explained by the fact that we are also considering the attribution of the background, which also includes some noise accumulated by the integrated gradients. Another effect of adding many random images is that the upper bound to activate the concept map also decreases slightly, as it depends on the concept examples. Overall, from these experiments, we can conclude that results from our method are meaningful even when adding a significant amount of noise (up to 100%) to the example images. In Fig. 19, we also provide examples of concept maps computed with different amounts of example images and with added noise.



(a) “striped” with 120 example images

(b) “striped” with 30 example images

(c) “steeple” with 75 example images

(d) “steeple” with 30 example images

Figure 17: Analysis of results for “striped” and “steeple” concepts computed with a different number of example images and with random images added as noise to the concept set. Each point represents the concept attribution and pooled-CAV obtained by adding N random images to the example images of a concept.

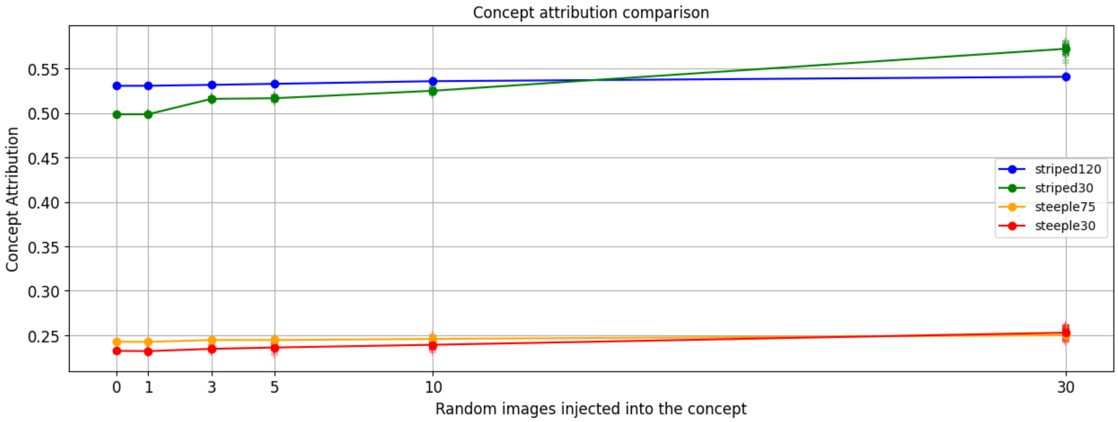
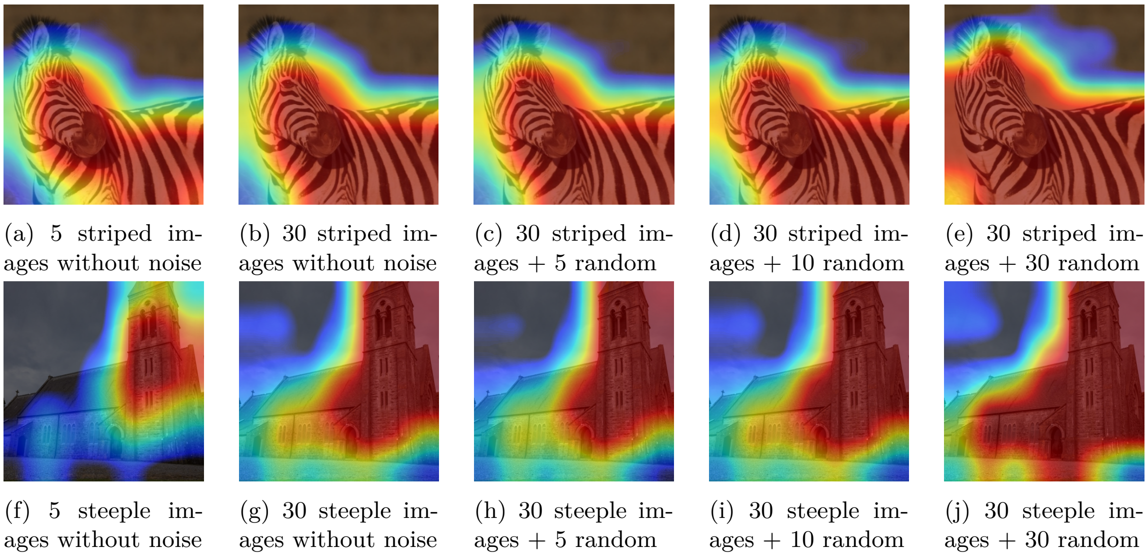


Figure 18: Summary of the concept attribution results for the second experiment. The line plots represent the variation of the average concept attribution when adding N random images to the example images of a concept.



