Attributing Learned Concepts in Neural Networks to Training Data

Nicholas Konz[1,2] **Charles Godfrey**[4,†] **Madelyn Shapiro**[1] **Jonathan Tu**[1]

Henry Kvinge[1,3] **Davis Brown**[1]

[1]Pacific Northwest National Laboratory [2]Duke University
[3]Department of Mathematics, University of Washington [4]Thompson Reuters Labs

Abstract

By now there is substantial evidence that deep learning models learn certain human-interpretable features as part of their internal representations of data. As having the right (or wrong) concepts is critical to trustworthy machine learning systems, it is natural to ask which inputs from the model's original training set were most important for learning a concept at a given layer. To answer this, we combine data attribution methods with methods for probing the concepts learned by a model. Training network and probe ensembles for two concept datasets on a range of network layers, we use the recently developed *TRAK* method for large-scale data attribution. We find some evidence for *convergence*, where removing the 10,000 top attributing images for a concept and retraining the model does not change the location of the concept in the network nor the probing sparsity of the concept. This suggests that rather than being highly dependent on a few specific examples, the features that inform the development of a concept are spread in a more diffuse manner across its exemplars, implying robustness in concept formation.

1 Introduction

Given the role that concepts play in understanding and explaining human reasoning, measuring their use in neural networks is important for the goal of developing explainable and trustworthy AI. Driven by this, substantial effort has gone into developing methods that measure the presence of a concept within a neural network. Relatedly, a growing body of empirical work shows that deep neural networks learn to encode features as *directions* in their intermediate hidden layers [Merullo et al., 2023, Wang et al., 2023]. A common approach to finding these directions (or **concept vectors**) is via linear probing [Alain and Bengio, 2016]. While probing has well-known shortcomings [Ravichander et al., 2020], it is hard to overstate the impact that concept probing has had on deep neural network interpretability. Prominent examples include probing for syntactic concepts in 'BERTology' [Rogers et al., 2021] and chess concepts in the AlphaZero network [McGrath et al., 2021].

A separate thread in explainability research explores **data attribution**, which, rather than measuring the importance of a concept for a model prediction, quantifies the impact of individual *training datapoints* on a given model prediction (e.g., which images in the training set were most relevant for a classifier's prediction "zebra"?). Data attribution methods have proven to be effective for identifying brittle predictions, quantifying train-test leakage, and tracing factual knowledge in language models back to training examples [Ilyas et al., 2022, Park et al., 2023, Grosse et al., 2023].

† Work done at Pacific Northwest National Laboratory.

ATTRIB: Workshop on Attributing Model Behavior at Scale at NeurIPS 2023.

In this work, we explore an interplay between concept vectors and data attribution, with the goal of obtaining a better understanding of how neural networks utilize human-understandable concepts. Namely, we ask the natural question:

Which examples in a model's training data were important for learning hidden-layer concepts?

Overall, we find that the process of learning a concept is robust to both removal of examples (no small subset of examples are critical to learning a concept– this is consistent with the observation made in [Engel et al., 2023] regarding attributions for the classification logit output of models on individual data points) and stable across independent model training runs. While this stability may not be surprising from a human perspective, given that concepts are, by design, supposed to be relatively unambiguous between observers, it is interesting that a similar phenomenon is seen in models.

2 Experimental Methods

In this section we describe the two methods that are central to our study: using a linear probe to detect a concept and then applying data attribution to concept predictions. Let $f(x)$ be an image classification neural network (the "base model") and assume that $f(x)$ has been pretrained on a training set $\mathcal{X}_{\text{tr}}$. Assume that $\mathcal{X}_{\text{val}}$ is the corresponding validation set. We write $f_{\leq i}$ for the composition of the first i layers of f. Finally, for each of the concepts c that we study, we assume that we have a concept training and test set $\mathcal{X}^c_{\text{tr}}$ and $\mathcal{X}^c_{\text{val}}$ where elements of these sets are labeled by whether or not the elements are examples of c.

