Visual Concept Connectome (VCC): Open World Concept Discovery and their Interlayer Connections in Deep Models

Supplementary Material

7. Introduction

This document provides additional material that is supplemental to our main submission. Section 8 outlines the algorithms used in our technical approach. Section 9 describes additional implementation details for our approach, including further Visual Concept Connectome (VCC) generation settings, model details, clustering details and target classes chosen for evaluation. Section 10 provides additional empirical results in terms of validation of the segment proposal method, validation of the concept discovery method, validation of the interlayer testing with concept activation vector (ITCAV) method, VCC visualizations comparing models, classes, and layers, as well as VCCs with a larger number of layers, including all layers. Section 12 discusses the limitations of VCCs and our associated methodology to generate them. Section 13 discusses the societal implications, both positive and negative, of our method. Finally, Section 14 details the used assets and accompanying licenses.

8. Algorithms

In this section, we present pseudocode for the three main algorithmic components of our method: (i) Top-down feature segmentation (Sec. 3.1 in the main paper), (ii) Layer-wise concept discovery (Sec. 3.2 in the main paper) and (iii) Interlayer testing with concept activation vectors (ITCAV) (Sec. 3.3 in the main paper). The top-down feature segmentation method is shown in Algorithm 1. The layer-wise concept discovery method is shown in Algorithm 2. Finally, The ITCAV method is shown in Algorithm 3. All references to equations in the algorithms refer to equations in the main paper.

Algorithm 1 Top-Down Feature Segmentation

Input: Model F, Set of images $\boldsymbol{\mathcal{I}}$, n selected layers of F to study, Spatial clustering algorithm C^{seg}
/C^{seg} is instatiated in terms of maskSLIC [28]*/*
Output: Set of RGB image masks $\mathbf{M}$
*/*Set lists for collection of masks and activations*/*
$\mathbf{M} \leftarrow [],$
for $j \in n$ **do**
  $\mathbf{M}_j, \mathbf{Z}_j \leftarrow [], []$
end for
*/*Collect activations at each layer, Eq. (1)*/*
for $\mathcal{I}^i \in \boldsymbol{\mathcal{I}}$ **do**
  for $j \in n$ **do**
    $\mathbf{z}_j^i \leftarrow f_j(\mathcal{I}^i)$
    $\mathbf{Z}_j$.append($\mathbf{z}_j^i$)
  end for
end for
*/*Iterate through all images*/*
for $i \in |\boldsymbol{\mathcal{I}}|$ **do**
  */*Iterate through layers top-down*/*
  for $j \in \{n, \dots, 1\}$ **do**
    $\mathbf{B}_j^i \leftarrow []$
    */*All features considered at top layer*/*
    if $j == n$ **then**
      $\widetilde{\mathbf{B}}_{j+1}^i(\mathbf{p}; 1) \leftarrow \{1\}^{h_j \times w_j}$
      $\Gamma_j^i \leftarrow \text{silhouette}(\mathbf{z}_j^i(\mathbf{p}) \odot \widetilde{\mathbf{B}}_{j+1}^i(\mathbf{p}; 1))$
      $\{\mathbf{B}_j^i(\mathbf{p}; \gamma)\}_{\gamma}^{\Gamma_j^i} \leftarrow \mathsf{C}_{\Gamma_j^i}^{seg}(\mathbf{z}_j^i(\mathbf{p}) \odot \widetilde{\mathbf{B}}_{j+1}^i(\mathbf{p}; 1))$
      $\mathbf{B}_j^i$.append($\{\mathbf{B}_j^i(\mathbf{p}; \gamma)\}_{\gamma}^{\Gamma_j^i}$)
    else
      */*Top-down masking for all other layers*/*
      for $\mathbf{B}_{j+1}^i(\mathbf{p}; k) \in \{\mathbf{B}_{j+1}^i\}_{\gamma}^{\Gamma_{j+1}^i}$ **do**
        */*Bilinear interpolate mask to feature shape*/*
        $\widetilde{\mathbf{B}}_{j+1}^i(\mathbf{p}; k) \leftarrow \text{BiInterp}(\mathbf{B}_{j+1}^i(\mathbf{p}; k), \mathbf{z}_j^i.\text{shape})$
        */*Mask with higher layer binary mask, Eq. (2)*/*
        $\Gamma_j^i \leftarrow \text{silhouette}(\mathbf{z}_j^i(\mathbf{p}) \odot \widetilde{\mathbf{B}}_{j+1}^i(\mathbf{p}; k))$
        $\{\mathbf{B}_j^i(\mathbf{p}; \gamma)\}_{\gamma}^{\Gamma_j^i} \leftarrow \mathsf{C}_{\Gamma_j^i}^{seg}(\mathbf{z}_j^i(\mathbf{p}) \odot \widetilde{\mathbf{B}}_{j+1}^i(\mathbf{p}; k))$
        $\mathbf{B}_j^i$.append($\{\mathbf{B}_j^i(\mathbf{p}; \gamma)\}_{\gamma}^{\Gamma_j^i}$)
      end for
    end if
    */*Create and save RGB Masks, Eq. (3)*/*
    for $\mathbf{B}_j^i(\mathbf{p}; \gamma) \in \mathbf{B}_j^i$ **do**
      $\mathbf{M}_j^i(\mathbf{p}; \gamma) \leftarrow \mathcal{I}^i \odot \mathbf{B}_j^i(\mathbf{p}; \gamma)$
      $\mathbf{M}_j$.append($\mathbf{M}_j^i(\mathbf{p}; \gamma)$)
    end for
    $\mathbf{M}$.append($\mathbf{M}_j$)
  end for
end for
Return M

9. Implementation details

VCC settings. 50 target images are used to generate each VCC. The statistical testing and training of the CAVs [29] use 20 unique sets of random images from the Broden dataset [3]. When computing the concept connection edge weight between the final selected layer and the class logit, the standard TCAV [29] score is used.

Model settings. For the CLIP-ResNet50 [42] experiments (Sec. 10.3), we follow the original paper [42] and compute the logit for each ImageNet [15] class using a single query sentence '*a photo of a* $\{class\}$'; where '$\{class\}$' is the target ImageNet class. The layer names used (according to the PyTorch [41] module nomenclature) for each model when generating all four-layer VCCs are as follows:

	ResNet50			VGG16			MViT			ViT-b		
	RF	ACE	Ours	RF	ACE	Ours	RF	ACE	Ours	RF	ACE	Ours
Layer1	43	2.4	4.2	10	2.8	3.5	224	1.7	7.5	224	1.3	10.0
Layer2	99	2.0	11.9	32	2.4	6.8	224	1.0	15.2	224	1.7	24.4
Layer3	211	1.92	23.9	80	2.2	15.5	224	2.5	28.1	224	2.4	35.8
Layer4	435	2.1	47.9	176	2.2	46.8	224	2.4	50.1	224	2.4	54.1

Table 1. Validation of segment proposal component of our method (Sec. 3.1). The relative concept segment size compared to the entire image for a given layer, is shown with the receptive field (RF) width/height of the same layer. We compare our method (Ours) to the baseline method, ACE [23]. For all models, the relative segment size discovered using our method has a stronger correlation with the receptive field size than the concepts discovered using ACE.

