Text2Concept: Concept Activation Vectors Directly from Text

Mazda Moayeri[1], Keivan Rezaei[1], Maziar Sanjabi[2], Soheil Feizi[1]
[1] University of Maryland, [2] Meta AI
{mmoayeri, krezaei, sfeizi}@umd.edu, msanjabi@meta.com

Abstract

Concept activation vectors (CAVs) enable interpretability of a model with respect to human concepts, though CAV generation requires the costly step of curating positive and negative examples for each concept one wishes to encode. To alleviate this bottleneck, we present **Text2Concept***, an efficient method for obtaining CAVs* **directly from text***. Text2Concept extends the multi-modal accessibility of a CLIP model's feature space to that of an arbitrary off-the-shelf vision model, with only the small extra step of training a linear layer on existing data to map the feature spaces to one another. We validate our method qualitatively, by sorting images by similarity to embedded concepts, and quantitatively, by showing surprisingly strong zero-shot classification (enabled via Text2Concept) performance for off-the-shelf vision encoders. Finally, we demonstrate two new interpretability applications of Text2Concept CAVs: building concept bottleneck models with no concept supervision, and diagnosing distribution shifts in terms of human concepts.*

1. Introduction

The representation spaces of deep vision models are undoubtedly rich in semantic structure. However, these deep feature spaces are notoriously challenging for humans to interpret, mainly because it is hard for us to digest thousands of numbers at once. Unlike deep models, which encode concepts as vectors in high (e.g. $d = 2048$) dimensional spaces, humans have developed language to describe the world around us concisely. In this work, we propose a method to map text to concept vectors that can be compared directly to image representations obtained from off-the-shelf vision encoders trained with *no text supervision*.

Our method works by mapping the representation space of a given vision model to the representation space of a CLIP [27] model. By design, CLIP representation space is shared across jointly trained vision and text encoders. Thus, CLIP models already have Text2Concept built in, via the text encoder. To extend this capability to off-the-shelf models, we propose to learn a mapping between representation spaces. Specifically, we optimize a linear layer to predict the representation of an image for a target model (i.e. CLIP) from the same image's representation for a source model (i.e. off-the-shelf vision model). We can then map the representations of the off-the-shelf model to CLIP space, where the aligned features would reside in the same space as the concept vector for the desired text. We can also learn the reverse mapping, which would map the CLIP text embedding for a concept to a CAV for the model of interest.

Figure 1 visually validates our approach: after encoding the concept "in a tree" in CLIP space and computing similarity with mapped (to CLIP) representations from a self-supervised ResNet, the classes with the highest average similarity are reasonable, and most similar class instances display the concept prominently, while the least similar ones do not. Stronger validation of our approach is found in performing *zero-shot classification using off-the-shelf encoders via Text2Concept.* Models achieve impressive zero-shot accuracy on many tasks, often being competitive with a CLIP model that is larger, trained on many more samples with richer supervision, and most notably, directly trained to align with the text encoder we use in Text2Concept.

Additionally, we demonstrate two new ways to use Text2Concept CAVs for improved interpretability. First, we show Text2Concept allows for converting existing vision encoders to *Concept Bottleneck Models* (CBMs) [15] with *no concept supervision*. CBMs decompose inference into a concept prediction step followed by class prediction using a white box model (i.e. linear head) on concept predictions, so that the contribution of each concept to the final logit can be precisely computed. With Text2Concept, we can first predict concept similarities in a zero-shot manner, and then train a new linear head mapping concept similarities to class labels. For RIVAL10 data [22], we obtain a CBM that accurately predicts attributes (AUROC of 0.8) and classes linearly from attributes (93.8% accuracy), leading to the desired interpretability benefits (see Figure 3). Next, we show Text2Concept can demystify large datasets, as the distribution of similarities between a bank of Text2Concept CAVs and aligned (to CLIP) representations of the data essentially summarizes what concepts are present in human