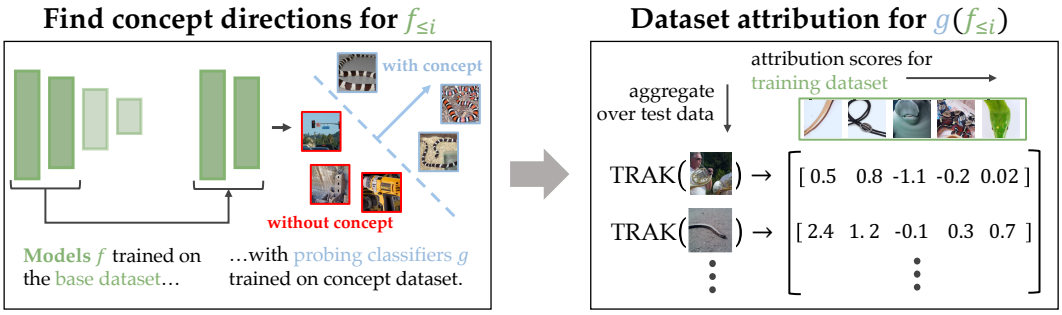

Figure 1: Schematic of our approach for concept attribution: (1) train N models with different random seeds (in green) on the training set. (2) We choose a hidden layer i, append a probing classifier g to its output, freeze the weights of $f_{\leq i}$, and train $(g \circ f_{\leq i})$ on the concept dataset. (3) We calculate attributions with TRAK [Park et al., 2023] for $g \circ f_{\leq i}$ on elements of the test set in terms of the original training data and aggregate across fixed layers and concepts.

2.1 Probing for Concept Learning

The purpose of training a concept probe is to detect whether a specific human-interpretable concept is encoded in a hidden layer of a model. If the linear probe can effectively separate encoded exemplars of the concepts from encoded examples that are unrelated to the concept, then we take this as evidence that the model has learned the concept. More specifically, having chosen the ith hidden layer of f for investigation and a concept c captured by concept training and test sets $\mathcal{X}^c_{\text{tr}}$ and $\mathcal{X}^c_{\text{val}}$, we follow the common approach of training an affine linear probe g on the outputs of $f_{\leq i}(\mathcal{X}^c_{\text{tr}})$ [Kim et al., 2018, McGrath et al., 2022]. Because g's decision boundary is linear, it is effectively summarized by a normal vector which, when the probe is effective, we take to point in the "direction" of concept c (up to a sign). This vector is called a *concept activation vector* (CAV). For short-hand we will use $g_{\leq i}(x) := g(f_{\leq i}(x))$ to describe the "subnetwork + probe" model.

2.2 Attribution of Concept Predictions

The data attribution question that we seek to answer in this paper is: "which examples in f's original training set $\mathcal{X}_{\text{tr}}$ were most important for it learning a concept c?" Since we can quantify how well f

learned a concept at a layer i by the accuracy of a trained probe $g_{\leq i}(x)$ on $\mathcal{X}^c_{\text{val}}$, it is more convenient to ask "which examples in the network's original training set $\mathcal{X}_{\text{tr}}$ were most important for the concept predictions of $g_{\leq i}$ on a set of test images?"

A *data attribution method* $\tau(x_{\text{val}}, x_{\text{tr}}; h)$ is a function that assigns a real-valued score to a training point $x_{\text{tr}} \in \mathcal{X}_{\text{tr}}$ according to its importance to the prediction of a model h on some test/validation point x_{val} [Park et al., 2023]. We will define the expected importance of a training point x_{tr} to concept predictions of $g_{\leq i}$ as

$$\tau_c(x_{\text{tr}}) := \mathbb{E}_{x_{\text{val}} \sim \mathcal{X}_{\text{val}}} \tau(x_{\text{val}}, x_{\text{tr}}; g_{\leq i}), \tag{1}$$

as suggested by Park et al. [2023]. For all attribution experiments we use a recently developed data attribution method, TRAK (Tracing with the Randomly-projected After Kernel) [Park et al., 2023]. For details on this method we defer to the original paper.

TRAK requires an ensemble of M trained models; we use $M = 20$, but found similar results for as few as $M = 5$. Each model in the ensemble is a "subnetwork+ probe" $g^{(j)}_{\leq i}$, where $1 \leq j \leq M$. $g^{(j)}_{\leq i}$ is created by training the same base model f on $\mathcal{X}_{\text{tr}}$ to obtain $f^{(j)}$, then training a probe for a concept c on the i^{th} layer of $f^{(j)}$ with the concept training set $\mathcal{X}^c_{\text{tr}}$, using $\mathcal{X}^c_{\text{val}}$ for validation.

The first step of TRAK is to "featurize" (process $\mathcal{X}_{\text{tr}}$) and score (process $\mathcal{X}_{\text{val}}$) each $g^{(j)}_{\leq i}$, which we run in parallel over the ensemble. After this, the attribution scores of all M networks are aggregated, resulting in $|\mathcal{X}_{\text{val}} \times \mathcal{X}_{\text{tr}}|$ final scores total, one for each pair $(x_{\text{val}}, x_{\text{tr}})$. An important note here is that TRAK requires a task/loss function and corresponding target labels to evaluate the predictions of the models $g^{(j)}_{\leq i}$ — in our case, concept prediction and binary concept labels, respectively. For consistency, unlike the concept probe training and validation sets $(\mathcal{X}^c_{\text{tr}}, \mathcal{X}^c_{\text{val}})$ which use manually-defined concept labels, we use one of the trained $g^{(j)}_{\leq i}$ to assign these labels to $\mathcal{X}_{\text{tr}}$ and $\mathcal{X}_{\text{val}}$ for attribution. We summarize our experimental design in Fig 1.