Algorithm 2 Concept Discovery

Input: Model F, n selected layers of F to study, Set of RGB Image Masks at each Layer $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_n\}$, Clustering algorithm C^{con}
/C^{con} is instatiated in terms of k-means [35]*/*
Output: Set of concept centroids $\mathbf{Q} = \{\mathbf{Q}_1, ..., \mathbf{Q}_n\}$

```
Q ← []
for j ∈ n do
    /*Cluster segment activations, Eq. (4)*/
    Z_{M_j} ← f_j(M_j)
    Q_j ← C^con(GAP(Z_{M_j}))
    /*Prune clusters (see Sec. 4.1) for details*/
    Q_j ← prune(Q_j)
    Q.append(q_j)
end for
Return Q
```

Algorithm 3 Interlayer Testing with Concept Activation Vectors

Input: Model F, higher layer selected to study l, lower layer selected to study j, Concept Centroid for higher layer $\mathbf{q}_l^{m_l}$, Concept centroid for lower layer $\mathbf{q}_j^{m_j}$, Set of RGB image masks each associated with higher layer concept centroid $\mathbf{M}_{\mathbf{q}_l^{m_l}}$, Set of RGB image masks each associated with lower layer concept centroid $\mathbf{M}_{\mathbf{q}_j^{m_j}}$, Set of random images $\boldsymbol{\mathcal{I}}_{rnd}$, Linear classifier h
Output: Concept connection edge weight between concepts $\mathbf{q}_j^{m_j}$ and $\mathbf{q}_l^{m_l}$: $e_{\mathbf{q}_j^{m_j}, \mathbf{q}_l^{m_l}}$

```
/*Get activations for lower level concept*/
z_{M_{q_j^{m_j}}} ← f_j(M_{q_j^{m_j}})
/*Get activations for random concept*/
z_{I_rnd} ← f_j(I_rnd)
/*Train CAV and get orthogonal vector to hyperplane in direction of lower concept*/
V_{q_j^{m_j}} ← h(z_{M_{q_j^{m_j}}}, z_{I_rnd}).train()
CountPositive ← 0
/*Iterate through higher concept segments*/
for x ∈ M_{q_l^{m_l}} do
    z_j ← f_j(x)
    /*Get gradient of segment at layer l with respect to lower layer j*/
    g_j ← ∇_{f_j} ||f_l(z_j) − q_l^{m_l}||_2
    /*Calculate sensitivity of upper concept to lower concept, Eq. (5)*/
    S_{q_j^{m_j}, q_l^{m_l}} = g_j · V_{q_j^{m_j}}
    if S_{q_j^{m_j}, q_l^{m_l}} > 1 then
        CountPositive = CountPositive + 1
    end if
end for
/*Calculate fraction of positive alignments, Eq. (6)*/
e_{q_j^{m_j}, q_l^{m_l}} = CountPositive / |M_{q_l^{m_l}}|
Return e_{q_j^{m_j}, q_l^{m_l}}
```

1. ResNet18 [25], ResNet50 [25] and CLIP-ResNet50 [42]: *Layer1, Layer2, Layer3, Layer4*
2. VGG16 [48]: *8, 15, 22, 29*
3. MobileNetv3 [44]: *0, 2, 4, 6*
4. MViT [18]: *1, 3, 9, 15*
5. ViT-b [51]: *2, 5, 8, 10*

The layer names used (according to the PyTorch [41] module nomenclature) for each model when generating the all-layer VCCs are as follows:

1. ResNet50 [42]: *layer1.0, layer1.1, layer1.2, layer2.0, layer2.1, layer2.2, layer2.3, layer3.0, layer3.1, layer3.2, layer3.3, layer3.4, layer3.5, layer4.0, layer4.1, layer4.2*
2. VGG16 [48]: *1, 3, 6, 8, 11, 13, 15, 18, 20, 22, 25, 27, 29*
3. MobileNetv3 [44]: *0.0, 1.0, 1.1, 2.0, 2.1, 2.2, 3.0, 3.1, 3.2, 3.3, 4.0, 4.1, 5.0, 5.1, 5.2, 6.0*
4. MViT [18]: *0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15*
5. ViT-b [51]: *0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10*

Clustering details. Following previous work [23], during the concept discovery clustering, C^{con}, we over-cluster and then prune to ensure that fewer concepts will be missed. We follow previous work [23] and choose the number of

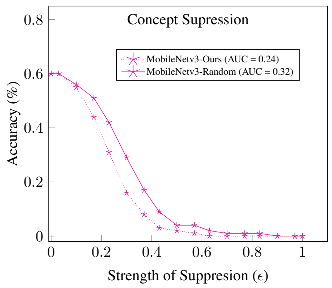

Figure 9. Additional validation results for the concept discovery method (Sec. 3.2 in the main paper) for the MobileNetv3 model [26]. For a set of 50 randomly selected ImageNet classes, we discover concepts in four layers of the model. During inference, one randomly selected concept at each layer is suppressed by a factor of ϵ.

clusters to be $k_m = 25$ in the concept discovery step. However, as VCCs target the discovery of concepts at potentially all layers, we select a different pruning protocol [23], where they prune based on a single minimum value. Instead, we prune clusters that have less than Y images, via the generalized logistic sigmoid

$$Y = A + \frac{K - A}{(C + Qe^{-Bt})^{1/\nu}}, \tag{9}$$

where $A = -102$, $K = 115$, $C = 1$, $Q = 1$, $B = 0.0004$ and $v = 1$. Pruning based on a sigmoid shaped function enables different levels of leniency when considering what constitutes a concept for each layer. This is crucial as different layers contain a different number of segments from the top-down segmentation algorithm (Sec. 3.1 in the main paper).

For the maskSLIC [28] clustering stage, we use a compactness of 0.8 and all other parameters are set to the Scikit-Image [50] defaults. The Euclidean distance is used for all clustering steps.

Randomized testing. When applying our ITCAV method to calculate the strength of connection between two concepts, we protect against the impact of spurious results by performing a statistical significance test on all ITCAV scores. More specifically, instead of simply calculating the ITCAV score with the target concept images, we calculate an additional 20 ITCAV scores using random images sampled from Broden [3]. We perform a two-sided t-test of the ITCAV score based on the 20 random scores. We test whether the null hypothesis (*i.e.* a ITCAV score of 0.5) can be rejected with a p-value of $p > 0.05$. All ITCAV scores shown in the main paper and supplement pass this statistical test, *i.e.* $p \leq 0.05$.

ImageNet classes. The 50 ImageNet classes used for the model and task analysis experiments (Sec. 4.3 in the main paper) are the following: *tow truck, sturgeon, sax, wool, basketball, whiptail, toy poodle, acorn, crutch, church, backpack, spaghetti squash, snowmobile, teapot, ant, chain, gorilla, holster, wreck, ice lolly schipperke, cradle, dowitcher, leopard, oystercatcher, saltshaker, drake, loupe, spotlight, Newfoundland, bagel, electric fan, ping-pong ball, streetcar, knot, plate, sea lion, leafhopper, tusker, punching bag, black widow, traffic light, tricycle, paper towel, guinea pig, castle, go-kart, platypus, badger and bicycle-built-for-two*.