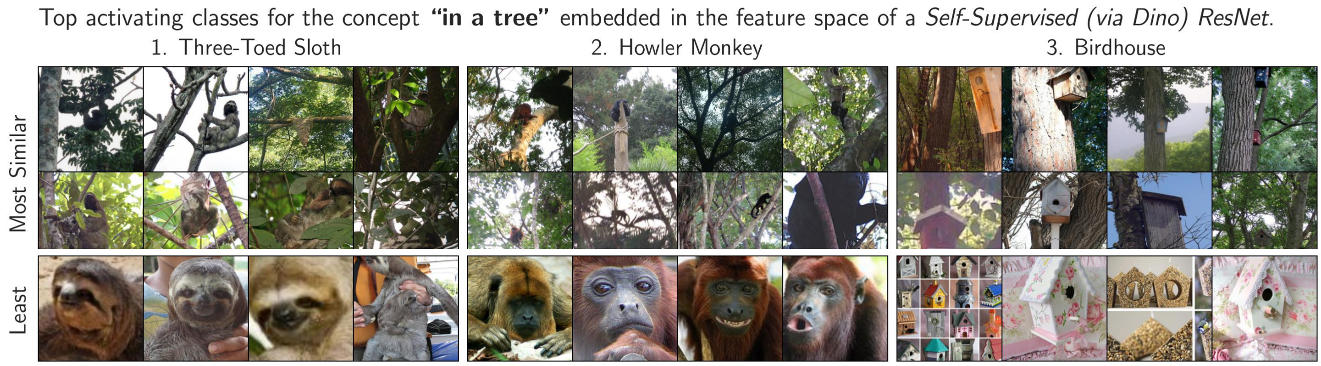

Top activating classes for the concept "in a tree" embedded in the feature space of a Self-Supervised (via Dino) ResNet.
1. Three-Toed Sloth
2. Howler Monkey
3. Birdhouse
Most Similar
Least

Figure 1. Qualitative validation of Text2Concept. ImageNet classes are sorted by the average cosine similarity of the CLIP embedding for "in a tree" to the *linearly aligned* (to CLIP) Dino ResNet representations of images within each class. Highest ranked classes indeed often appear in a tree, as is evident by the most similar instances. The least similar instances appropriately do not contain the concept.

terms. We can then diagnose distribution shifts w.r.t. to human-understandable concept similarities by comparing new data to training data. For example, when comparing ObjectNet [2] to ImageNet, we find the distribution of similarities for the "indoors" concept shifts dramatically, capturing a key reason why ObjectNet poses a challenge: images in ObjectNet were taken in people's homes.

2. Review of Literature

Concept Activation Vectors (CAVS) were popularized by [14], who encoded human concepts as directions in a model's deep feature space, and then interpreted the model by inspecting its sensitivity to changes along these directions. A major limitation is the need for example sets of data to define CAVs – this is an expensive step that scales poorly (i.e. with the number of concepts of interest). More recent efforts automatically discover CAVs [9, 10, 36], though annotating the discovered concepts with language is not straightforward. Our method efficiently obtains CAVs directly from text, resolving the need to curate data to define a CAV or annotate some direction after discovering it. The key step to our method is mapping the feature space of a fixed model to that of a CLIP vision encoder [27], which is jointly trained with vision+text supervision, making it possible to access the vision latent space with text and perform zero-shot classification. Some works leverage CLIP to interpret neural nodes or failure models of other models [13, 25], though they also require probe datasets or exemplars. We map to and from CLIP using linear layers, which is similar to model stitching, first introduced by [19] and later revisited by [1] and [7]. These works, however, typically stitch together models of the same architecture, while we consider a much more diverse set of models. Recently, [23] devise a zero-shot method for mapping across representation spaces based on relative positions to anchor points, though they do not use their mapping for interpretabilty.

3. Text2Concept Method and Validation

Text2Concept encodes text descriptions of semantic concepts as vectors that can be directly compared (i.e. via cosine similarity) with the mapped features of images obtained from an off-the-shelf vision encoder. Despite its simplicity, Text2Concept is surprisingly effective, which, after further detailing our method, we demonstrate qualitatively and quantitatively in this section. Notably, we show that the similarities of aligned image representations to class vectors obtained via Text2Concept enables *zero-shot classification for non-CLIP models off-the-shelf*, with zero-shot accuracy of much simpler models at times exceeding that of CLIP.