3 Datasets, Base Models and Concepts

We use "ImageNet10p" as the training and validation sets $\mathcal{X}_{\text{tr}}$ and $\mathcal{X}_{\text{val}}$, respectively, to train the base models. ImageNet10p is defined by randomly sampling 10% of the images of each class from the ImageNet [Deng et al., 2009] training and validation sets. The resulting ResNet-18 models obtained about 45% top-1 accuracy on $\mathcal{X}_{\text{val}}$ (see Appendix F for training details), from which we assume that the model learned meaningful enough visual representations for concepts to be present. We build two different concept probing datasets,[1] and show example images of each concept in Appendix B.

Concept 1: Snakes. We define the "Snakes" concept with the 17 ImageNet snake classes 477-493. The probe training set $\mathcal{X}^c_{\text{tr}}$ is constructed with (i) all examples of these classes in $\mathcal{X}_{\text{tr}}$ and (ii) the same number of non-snake images randomly sampled from $\mathcal{X}_{\text{tr}}$. The probe validation set $\mathcal{X}^c_{\text{val}}$ is constructed in the same manner as $\mathcal{X}_{\text{val}}$; this gives a concept training/validation split of $4,374/170$.

Concept 2: High-Low Spatial Frequencies. We define the "High-Low Frequencies" concept with images where a directional transition between a region with high spatial frequency to a region with low spatial frequency is present [Schubert et al., 2021].[2] We created this concept dataset by computing the top 0.001% highest-activating ImageNet images (by L_∞ norm) of the 'high-low frequency neurons' defined in layer `mixed3a` of InceptionV1 [Schubert et al., 2021]. These images are used with an equal number of random non-concept images to define the concept training and validations sets $\mathcal{X}^c_{\text{tr}}$ and $\mathcal{X}^c_{\text{val}}$, respectively, resulting in a training/validation split of 362/20. We found the high selection threshold necessary to identify images where the concept is clearly present.

[1] We initially experimented with using concepts from the Broden dataset [Bau et al., 2017] but we found probes trained on this dataset did not generalize well to arbitrary ImageNet images.

[2] An analogous variant of this concept was arguably also discovered in biological neurons [Ding et al., 2023].

4 Experiments and Results

4.1 Concept Attribution

In Fig. 2 we display the images in the training set $\mathcal{X}_{\text{tr}}$ which received the highest and lowest concept prediction attribution scores $\tau_c(x_{\text{tr}})$ (Eq. (1)) for each concept, for various layers of the base model. In Fig. 3 we show how the presence of each concept varies with network layer depth, where the two concepts were most present on average in `layer3`. Additional highest-attributed images are in Appendix C.1.

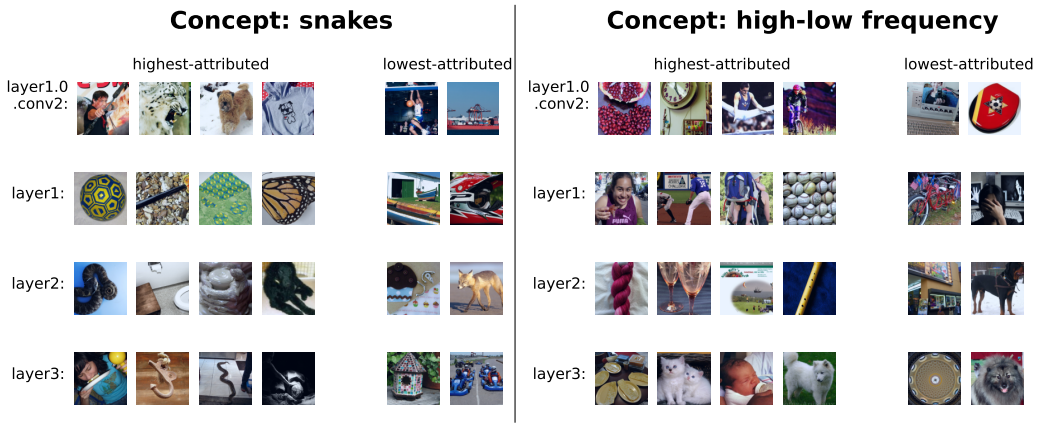

Figure 2: **Training set attributions for concept learning.** The four highest and two lowest attributed training set images (decreasing $\tau_c(x_{\text{tr}})$ from left to right) for concept learning at different network layers. **Left half:** snakes concept. **Right half:** high-low frequency concept.