The 10 ImageNet classes used for the all-layer VCC analysis experiments (Sec. 4.3 in the main paper) are the following: *tow truck, sturgeon, sax, wool, basketball, whiptail, toy poodle, acorn, crutch, church*

10. Additional empirical results

10.1. VCC component validation

10.1.1 Segment proposal validation

Table 1 presents additional results to validate our top-down feature segmentation approach (Sec. 3.1). In particular, we show results for four additional models. We observe findings consistent with those in the main paper: Our method produces concepts that increase in size as the information flows deeper through the model. It is interesting to observe a similar phenomenon in transformer-based architectures, *i.e.* MViT [18] and ViT-b [51]. While the relative concept size of the baseline, ACE [23], varies less than 2% across all architectures and layers, the size of concepts produced by our method can differ up to 20% between architectures (*i.e.* comparing VGG16-Ours with ViT-b-Ours) and 40% between layers. This finding is to be expected as it is unlikely for all architectures at all layers to capture concepts of the same size.

10.1.2 Concept validation

Figure 9 presents additional results to validate our layer-wise concept discovery method (Sec. 3.2 in the main paper) for the MobileNetv3 [26] model. The results are consistent with those in the main paper, *i.e.* the accuracy for the target class decreases faster when a concept is suppressed compared to a randomly chosen direction. This result implies that the concepts discovered throughout the model represent meaningful directions in the latent space.

10.1.3 Interlayer concept weight validation

We now extend the validation of our Interlayer Testing with Concept Activation Vectors (ITCAV) method (Sec. 3.3 in the main paper). In particular, we show results for four additional models in Fig. 10 and observe findings consistent with those in the main paper: There is a positive correlation between the average path strength (APS) and the logit sum

	Branching Factor		Number of Concepts		Edge Weight Ave.	
	R50	CLIP	R50	CLIP	R50	CLIP
Layer1	5.484	6.824	10.447	11.085	0.414	0.417
Layer2	4.945	4.141	8.000	7.468	0.554	0.54
Layer3	2.799	2.754	5.106	5.702	0.476	0.563
Layer4	2.915	1.574	2.957	2.383	0.917	0.634

Table 2. VCC metrics for ResNet50 [25] trained on ImageNet [15] and via contrastive image-language pretraining (CLIP) [42].

(LS) score. These results further suggest that the combination of ITCAV scores is predictive of whether a concept is representative of the target class.

10.2. Understanding models

Figure 11 extends the results from Fig. 6 in the main paper and shows a quantitative analysis for all-layer VCCs on two additional models: ResNet50 [25] and ViT-b [16]. Consistent with the results from the main paper, we again see that the branching pattern and number of concepts start at a higher value and converges, suggesting that many concepts are shared between classes at early layers while the later layers capture ImageNet's foreground-background structure. We also observe patterns in the ITCAV values and variances that are consistent with the main paper. The edge weight values are consistent until the final layer at which point they increase, denoting the stronger contribution of the final layers to the output. In terms of the ITCAV variance, we again see that transformers (ViT-b) have a higher variance than CNNs (ResNet50) in the last layer, further suggesting that transformers have greater compositionality of concepts before the final prediction.

10.3. Understanding tasks

We now explore how VCCs can reveal the effect of the training task on learned concepts and their connections. In particular, given the recent advances of image-language training paradigms, we compare the standard ResNet50 [25] model trained on ImageNet [15] with ResNet50 trained via Contrastive Language Image Pretraining (CLIP) [42].

Table 2 compares graph metrics over VCC layers between the two models at four residual blocks. We observe small but notable differences between the two models. First, CLIP contains a higher branching factor and number of concepts in the first layer than ResNet50, suggesting slightly more concepts are discovered and composed at the beginning of the network. The pattern is reversed at the end of the models, where CLIP has a slightly lower number of concepts and branching factor than ResNet50. When considering average edge weight values, we also observe a general consistency across models apart from the final layer, where ResNet50 has a much larger average value. This may be due to ImageNet trained CNNs having less compositionality at the end of the model as we observed both object and background classes having a large impact on the output in the main paper (Sec. 4.3).

10.4. Additional VCCs visualizations

10.4.1 Four layer VCCs

We now supplement the analysis from Sec. 4.3 from the main paper by generating VCCs for the entirety of the five models analyzed for different classes in the four layer setting. We specifically chose these models and layer settings as they are the same as in Sec. 4.3 in the main paper. The models shown are ResNet50 [25] (Fig. 13), VGG16 [48] (Fig. 14), MobileNetv3 [26] (Fig. 15), MViT [18] (Fig. 16) and ViT-b [51] (Fig. 17). All models are trained on ImageNet [15]. The layers selected are the same ones as detailed in Sec. 9.

We observe differences in mid and late layer connection strengths between CNNs and transformers. Similar to the main paper discussion (Sec. 4.3), CNNs (Figs. 13, 14 and 15) show stronger connections with less variance between the 4^{th} layer and class logit than the transformers (Figs. 16 and 17). Additionally, CNNs tend toward concepts which capture either the entire foreground or background in later layers. Meanwhile, the transformers produce concept shapes of varying shapes and sizes, *e.g.* the VCC for ViT-b in Fig. 17 contains concepts of both small patches and the entire images in the final VCC layer and concepts of varying sizes in the first VCC layer. These findings for the transformers are consistent with the ability of such models to form data associations across their input without the locality constraints that are inherent in convolutional models.

Finegrained dataset VCC. To show how VCCs generalize to other datasets, we generate a VCC for the CUB [52] finegrained classification dataset, where the goal is to classify different types of birds. Figure 18 shows a four layer VCC for the ResNet18 [25] model targeting the class "indigo bunting". We again see interesting concepts being composed. For example, branches and the color blue occur in stage1 and stage2, while stage4 bird concepts are composed from branch, background and bird head concepts in stage3.

All layer VCC. We show an additional all-layer VCC in Fig. 19 of the VGG16 model [49] targeting recognition of class "church". As in the visualization in the main paper, we visualize VCC subgraphs and observe interesting compositions occurring at different levels of abstraction corresponding at different depths of the model. At early layers (bottom left), we observe oriented brown patterns and yellow color composing the concept of brown and yellow orientation. Middle layers (right) show the concept of 'church roof with sky in the background' being composed of 'church roof' and 'sky'. The final layer concepts (top left) show that both foreground objects, *e.g.* churches, and background regions, *e.g.* trees or sky, concepts highly influence the final category.