(a) 5 striped images without noise

(b) 30 striped images without noise

(c) 30 striped images + 5 random

(d) 30 striped images + 10 random

(e) 30 striped images + 30 random

(f) 5 steeple images without noise

(g) 30 steeple images without noise

(h) 30 steeple images + 5 random

(i) 30 steeple images + 10 random

(j) 30 steeple images + 30 random

Figure 19: Examples of concept maps computed with different amounts of example images and with different amounts of added noise via random images.

H Ablation study of concept map normalization

We normalize the raw concept map M^c into $\hat{M}^c \in [0,1]$ using the bounded min–max approach in Eq. (2). Since this normalization depends on concept-specific bounds (ℓ_c, u_c) estimated from positive and random (negative) examples, we study how the choice of the summary statistic used to estimate these bounds affects the resulting concept maps, and we contrast it with a simple per-image min-max baseline.

The evaluation is carried out on the controlled tag validation setup from Section 4.4 using the same "100% tags" trained model. We consider the three tags T, C, Z and the last three convolutional layers, whose spatial resolutions are 28×28, 14×14, and 7×7 respectively. For each tag, we use 200 positive test images (containing the tag) and 200 negative test images (same class images without the tag). Ground-truth tag masks are obtained deterministically from the tag generation procedure (using rectangle location and size) and then downsampled to the feature map grid of each layer, so that small discrepancies due only to upsampling do not influence localization scores. We compare four normalization variants. Three of them use the bounded min-max form in Eq. (2) and differ only in how the concept-level bounds (ℓ_c, u_c) are estimated from the concept and random sets. These are the following: *Chmean* (the default in the main paper) uses the contraharmonic-mean statistic, while *Max* and *Mean* replace it with the spatial maximum and spatial mean, respectively. In addition, we include a *Per-image min–max* baseline that normalizes each concept map independently using its own minimum and maximum, without concept-level bounds.

Localization quality on positive images is quantified using *Balanced MSE* (lower is better), which balances errors inside and outside the tag region, and *Soft IoU* (Rahman & Wang, 2016) (higher is better), a continuous, threshold-free version of the Jaccard index computed directly from the normalized concept map and the binary tag mask. To quantify how much and how often the concept map activates when it should not, we also report the *mean activation on negatives* (lower is better), defined as the spatial average of $\hat{M}^c$ on images where the tag is absent. Table 3 summarizes results averaged over the three layers. Results for Z are reported for completeness but should not be used to evaluate normalization, since the "100% tags" model did not reliably learn the Z tag and the corresponding localization scores are near random chance.

Table 3: Concept maps normalization ablation on the synthetic tag experiment. Results are averaged over the last three convolutional layers. Balanced MSE and Negative mean activation are better when lower, while Soft IoU is better when higher. Results for Z are reported for completeness but should not be used to evaluate normalization since the underlying model did not reliably learn the Z tag in this setting.

Tag	Normalization	Balanced MSE↓	Soft IoU↑	Neg. mean act.↓
C	Chmean	0.0449	0.4252	0.0028
	Max	0.0496	**0.4601**	**0.0001**
	Mean	0.1055	0.2459	0.0337
	Per-image min–max	**0.0419**	0.3995	0.4406
T	Chmean	**0.0430**	**0.4460**	0.0012
	Max	0.0584	0.4449	**0.0003**
	Mean	0.0982	0.2743	0.0169
	Per-image min–max	0.0447	0.4139	0.2668
Z	Chmean	0.4892	0.0000	0.0000
	Max	0.4892	0.0000	0.0000
	Mean	0.4892	0.0000	0.0000
	Per-image min–max	0.4904	0.0003	0.0062

On T and C, *Chmean* yields the best (or near-best) Balanced MSE, suggesting concept maps that more closely match the desired "high inside and low outside" behavior. Compared to *Chmean*, the *Mean*-based variant consistently degrades both positive alignment and negative suppression, indicating that simple averaging is too sensitive to widespread low-magnitude activations. *Max* is consistently the most conservative choice on negatives (lowest mean activation), and can slightly improve Soft IoU for C, consistent with producing