How does learned concept attribution vary between network layers? For the snakes concept, full snake images appear to be important for concept learning in deeper network layers, while images that possess textures common for snakes are most important for the earlier `layer1`. The concept does not appear to be present in very early layers (`layer1.0.conv2`), which is reasonable given that "snakes" is an abstract concept (see also Fig. 3). These observations are compatible with the conventional wisdom that deeper network layers learn more complex abstract features (such as objects), while earlier layers learn more basic features (such as textures).

The high-low frequency concept is fairly present throughout the network (Fig. 2, right and Fig. 3). Highest-attributed (and certain lowest-attributed) training set images for this concept contain transitions from high to low spatial frequency, such as pomegranate seeds over a flat background (`layer1.0.conv2`, image 1), baseball threading alongside a flat casing (`layer1`, image 4), interwoven threads over a smooth background, (`layer2`, image 1), and fur over a smooth background (`layer3`, image 2). In comparison to the snakes concept probe which has increasing accuracy with network depth, likely due to its connection to the base models classification task, the high-low frequency concept fades after layer 3 as it is synthesized into higher-level concepts related to the label classes.

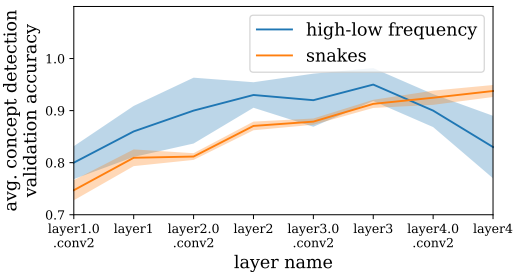

Figure 3: **Concept presence within different network layers.** Average concept detection validation accuracy of probes trained on different layers, for each concept; confidence bands are std. deviation over 5 base models.

Are the concepts that a model learns the result of a few select exemplars? We analyze the importance of images in the base model's training set $\mathcal{X}_{\text{tr}}$ for concept learning by (1) removing the T highest-attributed images from $\mathcal{X}_{\text{tr}}$ to obtain $\mathcal{X}_{\text{tr}}^{-T}$ ($T \in \{100, 1,000, 10,000\}$), (2) re-training

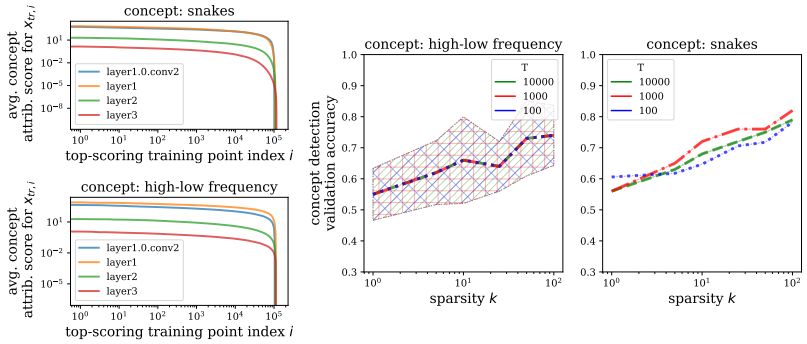

Figure 5: **Left:** sorted distribution of concept learning attribution scores for the training set $\mathcal{X}_{\text{tr}}$, averaged over the validation set $\mathcal{X}_{\text{val}}$, for both concepts. **Right:** Effect of re-training the base model on $\mathcal{X}_{\text{tr}}$ with the top T concept-attributed training examples removed ($\mathcal{X}_{\text{tr}}^{-T}$), for sparse probe concept detection on `layer3`. High-low frequency probe results averaged over 5 base models, with entirely overlapping std. deviation confidence intervals shown.

the M base models on $\mathcal{X}_{\text{tr}}^{-T}$, and (3) training concept probes on each of them for a given layer.[3] If the probe concept detection validation accuracy changes after the training set is pruned of the most important examples of a concept, then we conclude that these examples were primarily responsible for the model learning the concept. If this does not happen, it suggests that a model learns a concept in a more flexible way, from a broad range of examples. For this experiment we measure concepts in the layer where both were most present on average, `layer3`. Our results are shown in the middle and right plots in Figure 5 (where these first experiments correspond to sparsity equal to 10^2), concept validation accuracy did not change on models trained on $\mathcal{X}_{\text{tr}}^{-T}$ for varying T compared to the baseline of those on $\mathcal{X}_{\text{tr}}$, for either concept (Figure 3).[4] This provides further evidence that the learning of a concept is diffuse among exemplars and does not depend on a few special examples. We note that this result may not be surprising given that the attribution scores are mostly similar across examples from $\mathcal{X}_{\text{tr}}^{c}$, Figure 5 left, (e.g., if all examples have the same importance for learning a concept, removing a fraction of them will not have a large effect). In particular, the high-low frequencies concept could be learnable from the majority of images in the training set, especially if it suffices for the probes simply to learn to be boundary detectors; we investigate this with "relative probing" to discriminate between generic object boundaries and the high-low frequency concept in Appendix D.