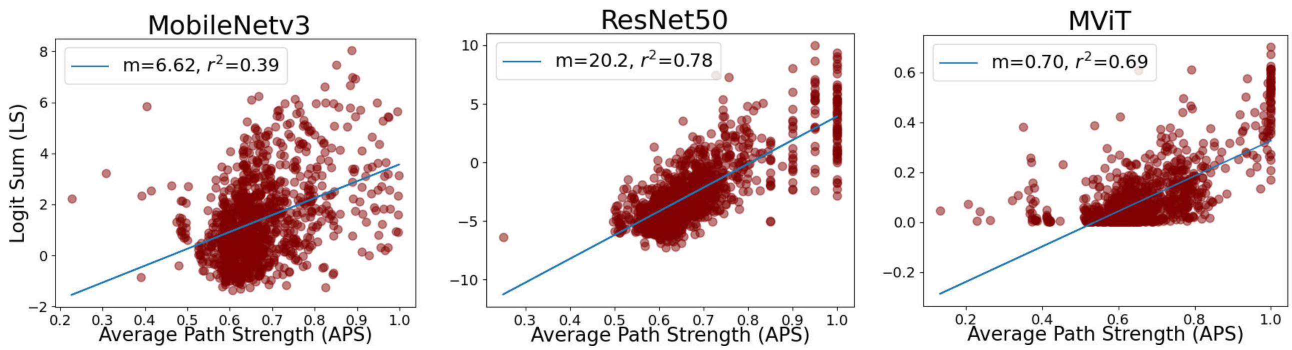

Figure 10. Additional validation results of interlayer concept weights. The unnormalized logit sum (LS) scores, main paper (Eq. 8), for the target class are plotted against the average path strength (APS) scores, main paper (Eq. 7). A positive correlation implies that the ITCAV edge weights connecting a concept to the class are predictive of the model output having a higher probability for that class.

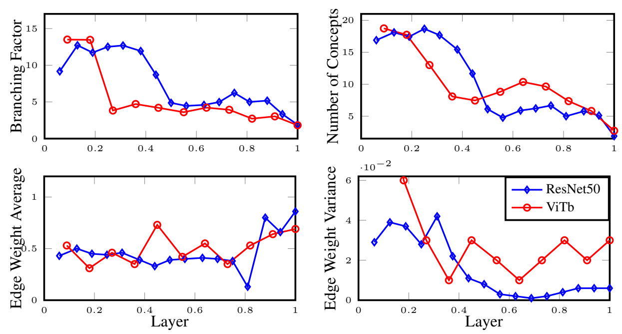

Figure 11. Graph metrics of all layer VCCs for two architectures. Layer number normalized to allow for comparison of models with different numbers of layers.

11. Application: Diagnosing failure predictions

To further show our VCC's practical utility, we show another example of model debugging. Figure 12 shows a 'church' incorrectly predicted as a 'vault' by a ResNet50 model, and the corresponding incorrect VCC ('vault', left) and correct VCC ('church', right). As the image is decomposed using our top-down segmentation (Sec. 3.1), it is revealed that several segments are closer, in terms of the l_2 distance of the pooled segment activations, to concepts in the 'vault' VCC (red outlines) than the 'church' VCC (green outlines). While the model correctly encoded the door as a 'church' concept, the regions outside the door are identified as 'vault' concepts from layers two through four, which may cause the error. We also note the lack of other 'church' specific concepts, such as the sky or cylindrical columns.

12. Limitations

We note some limitations of our method. We rely on the Silhouette method [43] to select the number of clusters (*i.e.* segments) during the top-down feature segmentation stage to automate this step. However, use of a different method for selecting the number of clusters could yield different results and therefore different overall VCCs. In practice, we have found that using the Silhouette method consistently produces meaningful segments; so, sensitivity to this choice is not a serious limitation. Another limitation arises is that we do not provide a method for selecting the set of layers to analyze. Such a method for automatic layer selection could reveal further interesting and useful patterns, such as uncovering the set of layers, along with their connections, which impact the model output most significantly. A direction to realize such an algorithm could be to construct a large VCC and subsequently trim the least important nodes and edges (*e.g.* based on the average path strength (APS) to the logit, as defined in Eq. 7 in the main paper).

13. Societal implications

Understanding the decision making processes of deep networks is an important and open problem in computer vision. Given their potential for negative impacts when deployed, various jurisdictions are moving forward with legislation that may curtail certain applications and mandate interpretable components in deployed systems [14]. VCCs are a step towards a holistic understanding of how concepts in deep networks are learned and in the future may provide a direction to design legally recognized interpretations of these models.

VCCs may have implications in terms of recognizing both *what* and *how* biases are learned by deep networks. While the learning of various biases by deep networks is well documented [37], it is not well understood *how* these biases are constructed and learned by the model. For example, it is not sufficient to explain a model's prediction by saying it uses the background as a feature. It would be more desirable to explain what concepts are composed in earlier layers that lead the model to encode the background feature in the later layers, which we have shown that VCCs can reveal. Moreover, such information could open up new directions for model debiasing.

In terms of negative consequences, VCCs (and explain-

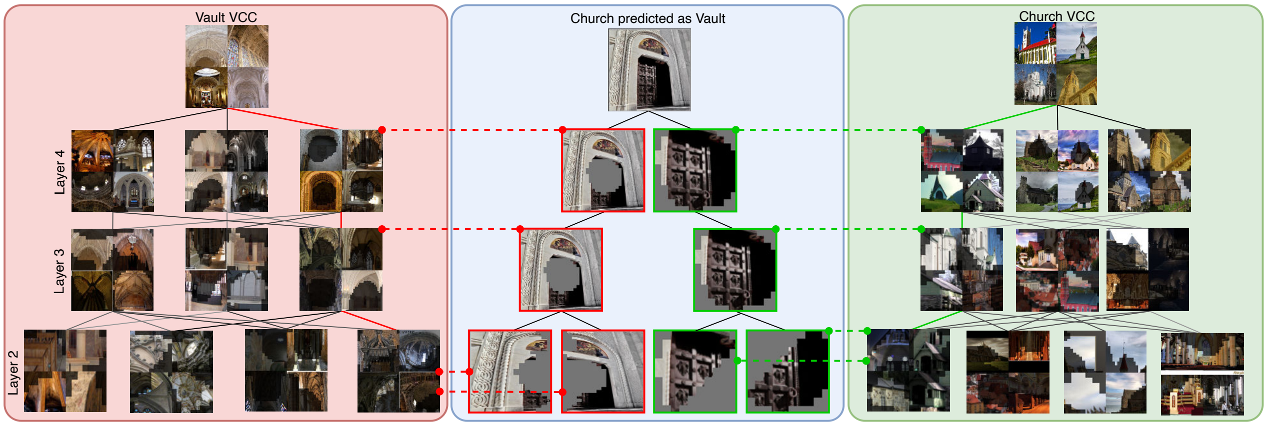

Figure 12. Debugging model failure modes with VCCs. We show an image of a church incorrectly predicted by a ResNet50 as a vault (middle) as well as the top-down segmentation of the image (Sec. 3.1). We also show the incorrect (left) and correct (right) VCCs. Following the hierarchy of concepts reveals that the model incorrectly focused heavily on the cement door frame, starting at Layer 2.

able AI in general) may give users a false sense of security and allow them to deploy models that ultimately do more harm than good. Furthermore, the contribution of additional explainable AI methods may contribute to the disagreement problem [33], *i.e.* when multiple explanations of a given model disagree with each other. It is an open research question on how to resolve such disagreements, when potentially dozens of possible explanations for a given model exist.

14. Assets and licensing

Models. We use provided code and trained weights from the MViT[1] and CLIP[2] repositories. MViT is licensed under the Apache 2.0 license[3] and CLIP is licensed under the MIT license[4].

Datasets. We use the ImageNet dataset[5] which is under the BSD 3-Clause License[6] and the Broden dataset[7] which is under the MIT license[8].