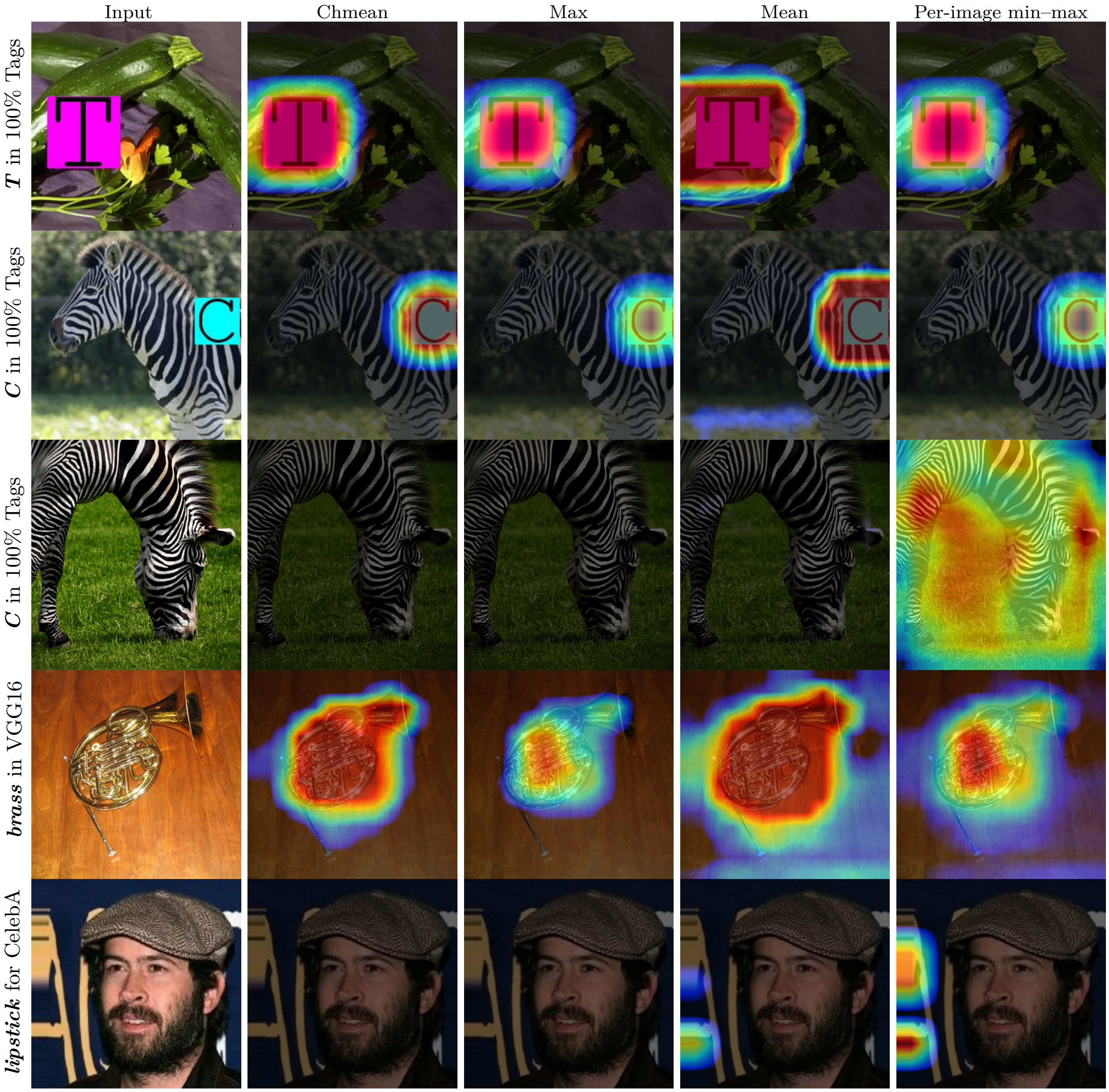


Figure 20: Qualitative comparison of concept map normalizations on synthetic tag examples and ImageNet and CelebA examples. Each row shows the same input image with overlays produced by different normalization strategies given a concept and a CNN model.

sparser maps. As a result, *Max* can be considered a reasonable and somewhat more strict alternative to *Chmean*. By contrast, *Per-image min–max* can appear competitive on positive alignment, but it substantially increases activation on negative images, highlighting the importance of concept-level bounds for meaningful cross-image comparisons. Qualitative examples on both the synthetic tag images and natural images are shown in Fig. 20, where each normalization is applied to the same inputs, and heatmaps are shown using only upscaling with disabled blur to better show the differences.

I IG baseline ablation

Our concept attribution is computed via Integrated Gradients (IG) in the feature maps space, i.e., by integrating gradients of the class logit w.r.t. the feature maps of a given layer. In the main paper, we implement IG using an all-zero feature map as a baseline (which we refer to as *Zero*). Since IG can depend on the baseline, in this appendix, we study how sensitive attributions are to changing such a baseline.