Finally, given evidence that semantically meaningful representations tend to be sparse in the neuron basis [Gurnee et al., 2023], we also trained sparse probes, thinking that in this regime removing a fraction of exemplars might have a larger effect. In the middle and right plots in Figure 5 and Figure 4 the concepts also remained robust in this setting.

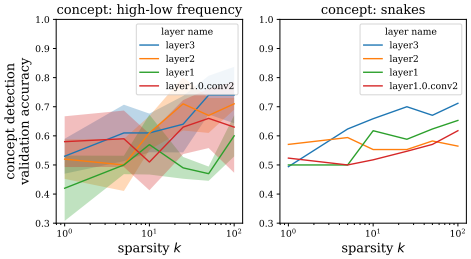

Figure 4: Concept detection validation accuracy vs. concept probe sparsity k.

How similar are different probes trained for the same concept/layer? We show how probe accuracy changes with respect to network layer depth in Fig. 3. For a fixed layer, different probes typically converge to similar performance.[5]

In the case of the high-low frequency concept, we see that probe accuracy is highest at intermediate layers, and comparatively low at the earliest and latest layers. This is consistent with the original work of Schubert et al. [2021], which discovered "high-low frequency detector neurons" in intermediate layers of InceptionV3 networks (but not in the earliest or latest layers). This stands in contrast to the snakes concept, where probe accuracy increases monotonically with network depth. One possible explanation for this observation is that the snakes concept dataset was obtained from a subset of ImageNet classes. As such, the base models have been trained to correctly classify positively-labelled

[3] Concept probe training and validations sets $\mathcal{X}_{\text{tr}}^{c}$, $\mathcal{X}_{\text{val}}^{c}$ are unchanged from their original definition (Sec. 3).

[4] High-low frequency probes at different T for the same base model obtained the same performance due to the small size of $\mathcal{X}_{\text{val}}^{c}$.

[5] In Appendix E we discuss why we do not compare probes via CAV similarity.

concept images using their output logits. Here our observations are consistent with Alain and Bengio [2016], which trained multi-class linear probes for ImageNet classification and found monotonically increasing accuracy with depth.

4.2 Sparse Concept Probing and Attribution

How does forcing probes to be sparse affect concept detection? Forcing a probe to be sparse (at most k of the CAV elements are non-zero) allows for even more interpretable concept directions [Gurnee et al., 2023]. To evaluate this for our concepts, we trained probes with a range of sparsities on different layers of f, using an approach similar to iterative hard thresholding [Jin et al., 2016]. After the first half of the training epochs, we set all but the k parameters of highest absolute value to zero, freeze all of the zeroed parameters from updating, then continue training.

In Fig 4 we show how the concept detection validation accuracy changes with probe sparsity k at multiple layers, for both concepts. Reasonably, probe accuracy typically increases with k, and we see a similar relative accuracy ranking of different layers as in the non-sparse case (Fig. 3). We see that the concepts are both somewhat learnable with sparse probes, but not nearly to the degree of the non-sparse probes (Fig. 3).

Fig. 6 displays the highest- and lowest-attributed images for probes of varying sparsity k, for each concept at `layer3` (compare with the attributions of non-sparse probes in Fig. 1). For the snakes concept, the attributions are different than those for the non-sparse probe, and yet very similar among the sparse probes. Interestingly, we see that almost all of the highest- or lowest-attributed images possess a "honeycomb"-like texture which appears similar to snake scales. For the high-low frequencies concept, we see that the $k = 1$ sparse probe obtained the same attributions as the non-sparse probe, yet the probes with more non-zero entries both obtained the same distinct attributions, which also appear to have examples of the concept.

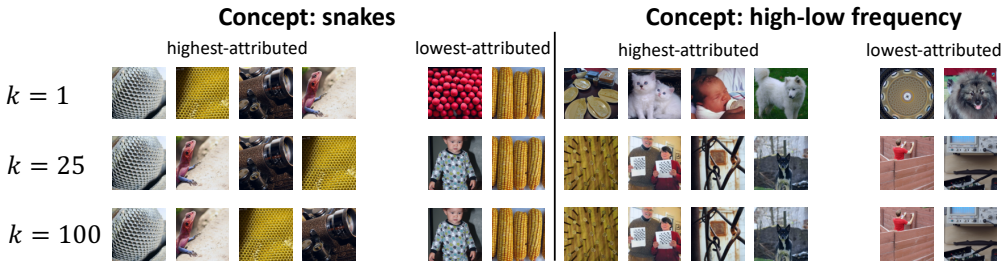

Figure 6: **Training set attributions for sparse probe concept learning.** The four highest and two lowest attributed training set images for concept learning at `layer3` for probes of different sparsity k. **Left half:** snakes concept. **Right half:** high-low frequency concept.