[1] https://github.com/facebookresearch/mvit

[2] https://github.com/openai/CLIP

[3] https://github.com/facebookresearch/mvit/blob/main/LICENSE

[4] https://github.com/openai/CLIP/blob/main/LICENSE

[5] https://www.image-net.org/

[6] https://github.com/floydhub/imagenet/blob/master/LICENSE

[7] https://github.com/CSAILVision/NetDissect-Lite

[8] https://github.com/davidbau/quick-netdissect/blob/master/LICENSE

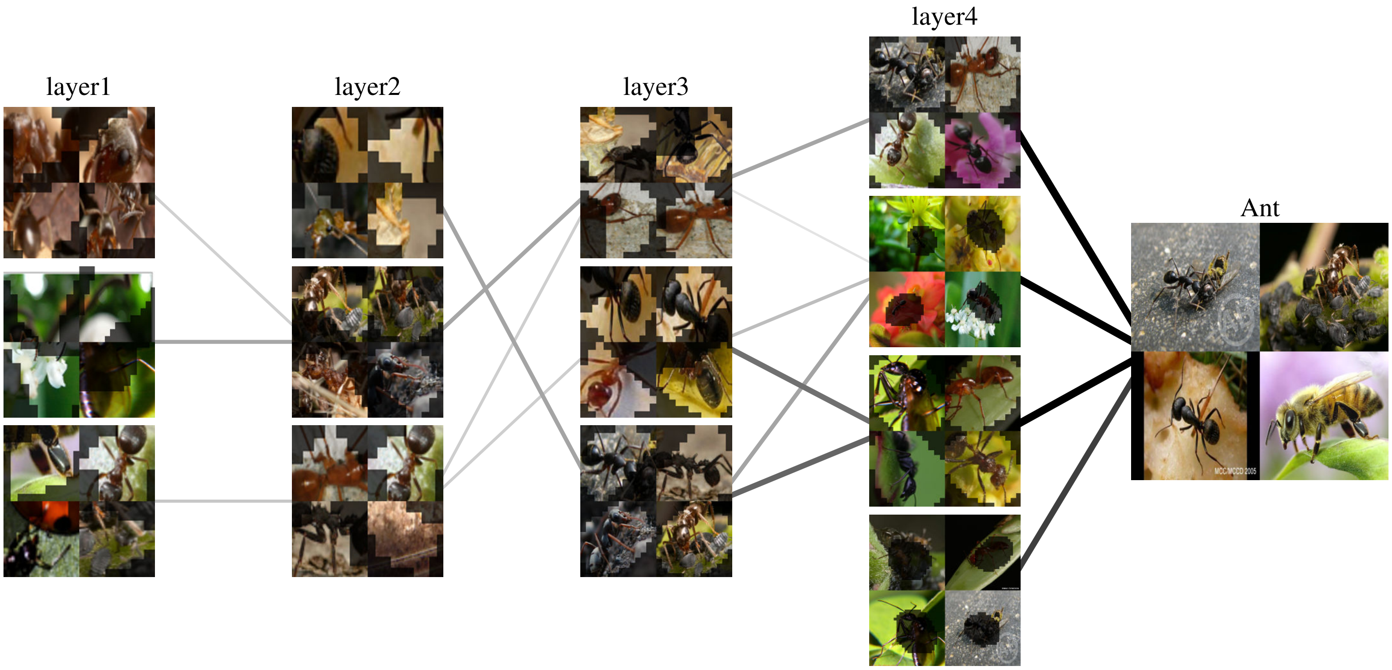

Figure 13. A VCC for four layers of a ResNet50 [25] model targeting the class “ant”. Darker lines denote larger concept contributions.

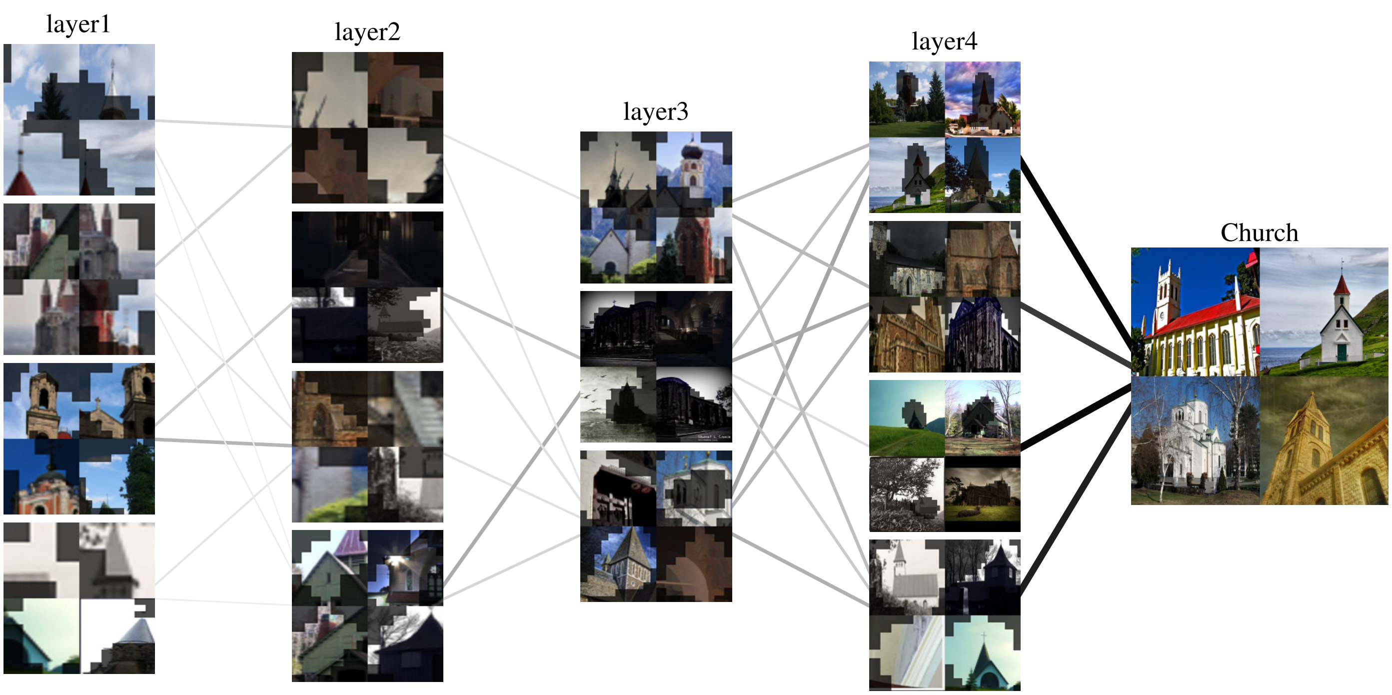

Figure 14. A VCC for four layers of a VGG16 [49] model targeting the class “church”. Darker lines denote larger concept contributions.