We compare *Zero* against two alternative baselines, which we refer to as *Mean* and *Random*. *Mean* is a dataset-mean baseline that uses, for each layer, the mean feature map computed from a pool of random images at that layer. *Random* is a random-noise baseline in feature maps space. For each test image, we sample Gaussian noise at the same spatial and channel resolution as the layer feature map, with noise scale matched to layer statistics estimated on the same random pool. The noise is centered at *Mean* but still enforcing non-negativity for post-ReLU layers. All results are reported as deviations w.r.t. the *Zero* baseline. The ablation is run using 5 concept-class pairs and the same 3 pretrained architectures (ResNet50V2, InceptionV3, and VGG16) used in the main paper, as well as the same 7 convolutional layers. The evaluated pairs are zebra (*striped*), honeycomb (*honeycombed*), crossword puzzle (*chequered*), waffle iron (*waffled*), and French horn (*brass*), and each pair uses 200 test images. As metrics, we report mean absolute error (MAE; lower is better) and Spearman rank correlation (ρ; higher is better).

Table 4: IG baseline ablation, summary by architecture. Values are deviations from the *Zero* baseline across 5 concept-class pairs and 7 layers for each architecture. Lower MAE is better, higher Spearman ρ is better.

Model	Baseline	MAE ↓	Spearman ρ ↑
InceptionV3	Mean	0.0157 ± 0.0150	0.9802 ± 0.0261
InceptionV3	Random	0.0272 ± 0.0267	0.9689 ± 0.0520
ResNet50V2	Mean	0.0168 ± 0.0205	0.9882 ± 0.0101
ResNet50V2	Random	0.0091 ± 0.0084	0.9795 ± 0.0436
VGG16	Mean	0.0090 ± 0.0098	0.9870 ± 0.0117
VGG16	Random	0.0233 ± 0.0293	0.9767 ± 0.0243

As shown in Table 4, both baselines show low MAE and high Spearman ρ w.r.t. *Zero* across all architectures, indicating that Visual-TCAV attributions are largely robust to reasonable IG baseline choices. The dataset-mean baseline (*Mean*) is consistently close to *Zero* and exhibits stable rank agreement. The random-noise baseline (*Random*) remains strongly rank-consistent as well, while showing slightly larger absolute deviations and variability in some architectures (notably InceptionV3 and VGG16).

Table 5: Trend by layer depth across architectures and concepts. Values are deviations from *Zero*. Layers are aligned by a common depth index L1-L7 (from early to late). Results are aggregated across models and concept-class pairs. Lower MAE is better, higher Spearman ρ is better.

	Mean		Random	
Layer	MAE ↓	Spearman ρ ↑	MAE ↓	Spearman ρ ↑
L1	0.0046 ± 0.0046	0.9872 ± 0.0089	0.0079 ± 0.0066	0.9792 ± 0.0151
L2	0.0063 ± 0.0074	0.9850 ± 0.0150	0.0091 ± 0.0096	0.9719 ± 0.0604
L3	0.0060 ± 0.0056	0.9891 ± 0.0102	0.0089 ± 0.0089	0.9833 ± 0.0257
L4	0.0080 ± 0.0053	0.9797 ± 0.0208	0.0178 ± 0.0166	0.9703 ± 0.0308
L5	0.0211 ± 0.0234	0.9887 ± 0.0060	0.0187 ± 0.0143	0.9858 ± 0.0105
L6	0.0232 ± 0.0177	0.9803 ± 0.0322	0.0298 ± 0.0295	0.9640 ± 0.0730
L7	0.0273 ± 0.0172	0.9860 ± 0.0179	0.0469 ± 0.0395	0.9708 ± 0.0380

Table 5 shows low MAE and high Spearman ρ in all layers, with MAE that tends to increase slightly in deeper layers (L5-L7), while rank agreement remains generally high for both baselines. This is consistent with later layers typically producing larger-magnitude concept attributions, which can translate into a larger absolute MAE. The random-noise *Random* baseline is slightly less stable as it shows larger deviations than *Mean* in the deepest layers (notably L6-L7); however, it still largely preserves the overall ordering of images by concept attribution (high Spearman ρ).

J IG steps ablation

Our concept attribution is computed via IG in the feature maps space, i.e., by integrating gradients of the class logit w.r.t. the feature maps of a given layer. IG is approximated numerically using a finite number of integration steps, and the original IG paper suggests using between 20 and 300 steps (Sundararajan et al., 2017). In the main paper, we use 300 steps as a conservative choice for maximum fidelity; however, this also comes with a considerable computational cost. In this appendix, we study how sensitive attributions are to reducing the number of steps, comparing concept attributions between 20, 50, 100, and 200 steps against the 300 steps reference, reporting results as deviations w.r.t. 300 steps. The ablation is run using the same concept-class pairs, architectures, and layers used for the ablation baseline (Appendix I). Each pair uses 200 test images. As metrics, we report mean absolute error (MAE; lower is better) and Spearman rank correlation (ρ; higher is better).