5 Limitations

In order to experimentally vary factors including model layer, concept and concept training data, we were forced to restrict other variables. We only experiment with ResNet18 image classifiers trained on ImageNet10p and two concept datasets (snakes and high-low frequency) — adding additional base models and concepts would increase the breadth of this study. Another interesting future direction would be conducting analogous experiments on a different modality (e.g., natural language) or task. Finally, we only use TRAK for data attribution, and it would be interesting to know the extent to which our experimental results are particular to TRAK.

Conclusion

In this paper we explored the importance of individual datapoints in concept learning. We found evidence of "convergence" in several senses, including stability under removal of exemplars and across independent training runs. Although more extensive experiments are needed (with better aggregation methods), our results suggest a robustness to the way that concepts are learned and stored in a deep learning model.

Acknowledgments

This research was supported by the Mathematics for Artificial Reasoning in Science (MARS) initiative at Pacific Northwest National Laboratory. It was conducted under the Laboratory Directed Research and Development (LDRD) Program at at Pacific Northwest National Laboratory (PNNL), a multiprogram National Laboratory operated by Battelle Memorial Institute for the U.S. Department of Energy under Contract DE-AC05-76RL01830.

The authors would also like to thank Andrew Engel for useful conversations and feedback related to the paper.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *International Conference On Learning Representations*, 2016.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

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*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Zhiwei Ding, Dat T. Tran, Kayla Ponder, Erick Cobos, Zhuokun Ding, Paul G. Fahey, Eric Wang, Taliah Muhammad, Jiakun Fu, Santiago A. Cadena, Stelios Papadopoulos, Saumil Patel, Katrin Franke, Jacob Reimer, Fabian H. Sinz, Alexander S. Ecker, Xaq Pitkow, and Andreas S. Tolias. Bipartite invariance in mouse primary visual cortex. *bioRxiv*, 2023. doi: 10.1101/2023.03.15.532836. URL `https://www.biorxiv.org/content/early/2023/03/16/2023.03.15.532836`.

Andrew Engel, Zhichao Wang, Natalie S. Frank, Ioana Dumitriu, Sutanay Choudhury, Anand Sarwate, and Tony Chiang. Faithful and efficient explanations for neural networks via neural tangent kernel surrogate models. *arXiv preprint arXiv: 2305.14585*, 2023.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying large language model generalization with influence functions. *arXiv preprint arXiv: 2308.03296*, 2023.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9525–9587. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/ilyas22a.html`.

Xiaojie Jin, Xiaotong Yuan, Jiashi Feng, and Shuicheng Yan. Training skinny deep neural networks with iterative hard thresholding methods. *arXiv preprint arXiv:1607.05423*, 2016.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. FFCV: Accelerating training by removing data bottlenecks. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. `https://github.com/libffcv/ffcv/`. commit `xxxxxxx`.

Thomas McGrath, A. Kapishnikov, Nenad Tomašev, Adam Pearce, D. Hassabis, Been Kim, U. Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 2021. doi: 10.1073/pnas.2206625119.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple word2vec-style vector arithmetic. *arXiv preprint arXiv: 2305.16130*, 2023.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Abhilasha Ravichander, Yonatan Belinkov, and E. Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? *Conference of the European Chapter of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2021.eacl-main.295.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2021.

Ludwig Schubert, Chelsea Voss, Nick Cammarata, Gabriel Goh, and Chris Olah. High-low frequency detectors. *Distill*, 2021. doi: 10.23915/distill.00024.005. https://distill.pub/2020/circuits/frequency-edges.

Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept algebra for score-based conditional model. In *ICML 2023 Workshop on Structured Probabilistic Inference $\{\backslash\&\}$ Generative Modeling*, 2023.

A Related Work

Concept vectors: Using linear probes to study the hidden features of neural networks dates back at least as far as [Alain and Bengio, 2016], although in that work, the probes were performing the same task used to train the underlying neural network. The framing of concept vectors originates in [Kim et al., 2018], which demonstrated their usefulness for making neural networks more interpretable. There has been a large amount of follow-up work concerning concept vectors, neuron interpretability, and linear representations.

Data attribution: Our primary references are the *datamodels* framework [Ilyas et al., 2022] and its more computationally-tractable approximation TRAK [Park et al., 2023]. For a more thorough discussion of related work on data attribution (which is a classical topic in statistical learning), we refer to these two papers.