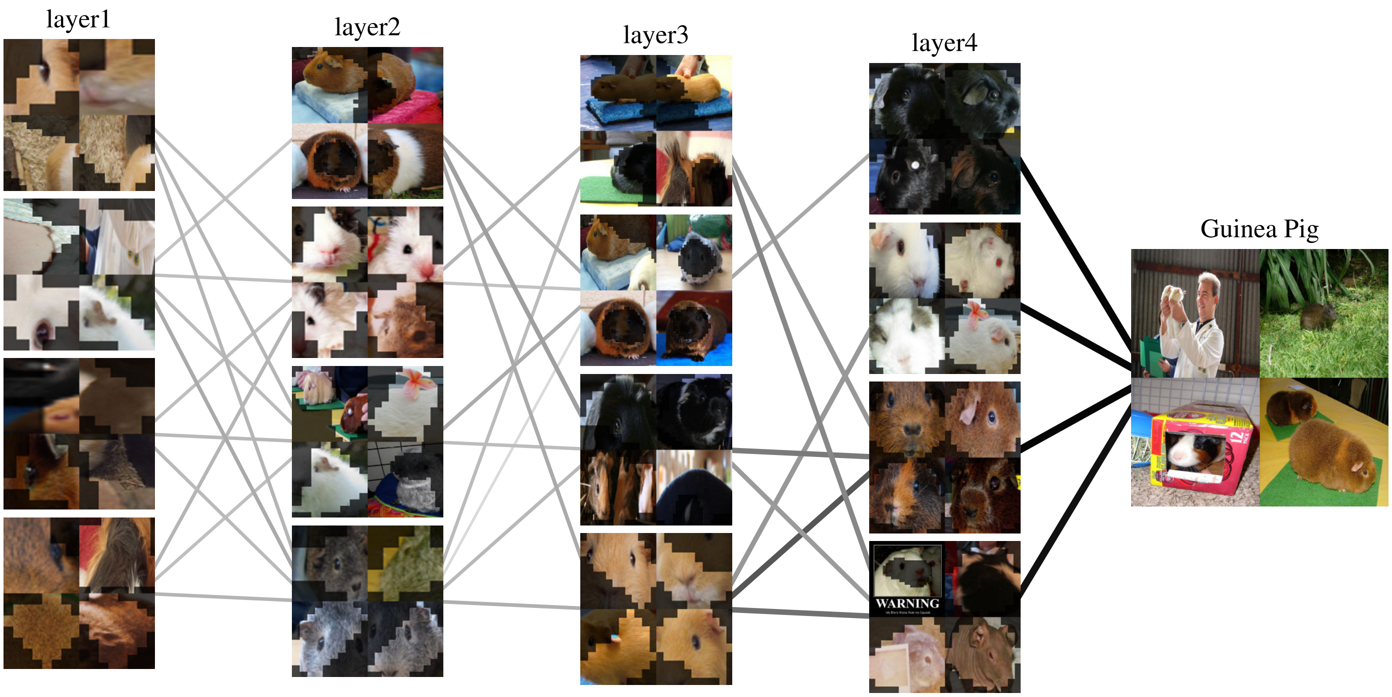

Figure 15. A VCC for four layers of a MobileNetv3 [26] model targeting the class “guinea pig”. Darker lines denote larger concept contributions.

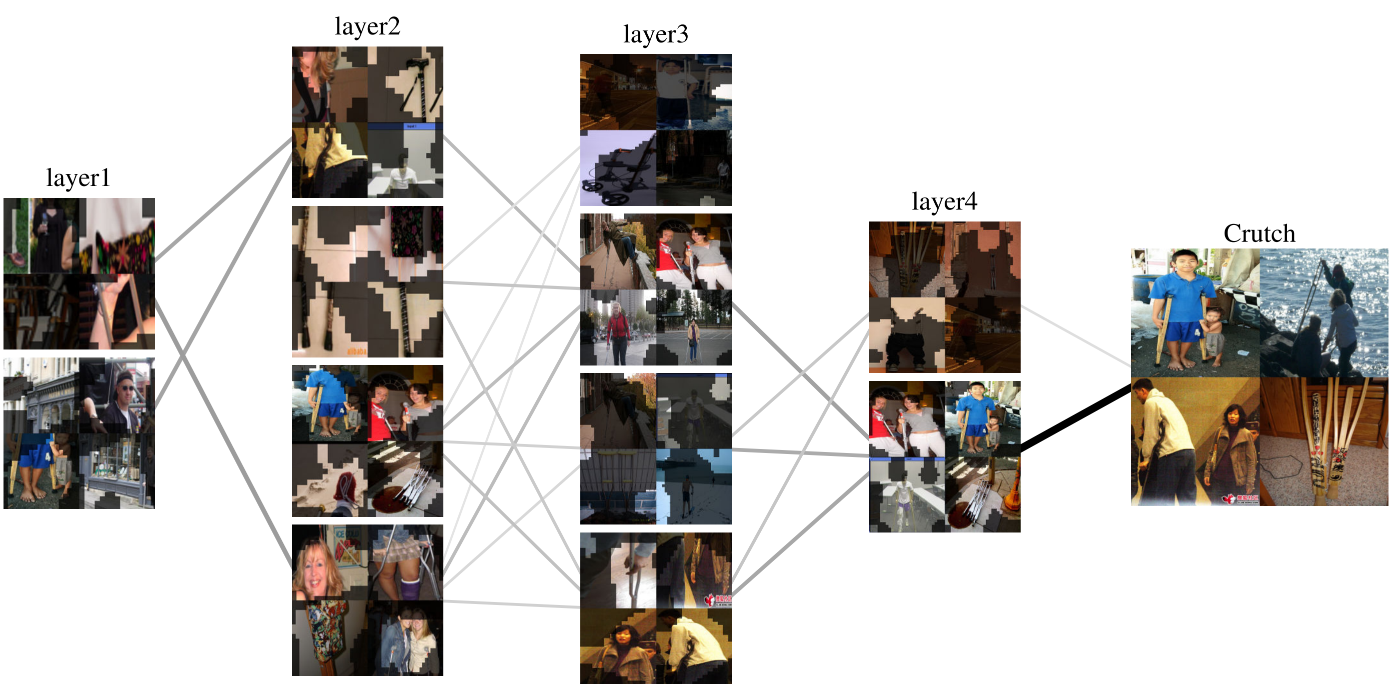

Figure 16. A VCC for four layers of a MViT [18] model targeting the class “crutch”. Darker lines denote larger concept contributions.