3.1. Method Details

We define Text2Concept as a procedure for obtaining vectors corresponding to concepts described as text that can be directly compared (i.e. via cosine similarity) to image representations from a fixed vision encoder. Our method begins with a string describing some concept, like "red food". We then prepend this string with a number of template prompts (e.g. "a photo of {}"); we use the same template prompts as in CLIP's original paper for ImageNet zero-shot classification. Then, we embed the templated text to CLIP space using CLIP's text encoder, and average the resultant vectors over all templates to obtain a single concept vector (as is standard). For some object agnostic concepts, such as contexts like "in a tree", we can encode a general prompt like "a photo of an object in a tree", or we can obtain a more refined vector by encoding "{*prompt*} {*class name*} in a tree", averaging over all choices for *class name* and *prompt*. There are countless ways to prompt engineer; we elect to use general prompts in most cases, as prompt engineering is not the focus of our work.[1]

Then, for a given model, we train a linear layer to map its representation space to CLIP (specifically, CLIP ViT-B/16).

[1] See Appendix A for complete details on all prompts used.

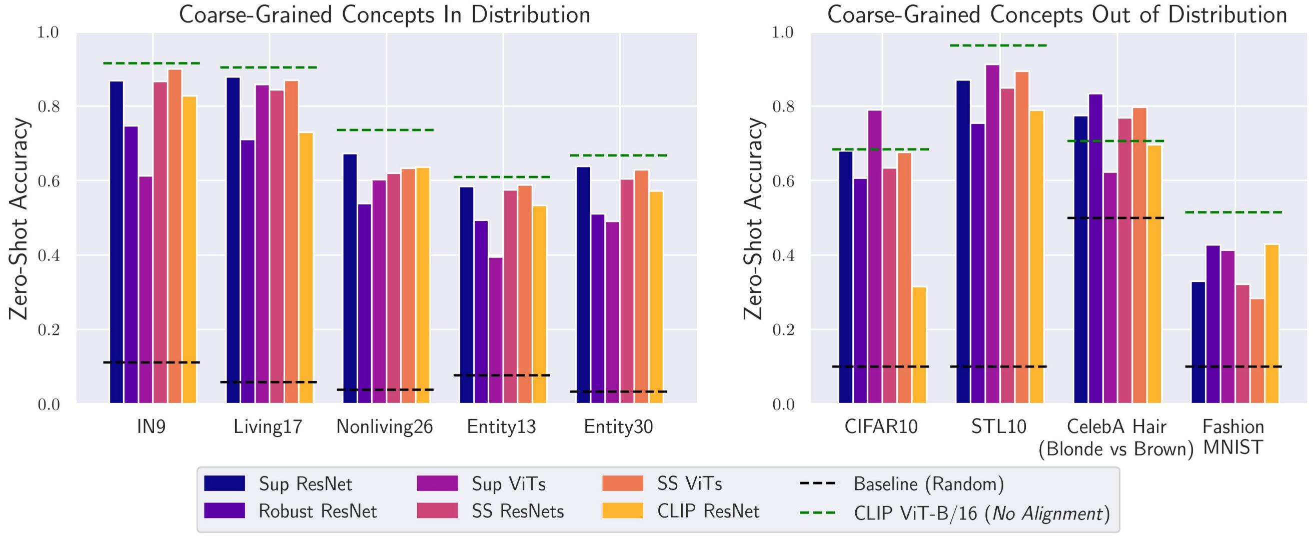


Figure 2. The zero-shot capabilities of CLIP can extend to off-the-shelf vision encoders via alignment based Text2Concept. (**Left**) Models trained on ImageNet can recognize coarse categorizations of ImageNet classes, despite never explicitly being taught them. (**Right**) Off-the-shelf models remain strong zero-shot classifiers even when images are out of distribution. In some cases, they surprisingly surpass the accuracy of the CLIP vision encoder whose jointly-trained text encoder was used to embed each class vector.

We pass ImageNet training images to the given model's feature encoder and CLIP's vision encoder, resulting in a dataset of paired representations to train our linear mapping. Now, we have two functions that map to CLIP's vision space: the CLIP text encoder (since the text and vision representation spaces are shared), and the composition of the given model's encoder with the linear aligner. Since the concept vector obtained via CLIP's text encoder and aligned representations from the given model are both mapped to the same space, we can compare them directly, thus satisfying our definition of Text2Concept. Alternatively, we could train a layer from CLIP to the given model's representation space, mapping the text embedding instead of the features, though we found this method to be less effective. Since we use a simple affine transformation, the mapping minimally changes the content of the representation obtained from the off-the-shelf model. Also, **our approach is very efficient**: after training a linear layer once, we can encode any number of new concepts from text at no additional training cost.