B Example Concept Images

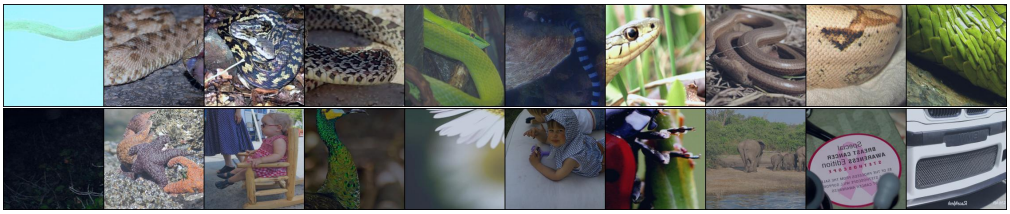

Figure 7: Example images from our concept datasets. **Top:** Snakes; **Bottom:** High-Low frequencies.

C Additional Results

C.1 Highest-Attributed Training Set Images for Concept Learning

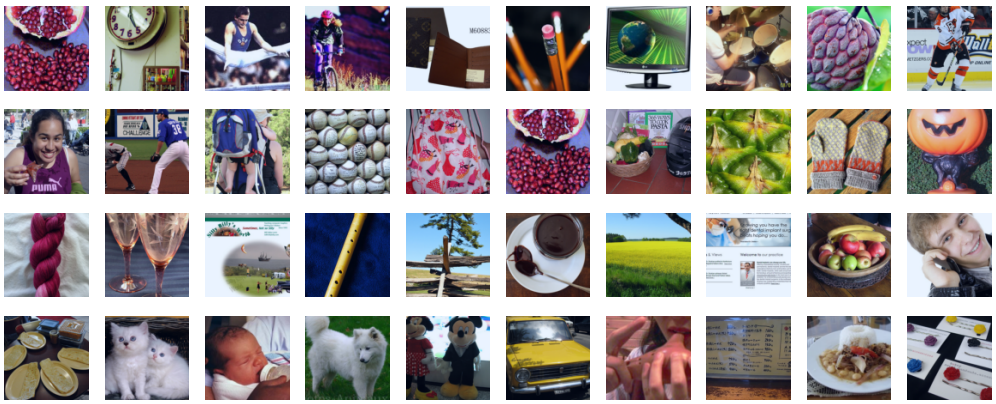

Figure 8: For the high-low frequency concept, the ten highest attributed training set images (decreasing $\tau_c(x_{\text{tr}})$ from left to right) for concept learning at different network layers. **From top to bottom:** `layer1.0.conv2`, `layer1`, `layer2`, `layer3`.

D Relative Concept Probing: High-Low Frequencies vs. Boundaries

In this section, we explore probing for the "High-Low Frequencies" concept further, to attempt to judge if the probes can actually detect the concept, or are relying on the simpler task of boundary detection. To do so, we train probes to differentiate between the high-low frequencies concept and a concept of boundaries, as follows.

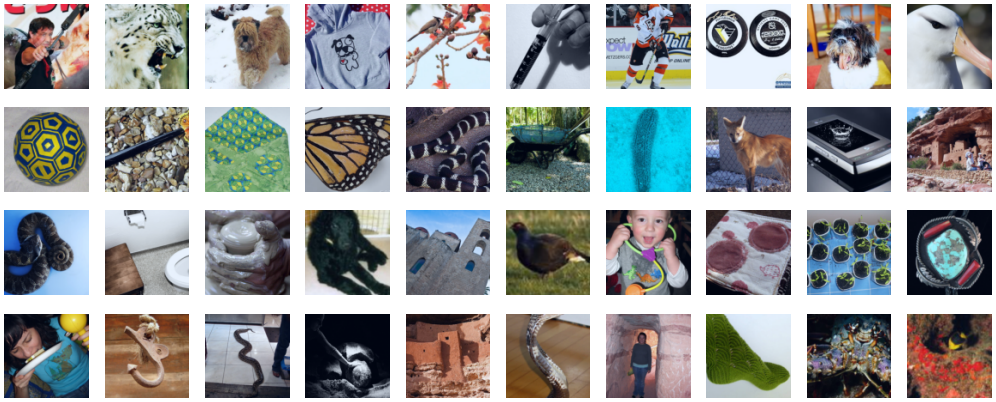

Figure 9: For the snakes concept, the ten highest attributed training set images (decreasing $\tau_c(x_{\text{tr}})$ from left to right) for concept learning at different network layers. **From top to bottom:** `layer1.0.conv2`, `layer1`, `layer2`, `layer3`.