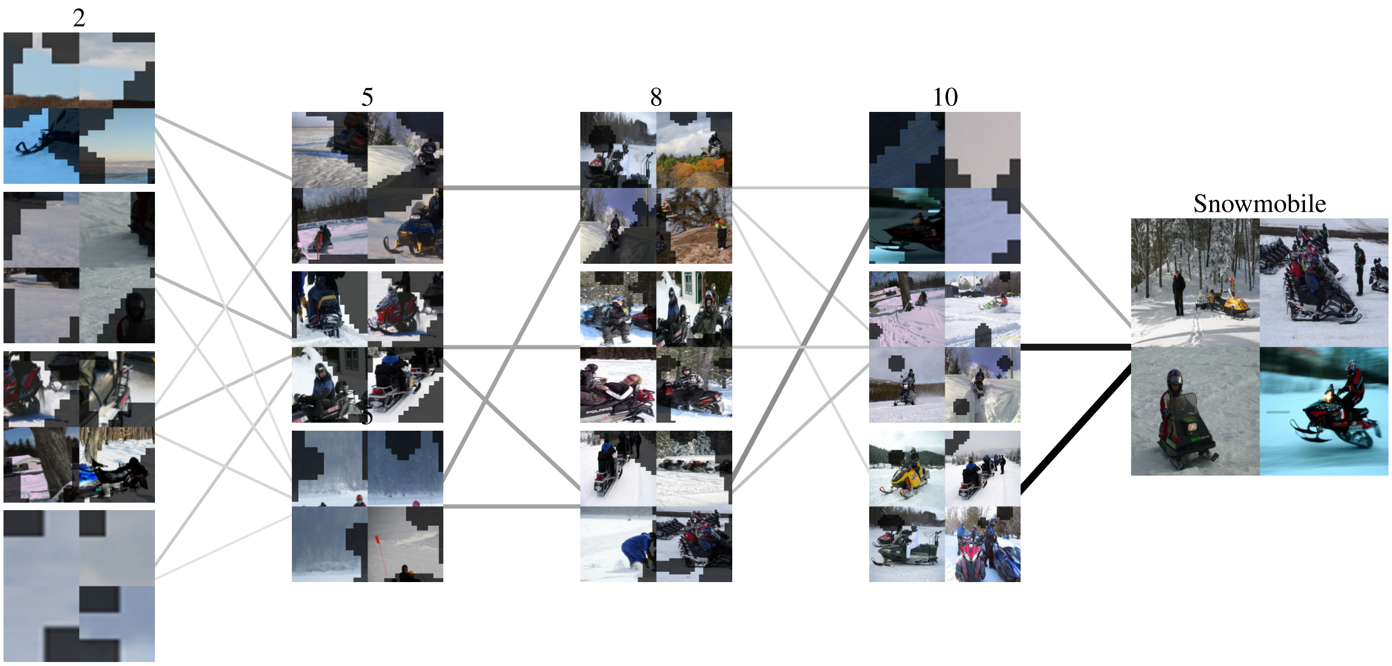

Figure 17. A VCC for four layers of a ViT [16] model targeting the class "snowmobile". Darker lines denote larger concept contributions.

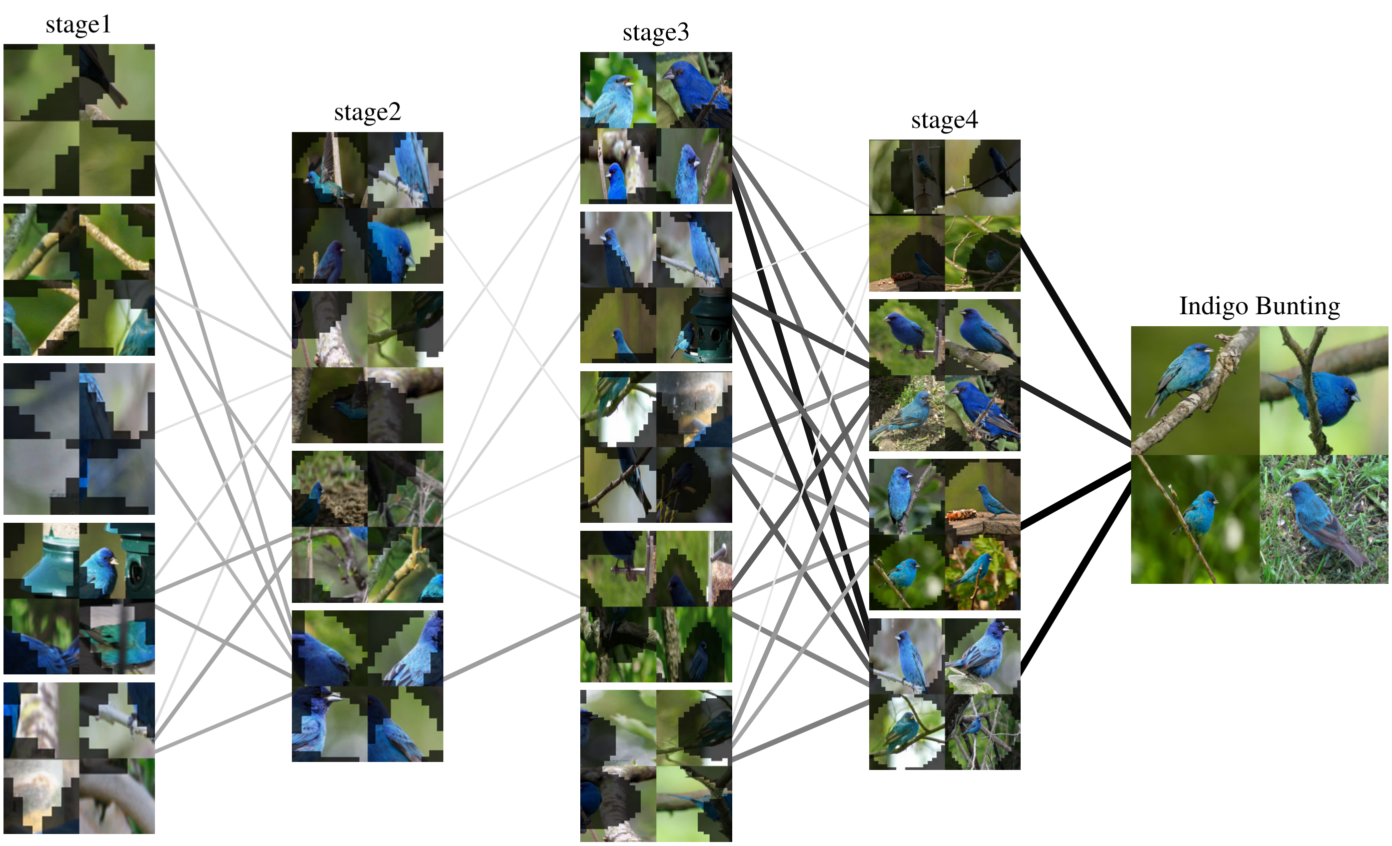

Figure 18. A VCC for four layers of a ResNet18 [25] model trained on the finegrained CUB [52] dataset, targeting the class "indigo bunting". Darker lines denote larger concept contributions.