Table 6: Summary by architecture of the IG steps ablation experiments. Values are deviations from the 300 steps setup and averaged across 5 concept-class pairs and 7 layers for each architecture, with runtime reported as average per concept-class pair. Lower MAE is better, higher Spearman ρ is better. For comparison, the runtime at 300 steps for one concept-class pair is the following: ResNet50V2 = 6:32, InceptionV3 = 23:43, VGG16 = 16:04 (min:s).

Model	Steps	MAE ↓	Spearman ρ ↑	Time (min:s) ↓
InceptionV3	20	0.0002 ± 0.0002	0.9997 ± 0.0004	3:27
InceptionV3	50	0.0001 ± 0.0001	0.9999 ± 0.0001	5:10
InceptionV3	100	0.0000 ± 0.0000	1.0000 ± 0.0000	10:52
InceptionV3	200	0.0000 ± 0.0000	1.0000 ± 0.0000	16:25
ResNet50V2	20	0.0003 ± 0.0003	0.9999 ± 0.0001	2:16
ResNet50V2	50	0.0001 ± 0.0001	0.9999 ± 0.0001	2:41
ResNet50V2	100	0.0000 ± 0.0000	1.0000 ± 0.0000	3:16
ResNet50V2	200	0.0000 ± 0.0000	1.0000 ± 0.0000	4:21
VGG16	20	0.0019 ± 0.0016	0.9985 ± 0.0013	2:28
VGG16	50	0.0006 ± 0.0006	0.9996 ± 0.0004	3:32
VGG16	100	0.0003 ± 0.0002	0.9999 ± 0.0002	5:13
VGG16	200	0.0001 ± 0.0001	1.0000 ± 0.0001	8:18

As shown in Table 6, reducing IG steps produces extremely small absolute deviations and near-perfect rank agreement w.r.t. the 300 steps reference across all architectures. Table 7 shows that stability is also consistent across layer depth. Even at 20 Steps, Spearman ρ remains above 0.998 across models, while MAE remains below 0.002 (with the largest deviation observed for VGG16). At 50 steps, deviations further decrease, and rank agreement is essentially perfect with significantly less runtime.

Table 7: Trend by layer across architectures and concept-class pairs. Values are deviations from the 300 steps reference. Layers are aligned by a common depth index L1-L7 (from early to late). Lower MAE is better, higher Spearman ρ is better.

	20 Steps		50 Steps		100 Steps		200 Steps	
Layer	MAE ↓	Spearman ρ ↑	MAE ↓	Spearman ρ ↑	MAE ↓	Spearman ρ ↑	MAE ↓	Spearman ρ ↑
L1	0.0008 ± 0.0010	0.9989 ± 0.0010	0.0003 ± 0.0004	0.9998 ± 0.0003	0.0001 ± 0.0002	0.9999 ± 0.0001	0.0000 ± 0.0000	1.0000 ± 0.0000
L2	0.0009 ± 0.0012	0.9988 ± 0.0017	0.0003 ± 0.0004	0.9997 ± 0.0004	0.0001 ± 0.0002	0.9999 ± 0.0001	0.0000 ± 0.0000	1.0000 ± 0.0000
L3	0.0007 ± 0.0009	0.9993 ± 0.0008	0.0002 ± 0.0002	0.9999 ± 0.0001	0.0001 ± 0.0001	0.9999 ± 0.0000	0.0000 ± 0.0000	1.0000 ± 0.0000
L4	0.0011 ± 0.0015	0.9993 ± 0.0008	0.0004 ± 0.0005	0.9998 ± 0.0002	0.0002 ± 0.0002	0.9999 ± 0.0001	0.0000 ± 0.0000	1.0000 ± 0.0000
L5	0.0006 ± 0.0008	0.9997 ± 0.0004	0.0002 ± 0.0002	0.9999 ± 0.0001	0.0001 ± 0.0001	0.9999 ± 0.0001	0.0000 ± 0.0000	1.0000 ± 0.0001
L6	0.0007 ± 0.0006	0.9998 ± 0.0002	0.0002 ± 0.0002	0.9999 ± 0.0001	0.0001 ± 0.0001	1.0000 ± 0.0000	0.0000 ± 0.0000	1.0000 ± 0.0000
L7	0.0008 ± 0.0009	0.9998 ± 0.0003	0.0003 ± 0.0003	0.9999 ± 0.0001	0.0001 ± 0.0001	1.0000 ± 0.0001	0.0000 ± 0.0000	1.0000 ± 0.0000

K C-Insertion and C-Deletion faithfulness experiment

This appendix provides a faithfulness experiment on larger models by intervening directly in feature-map space under the C-Insertion and C-Deletion frameworks. Given a concept direction, we progressively remove and re-insert its aligned component from the feature maps and measure the effect on the target class logit. We evaluate on all concept-class pairs analyzed in Figs. 5 and 12, for a total of 49 pairs, using ResNet50V2, InceptionV3, and VGG16 pre-trained on ImageNet, and the ResNet50V2 model trained on CelebA for gender classification. For each pair, we use the last layer and 200 test images.