Figure 1 shows images selected based on the cosine similarity of their aligned representations (obtained using an off-the-shelf encoder and a trained linear aligner) to certain concept vectors. For each concept, we present the classes with the highest average similarity, as well as the most and least similar images within them. The retrieved classes are sensible for each concept (e.g. *American Lobster* for "red food", see Figure 5). Sorting images within each class separates examples where the concept is extremely prominent from those where the concept is absent (e.g. images of uncooked lobsters are least similar to the "red food" concept).

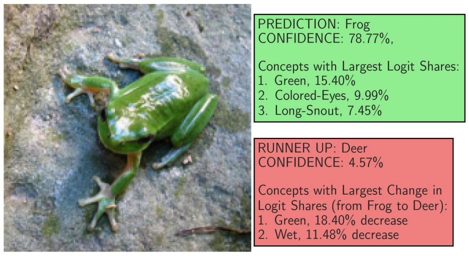
3.2. Zero-Shot Classification

CLIP models can classify a test image to an arbitrary set of classes by embedding text strings describing each class and choosing the class whose embedding is most similar to the test image's representation. This is referred to as zero-shot since no labeled instances from the candidate classes are used. Considering classes as concepts, we can then use Text2Concept to obtain vectors that are directly comparable to aligned representations from off-the-shelf vision encoders, thus extending CLIP's zero-shot capabilities. Zero-shot classification accuracy serves as a quantitative measure of Text2Concept, as higher acccuracy is attained when a Text2Concept class vector aligns better with representations of samples in that class. Thus, we explore zero-shot classification over many datasets to shed insight on when and how well Text2Concept works. We consider models with diverse architectures and training procedures, though all models are roughly equal in size (∼25M parameters) and are only trained on ImageNet (except for CLIP). Also, the baseline CLIP model (ViT-B/16) whose text encoder is used to embed concepts is much larger in size (∼80M parameters); this baseline is intended more so as an upper bound.

First, we ask if models can recognize new categorizations of the data they were trained over. Namely, we consider coarse grained categorizations of ImageNet classes (e.g. distinguishing *insects* from *carnivores*, see [29,34] and Appendix B). Then, we investigate if these coarse grained concepts can still be recognized even if images are taken out of the training distribution. We observe impressive zero-shot performance in both cases (see figure 2). For example, on a 17-way classification problem, self-supervised ViTs achieve 85% accuracy, despite never receiving supervision about these classes, or any classes at that. Shockingly, in a few cases, even the performance of the CLIP model whose text encoder (with which it was jointly trained) was used to obtain concept vectors is surpassed (see Appendix B).

4. Additional Applications of Text2Concept

4.1. Concept-Bottleneck Networks for Free


PREDICTION: Frog
CONFIDENCE: 78.77%,

Concepts with Largest Logit Shares:
1. Green, 15.40%
2. Colored-Eyes, 9.99%
3. Long-Snout, 7.45%

RUNNER UP: Deer
CONFIDENCE: 4.57%

Concepts with Largest Change in Logit Shares (from Frog to Deer):
1. Green, 18.40% decrease
2. Wet, 11.48% decrease

Figure 3. Example inference for a Concept Bottleneck Model (CBM) obtained via training a linear layer on zero-shot concepts. Since logits in the CBM are linear functions of concept scores, we can precisely quantify the contribution of concepts to each logit.

The zero-shot results suggest models are already aware of many concepts beyond those which they are directly trained to learn. One case where knowledge of concepts related to the classification task is salient is Concept Bottleneck Models (CBMs) [15]. CBMs are intepretable by design, as they first predict the presence of concepts using a black box, and then obtain class logits with a white box (e.g. linear layer) atop concept predictions. Thus, the contribution of each concept to the predicted logit can be computed directly, allowing predictions to be faithfully explained with semantic reasons. A major barrier to using CBMs is that they require concept supervision, which can be prohibitively expensive. Text2Concept, however, alleviates this constraint, thanks to zero-shot concept prediction.

We use RIVAL10 classification [22] as an example for how a CBM can be implemented with *no concept supervision* using Text2Concept. RIVAL10 is an attributed dataset, though we do not use these labels during training. We use RIVAL10 because a linear classifier with attribute labels as input achieves 94.5%, indicating that a CBM could be effective. Further, the attribute labels allow for quantifying the quality of the zero-shot concept vectors we obtain.