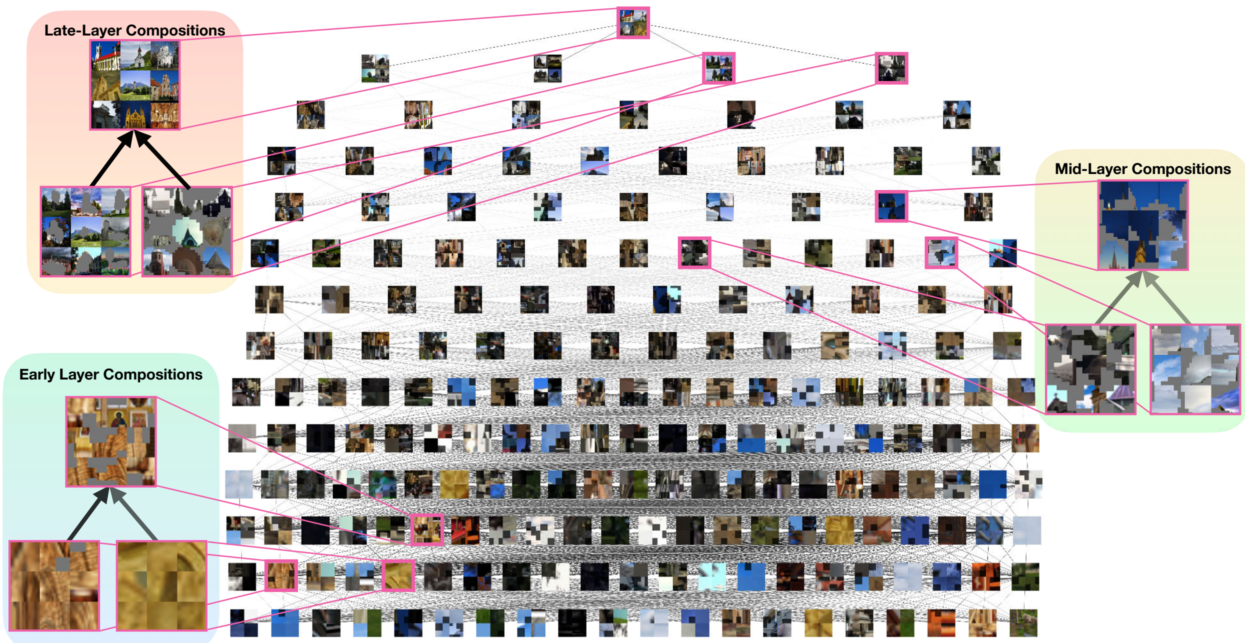

Figure 19. An all-layer VCC of the VGG16 network targetting the class “church”. Darker lines denote larger concept contributions.

References

[1] Reduan Achtibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. From "where" to "what": Towards human-understandable explanations through concept relevance propagation. *arXiv preprint arXiv:2206.03208*, 2022. 3

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 2018. 2

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Conference on Computer Vision and Pattern Recognition*, pages 6541–6549, 2017. 2, 5, 1, 3

[4] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *European Conference on Computer Vision*, pages 351–369. Springer, 2020. 2

[5] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *Artificial Neural Networks and Machine Learning*, pages 63–71. Springer, 2016. 2

[6] Moritz Böhle, Mario Fritz, and Bernt Schiele. B-cos networks: Alignment is all we need for interpretability. In *Conference on Computer Vision and Pattern Recognition*, pages 10329–10338, 2022. 2

[7] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html. 3

[8] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018. 1

[9] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *Winter Conference on Applications of Computer Vision*, pages 839–847, 2018. 2

[10] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021. 1

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 7

[12] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. 2

[13] Ming-Ming Cheng, Peng-Tao Jiang, Ling-Hao Han, Liang Wang, and Philip Torr. Deeply explain CNN via hierarchical decomposition. *International Journal of Computer Vision*, 131:1091–1105, 2023. 3, 8

[14] European Commision. Laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. *European Commision*, 2021. 5

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 6, 1, 4

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. 2021. 6, 4, 9

[17] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html. 3

[18] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *International Conference on Computer Vision*, 2021. 5, 2, 3, 4, 8

[19] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman. Deep insights into convolutional networks for video recognition. *International Journal of Computer Vision*, 128:420–437, 2020. 1, 2

[20] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: Concept recursive activation factorization for explainability. In *Conference on Computer Vision and Pattern Recognition*, pages 2711–2721, 2023. 1, 2, 3, 6

[21] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations*, 2018. 7

[22] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *AAAI Conference on Artificial Intelligence*, pages 3681–3688, 2019. 2

[23] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, 2019. 1, 2, 3, 4, 5, 6

[24] Sven Ove Hansson, Matts-Åke Belin, and Björn Lundgren. Self-driving vehicles-An ethical overview. *Philosophy & Technology*, 34:1383–1408, 2021. 1

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference*

on Computer Vision and Pattern Recognition, pages 770–778, 2016. 5, 6, 7, 2, 4, 9

[26] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, and Vijay Vasudevan. Searching for MobilNetv3. In *International Conference on Computer Vision*, pages 1314–1324, 2019. 5, 3, 4, 8

[27] Haiyang Huang, Zhi Chen, and Cynthia Rudin. SegDiscover: Visual concept discovery via unsupervised semantic segmentation. *arXiv preprint arXiv:2204.10926*, 2022. 2

[28] Benjamin Irving. MaskSLIC: Regional superpixel generation with application to local pathology characterisation in medical images. *arXiv preprint arXiv:1606.09518*, 2016. 4, 1, 3

[29] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, and Fernanda Viegas. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677, 2018. 1, 2, 4, 5, 6

[30] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280, 2019. 2

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

[32] Matthew Kowal, Achal Dave, Rares Ambrus, Adrien Gaidon, Konstantinos G Derpanis, and Pavel Tokmakov. Understanding video transformers via universal concept discovery. *arXiv preprint arXiv:2401.10831*, 2024. 1

[33] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv preprint arXiv:2202.01602*, 2022. 6

[34] Mengchen Liu, Jiaxin Shi, Kelei Cao, Jun Zhu, and Shixia Liu. Analyzing the training processes of deep generative models. *Transactions on Visualization and Computer Graphics*, 24(1):77–87, 2017. 2

[35] Stuart Lloyd. Least squares quantization in PCM. *Transactions on information theory*, 28(2):129–137, 1982. 2

[36] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Conference on Computer Vision and Pattern Recognition*, pages 5188–5196, 2015. 2

[37] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54 (6):1–35, 2021. 5

[38] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 1, 2

[39] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks. 1

[40] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in. 2, 3

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 1, 2, 4

[43] Peter J Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. 4, 5

[44] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetv2: Inverted residuals and linear bottlenecks. In *Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2

[45] Anirban Sarkar, Deepak Vijaykeerthy, Anindya Sarkar, and Vineeth N Balasubramanian. A framework for learning ante-hoc explainable models via concepts. In *Conference on Computer Vision and Pattern Recognition*, pages 10286–10295, 2022. 2

[46] 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 *International Conference on Computer Vision*, pages 618–626, 2017. 1, 2, 8

[47] S. Seung. *Connectome: How the Brain's Wiring Makes Us Who We Are*. Houghton Mifflin Harcourt, 2012. 1

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2014. 5, 6, 2, 4

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2, 6, 7, 4

[50] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. Scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. 3

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 5, 7, 2, 3, 4

[52] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech UCSD birds 2011 dataset. Technical

Report CNS-TR-2011-001, California Institute of Technology, 2011. 5, 4, 9

[53] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 20554–20565, 2020. 2

[54] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014. 1, 2

[55] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting CNN knowledge via an explanatory graph. In *AAAI Conference on Artificial Intelligence*, pages 4454–4463, 2018. 2

[56] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018. 2

[57] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting CNNs via decision trees. In *Conference on Computer Vision and Pattern Recognition*, pages 6261–6270, 2019. 2

[58] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for CNN models with non-negative concept activation vectors. In *AAAI Conference on Artificial Intelligence*, pages 11682–11690, 2021. 2

[59] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. 1, 2

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