For a given image, let $\mathbf{F}$ be the feature maps at the chosen layer, and let $\boldsymbol{p}^c$ be the pooled-CAV for concept c (as defined in Section 3). At each spatial location (i, j) we view $\mathbf{F}_{i,j,:}$ as a vector and define the following projection coefficient:

$$\alpha_{i,j} = \frac{\mathbf{F}_{i,j,:}^{\top} \boldsymbol{p}^c}{\|\boldsymbol{p}^c\|^2} \tag{7}$$

Let $\mathcal{D}^c_{\text{neg}}$ denote the set of negative (random) images associated with concept c (i.e., the negative examples used when learning the CAV). We define β as a baseline obtained by averaging $\alpha_{i,j}$ over all spatial locations and over all images in $\mathcal{D}^c_{\text{neg}}$. We then define the concept-aligned component at location (i, j) as:

$$\mathbf{D}^c_{i,j,:} = ReLU\big(\alpha_{i,j} - \beta\big)\, \boldsymbol{p}^c \qquad \forall i, j \tag{8}$$

so that only the excess positive alignment with $\boldsymbol{p}^c$ relative to the negatives is removed.

We parameterize the intervention by $t \in [0, 1]$ and evaluate it on the uniform grid $t \in \{0, 0.1, \ldots, 1\}$, using a step size of 0.1. The C-Deletion and C-Insertion paths in the feature maps space are:

$$\mathbf{F}^{\text{del}}(t) = \mathbf{F} - t\, \mathbf{D}^c, \qquad \mathbf{F}^{\text{ins}}(t) = \mathbf{F} - (1 - t)\, \mathbf{D}^c. \tag{9}$$

C-Deletion starts from the original representation (t=0) and ends at a representation where the concept-aligned component is removed (t=1). C-Insertion starts from the removed representation (t=0) and returns to the original representation (t=1). For each t, we run a forward pass while replacing the feature maps with $\mathbf{F}^{\text{del}}(t)$ or $\mathbf{F}^{\text{ins}}(t)$, and record the target class logit, denoted by $z_k^{\text{del}}(t)$ and $z_k^{\text{ins}}(t)$. For post-ReLU layers, we apply a ReLU to the modified feature maps at each step. Since $\mathbf{F}^{\text{ins}}(t) = \mathbf{F}^{\text{del}}(1 - t)$, the C-Insertion trajectory is mathematically equivalent to the C-Deletion trajectory traversed in reverse. We still report C-Insertion curves for completeness w.r.t. the standard insertion and deletion terminology.

To summarize and aggregate data on each curve, we report an area under the curve (AUC) computed with the trapezoidal rule. For deletion, we integrate the logit curve directly, while for insertion, we integrate the logit gain relative to the removed baseline (t=0):

$$\text{AUC}_{\text{del}} = \int_0^1 z_k^{\text{del}}(t)\, dt, \qquad \text{AUC}_{\text{ins}} = \int_0^1 \big(z_k^{\text{ins}}(t) - z_k^{\text{ins}}(0)\big)\, dt. \tag{10}$$

Intuitively, a concept with a stronger positive influence on class k should produce a lower AUC_{del} (faster deletion) and a higher AUC_{ins} (faster recovery). For reporting, we average AUCs across the 200 images of

Table 8: C-Deletion and C-Insertion summary by attribution rank. Values are aggregated over all evaluated concept-class pairs (200 images per pair). Lower $\mathrm{AUC}_{\mathrm{del}}$ and higher $\mathrm{AUC}_{\mathrm{ins}}$ indicate higher importance. We report Spearman ρ between attribution rank and $\mathrm{AUC}_{\mathrm{del}}$, computed per class and averaged over classes.