To implement the network, we use Text2Concept to encode the 28 attributes annotated in RIVAL10 as vectors in CLIP space. We then compute the similarities between the attribute vectors and mapped (to CLIP) features from an ImageNet pretrained ResNet-50. Finally, we fit a linear layer atop image-attribute similarities (i.e. in representation space) to predict class labels. Note that the only training we conduct is that of the final classification head and the aligner, both of which are linear layers, making them very time and sample efficient to optimize. The resultant CBM achieves 93.8% accuracy, and yields the desired interpretability advantages, as shown in figure 3. Moreover, using image-attribute similarities (via Text2Concept) as a score for predicting attributes achieves an AUROC of 0.8, with 72% of attributes achieving at least an AUROC of 0.75. Thus, zero-shots concepts are relatively accurate in predicting RIVAL10 attributes. See appendix C for details.

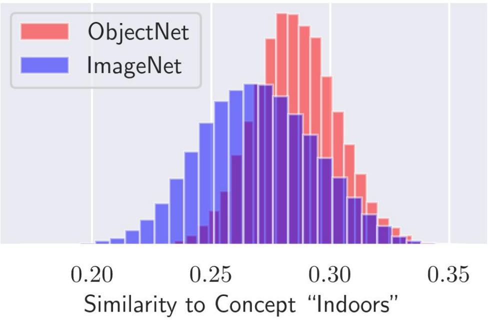


Figure 4. Concept similarities can reveal distribution shifts, like in ObjectNet, where photos are taken within people's homes.

4.2. Concept-Based Dataset Summarization and Distribution Shift Diagnosis

The interpretability benefits of Text2Concept also apply to demystifying large datasets. Namely, one can discern the presence of a concept in their data by using Text2Concept to obtain a corresponding vector, and computing the similarity of this vector to all (mapped) images representations. As modern datasets continue to grow, the need for efficient concept-based summaries of these datasets will also grow; Text2Concept can provide such summaries easily.

Moreover, one can track the distribution of concept similarities for a stream of data over time. Suppose for example a model is deployed to a new setting and it begins to fail. By comparing the distribution of concept similarities in the training set to the new data, one can diagnose the distribution shifts at play. As a proof of concept, we inspect ObjectNet [2], a challenging distribution shift for ImageNet models consisting of images taken within people's homes. Figure 4 shows the distribution of image similarities to the vector for the concept 'indoors' for representations obtained from a ResNet-50 of ImageNet and ObjectNet samples. For ObjectNet, the distribution is significantly (according to a Kolmogorov-Smirnov test) shifted to the right compared to ImageNet. In practice, one may maintain a bank of concepts and track similarities over their stream of data, automatically flagging concepts that experience significant shift.

References

[1] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 34:225–236, 2021. 2

[2] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Joshua B. Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019. 2, 4

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 13

[4] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 13

[5] 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. 8

[6] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 8

[7] Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. *Advances in Neural Information Processing Systems*, 34:5656–5668, 2021. 2

[8] 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. Ieee, 2009. 13

[9] Thomas Fel, Agustin Picard, Louis Béthune, Thibaut Boissin, David Vigouroux, Julien Colin, R'emi Cadene, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. *ArXiv*, abs/2211.10154, 2022. 2

[10] 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, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 13

[12] Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Open clip, 7 2021. 13

[13] Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling model failures as directions in latent space. *arXiv preprint arXiv:2206.14754*, 2022. 2

[14] 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. 2, 10

[15] 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. PMLR, 2020. 1, 4

[16] Anas Korchi and Youssef Ghanou. 2d geometric shapes dataset – for machine learning and pattern recognition. *Data in Brief*, 32:106090, 07 2020. 8

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8

[18] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 8

[19] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015. 2

[20] 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. 8

[21] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space, 2023. 13

[22] Mazda Moayeri, Phillip E. Pope, Yogesh Balaji, and Soheil Feizi. A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19065–19075, 2022. 1, 4, 9

[23] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. *ArXiv*, abs/2209.15430, 2022. 2

[24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 8

[25] Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965*, 2022. 2

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 13

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, A. 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 *ICML*, 2021. 1, 2, 13

[28] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020. 13

[29] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. BREEDS: benchmarks for subpopulation shift. *CoRR*, abs/2008.04859, 2020. 3, 8

[30] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 13

[31] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 13

[32] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 13

[33] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017. 8

[34] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020. 3, 8

[35] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18102–18112, 2021. 13

[36] 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. In *AAAI Conference on Artificial Intelligence*, 2020. 2