We create the boundaries concept with a similar procedure to the creation of the high-low concept (Sec. 3): we take the top 0.001% highest-activating ImageNet training images in for the 57 neurons identified as boundary or curve detectors in `mixed4b` of InceptionV1 [Schubert et al., 2021], to define the boundary images for concept training, and we do the same from the ImageNet validation set (using the activation percentile created from the training set) to define the boundary images used for concept validation. The final concept training and validation sets $\mathcal{X}^c_{\text{tr}}$ and $\mathcal{X}^c_{\text{val}}$, respecitvely, are then created by combining these boundary concept images with the respective high-low frequency concept training or validation images, and random-sampling from the concept with more examples to make the final $\mathcal{X}^c_{\text{tr}}$ and $\mathcal{X}^c_{\text{val}}$ equally class-balanced. This results in a concept train/validation split of 362/20.

After training probes to differentiate between these two concepts, we see that across the network depth, they are able to succeed ($\sim$ 75–90$\%$ accuracy on $\mathcal{X}^c_{\text{val}}$), especially in `layer2` (Fig. 10). This leads us to believe that our high-low frequency probes can indeed tell the difference between that more complex concept than the "proxy" of simple boundaries. The base model training set attributions for this relative probing are also interesting (Fig. 11). Top-attributed images have clear examples of high-low frequency transitions (top three images) or boundaries (fourth-highest image); intuitively the type of images from which the differentiation between these two concepts could be learned.

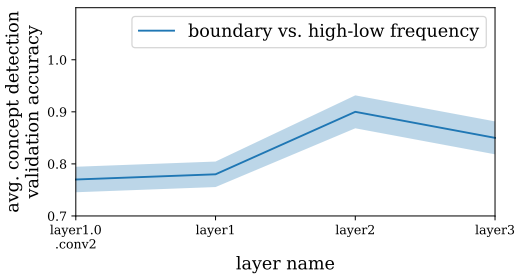

Figure 10: Average concept detection validation accuracy of probes trained on different layers, for boundary vs. high-low frequency concept classication; confidence bands are std. deviation over 5 base models.

E Additional Discussion

Why we compare probes with concept detection accuracy instead of CAV similarity. Conceivably, we could compare the concept activation vectors (CAVs) of the M different probes+base models trained for the same concept and layer in order to see how much the learned concept direction varies, e.g. with the cosine similarity of two probe CAVs. However, as each probe

Figure 11: Four highest- (**left**) and two lowest-attributed (**right**) images in the base model training set $\mathcal{X}_{\text{tr}}$ for relative probing of boundary vs. high-low frequency concepts in `layer 2` of the base model.

is trained on a different base model, the activations $f_{\leq i}(x)$ in the given layer for each model may possess different bases, such that comparing the probe CAVs directly may not be informative. If we assume the activation bases to only vary by some rotation U, we can instead simply compare the probe predictions, as follows. Consider two trained base networks $f^{(1)}$ and $f^{(2)}$ with CAVs $a^{(1)}$ and $a^{(2)}$, respectively, such that $f^{(1)}_{\leq i}(x) = U f^{(2)}_{\leq i}(x)$ and $a^{(1)} = U a^{(2)}$. Then $(a^{(1)})^T f^{(1)}_{\leq i}(x) = (U a^{(2)})^T U f^{(2)}_{\leq i}(x) = (a^{(2)})^T U^T U f^{(2)}_{\leq i}(x) = (a^{(2)})^T f^{(2)}_{\leq i}(x)$, since $U^T U = I$. Therefore, the probe logits $(a^{(j)})^T f^{(j)}_{\leq i}(x)$ (and predictions, ignoring bias terms) can all be compared directly.

F Base Model and Concept Probe Training Details

For training our base ResNet18s, we use the FFCV library [Leclerc et al., 2023] — our training code is a fork of their ImageNet demo with the following changes:

- We drop in our ImageNet10p (and for the "drop top-T highest attribution images" experiments, the appropriate subsets of ImageNet10p) datasets.
- We create a variant of their `rn18_88_epochs.yaml` 88-epoch ResNet18 training configuration with a lower base learning rate (0.125 as opposed to 0.5), lower batch size (256 as opposed to 1024) and greater total number of epochs (176 as opposed to 88).

The batch size was decreased to facilitate training on NVIDIA V100 GPUs (as opposed to A100s, which the original training config targeted) and the learning rate was decreased proportionally. The total number of epochs was doubled to ensure we trained to convergence, i.e. to be more confident that the low accuracy ($\approx 45\%$) was due to training on only 10% of ImageNet, and not a result of under-optimization. For each variant of ImageNet10p, we trained 20 ResNet18s from different random seeds in parallel on a GPU cluster.

We train concept probes with a binary cross-entropy loss and Adam [Kingma and Ba, 2015], using a learning rate of 5×10^{-5}, batch size of 64, and weight decay of 10^{-5}, for 20 and 50 epochs for our two concepts (snakes and high-low frequencies), respectively. We use standard image augmentations during probe training.