Model	Rank	$\mathrm{AUC}_{\mathrm{del}} \downarrow$	$\mathrm{AUC}_{\mathrm{ins}} \uparrow$	Spearman $\rho \uparrow$
ResNet50V2 (ImageNet)	Rank 1	**14.23**	**4.23**	1.00
	Rank 2	18.37	0.49	
	Rank 3	18.87	0.02	
InceptionV3 (ImageNet)	Rank 1	**11.00**	**2.16**	0.75
	Rank 2	12.10	1.12	
	Rank 3	12.96	0.27	
VGG16 (ImageNet)	Rank 1	**9.68**	**1.50**	1.00
	Rank 2	10.32	0.91	
	Rank 3	10.75	0.46	
ResNet50V2 (CelebA)	Rank 1	**5.87**	**2.23**	1.00
	Rank 2	8.70	-0.07	

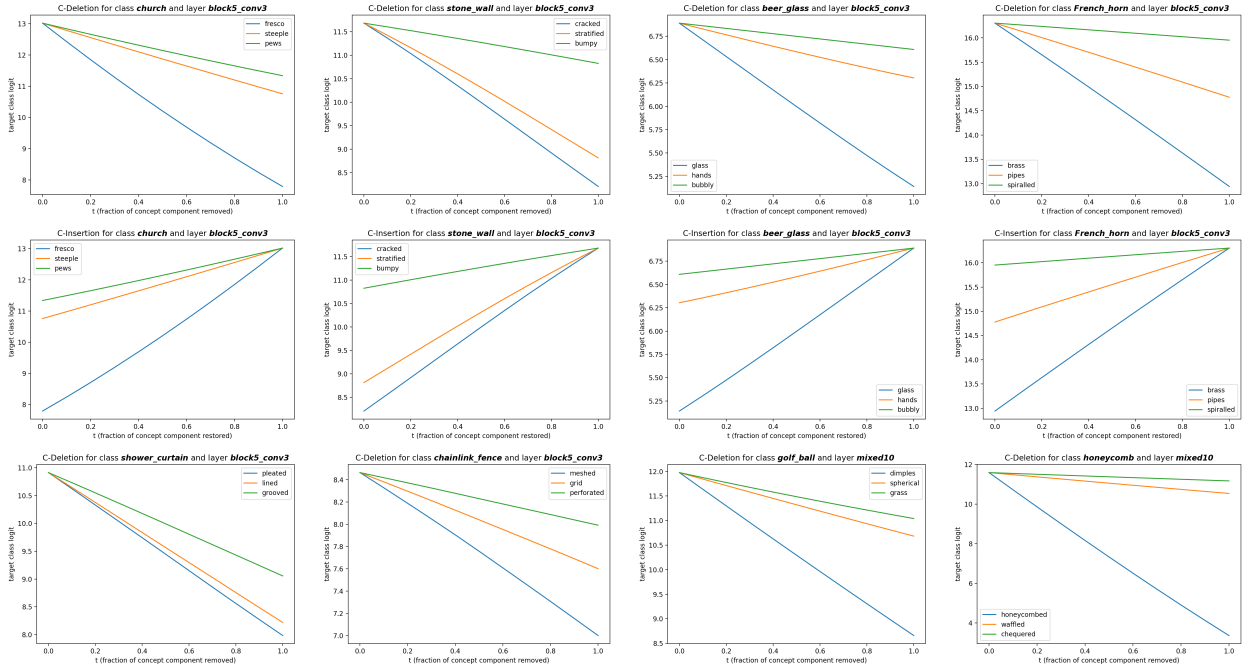
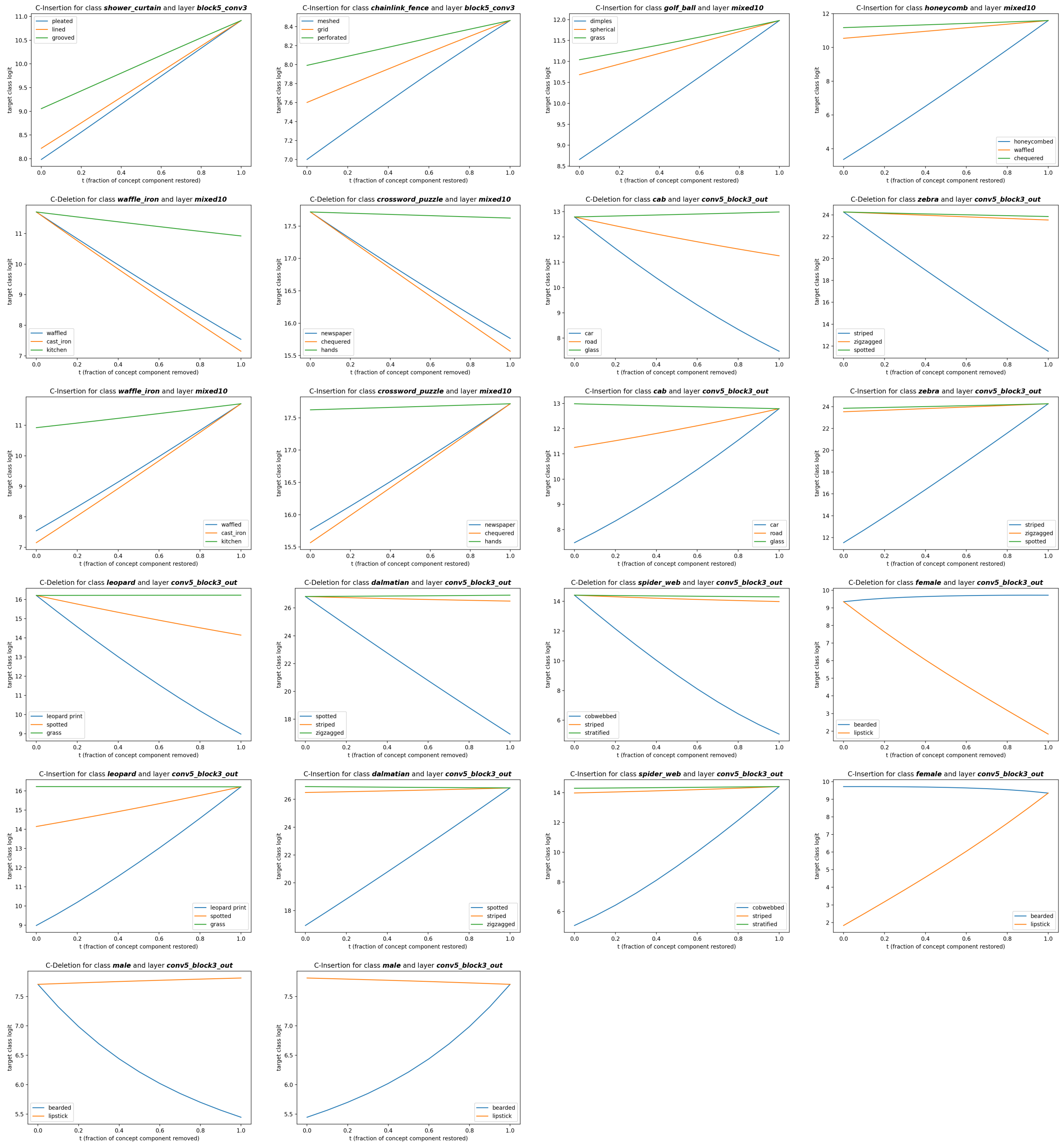


Figure 21: C-Deletion and C-Insertion curves. Each plot shows the mean target class logit over 200 images as a function of the deletion/insertion level t (Part 1).

each pair, and aggregate results by model and concept rank (Rank 1 highest, Rank 3 lowest) using the global attribution ordering from Figs. 5 and 12 at the last layer.



C-Insertion for class ***shower_curtain*** and layer ***block5_conv3***

C-Insertion for class ***chainlink_fence*** and layer ***block5_conv3***

C-Insertion for class ***golf_ball*** and layer ***mixed10***

C-Insertion for class ***honeycomb*** and layer ***mixed10***

C-Deletion for class ***waffle_iron*** and layer ***mixed10***

C-Deletion for class ***crossword_puzzle*** and layer ***mixed10***

C-Deletion for class ***cab*** and layer ***conv5_block3_out***

C-Deletion for class ***zebra*** and layer ***conv5_block3_out***

C-Insertion for class ***waffle_iron*** and layer ***mixed10***

C-Insertion for class ***crossword_puzzle*** and layer ***mixed10***

C-Insertion for class ***cab*** and layer ***conv5_block3_out***

C-Insertion for class ***zebra*** and layer ***conv5_block3_out***

C-Deletion for class ***leopard*** and layer ***conv5_block3_out***

C-Deletion for class ***dalmatian*** and layer ***conv5_block3_out***

C-Deletion for class ***spider_web*** and layer ***conv5_block3_out***

C-Deletion for class ***female*** and layer ***conv5_block3_out***

C-Insertion for class ***leopard*** and layer ***conv5_block3_out***

C-Insertion for class ***dalmatian*** and layer ***conv5_block3_out***

C-Insertion for class ***spider_web*** and layer ***conv5_block3_out***

C-Insertion for class ***female*** and layer ***conv5_block3_out***

C-Deletion for class ***male*** and layer ***conv5_block3_out***

C-Insertion for class ***male*** and layer ***conv5_block3_out***

Figure 22: C-Deletion and C-Insertion curves. Each plot shows the mean target class logit over 200 images as a function of the deletion/insertion level t (Part 2).