



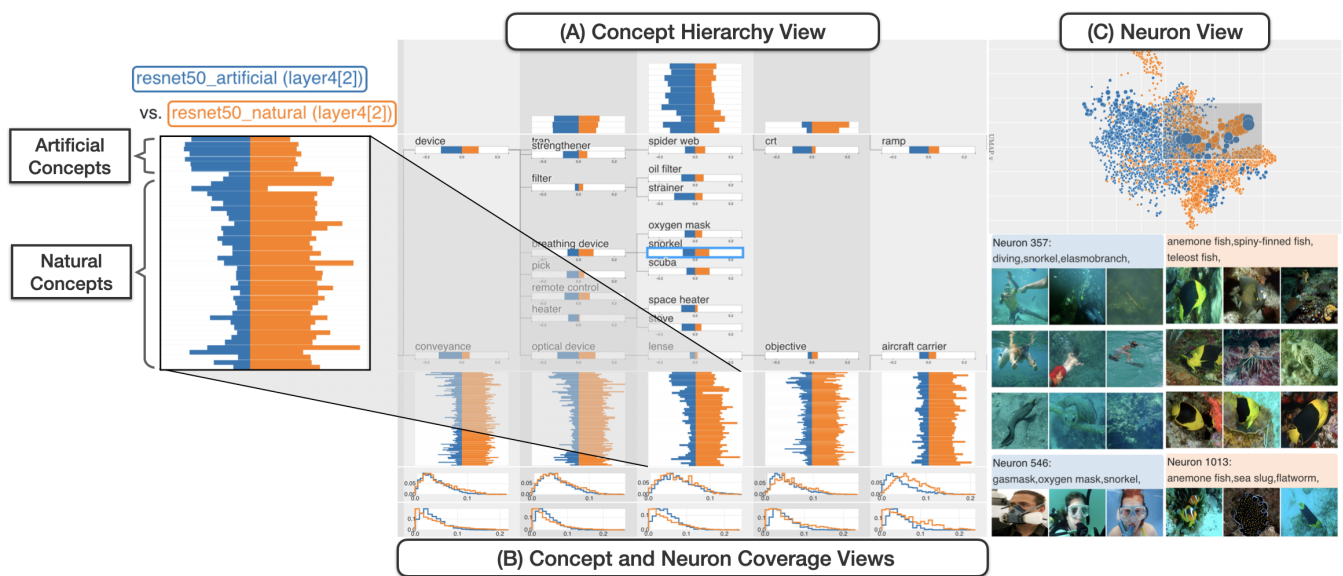


# CAN: Concept-Aligned Neurons for Visual Comparison of Deep Neural Network Models

M. Li<sup>1</sup>  S. Jeong<sup>1</sup>  S. Liu<sup>2</sup>  and M. Berger<sup>1</sup> <sup>1</sup>Vanderbilt University, USA<sup>2</sup>Lawrence Livermore National Laboratory, USA

**Figure 1:** Concept-Aligned Neurons (CAN) compares models trained on different types of data. (A) Concept hierarchy view compares the fraction of neurons in each model that is responsible for each concept word, where the concept is arranged in a hierarchy. We observed that the [model trained on artificial objects](#) (e.g., tools, furniture), compared to [model trained on natural objects](#), fires up concepts of natural objects (e.g., animals, vegetables) much less frequently. The brushed region in the neuron scatter plot (top right, C) highlights the neurons in both models that capture underwater objects in images. Focusing on “snorkel” in the concept hierarchy view (A), CAN indicates that the model trained on artificial objects captures concepts such as diving, snorkel, and oxygen mask, whereas the model trained on natural objects detects sea lives such as anemone fish, sea slug and flatworm (bottom right). c.f. [Sec. 6.2](#)

## Abstract

We present concept-aligned neurons, or CAN, a visualization design for comparing deep neural networks. The goal of CAN is to support users in understanding the similarities and differences between neural networks, with an emphasis on comparing neuron functionality across different models. To make this comparison intuitive, CAN uses concept-based representations of neurons to visually align models in an interpretable manner. A key feature of CAN is the hierarchical organization of concepts, which permits users to relate sets of neurons at different levels of detail. CAN’s visualization is designed to help compare the semantic coverage of neurons, as well as assess the distinctiveness, redundancy, and multi-semantic alignment of neurons or groups of neurons, all at different concept granularity. We demonstrate the generality and effectiveness of CAN by comparing models trained on different datasets, neural networks with different architectures, and models trained for different objectives, e.g. adversarial robustness, and robustness to out-of-distribution data.

## CCS Concepts

• **Human-centered computing** → **Visualization; Visual analytics;**

## 1. Introduction

The growing accessibility of deep neural networks has led to an ecosystem of pre-trained models optimized to solve a variety of tasks. This provides users such as machine learning practitioners a wide variety of models from which to choose, in order to address a given downstream problem, e.g., classification or generation. However, such a large collection of models brings its own problem: for a set of machine learning models designed to solve a specific task, which one should the user choose, and why? As part of making this decision, a user will inevitably have to *compare* models.

There are many ways to conduct comparison [ZWM\*18], and perhaps the most prominent is to summarize a model's performance on a withheld test dataset [RAL\*16]. Yet, many models in areas such as computer vision [DK17] and natural language processing [KBN\*21] are already shown to be highly performant, with only marginal differences between models when summarizing their accuracy. Beyond high-level performance comparisons, one may wish to compare models by their *functionality*, or the organization of knowledge that underlies a model's prediction. Ideally, a model's functionality corresponds well to a human's reasoning for inferential tasks [KWG\*18], and thus functionality can be a strong basis for model comparison. Often, model functionality in neural networks is represented by the functionality of *neurons* [BZK\*17, KNLH19], and it is common to measure what aspects of data that neurons respond to [BZK\*17] in order to associate neurons with human-interpretable *concepts*. Summarizing the resulting attributions, however, gives only a surface-level comparison of model functionality. On the other hand, seeking a more fine-grained comparison of models presents a scalability challenge. Namely, it is difficult to posit *a priori* what concepts should be used to interpret models, and thus recent work [HSB\*21, OW22, BKN\*23] leverages multimodal vision-language models [RKH\*21] to measure neuron functionality using open-vocabulary concept sets. This process can yield thousands of concepts, making it difficult to perform model comparisons at varying concept granularity, e.g. from high-level summaries, down to detailed concept-based analyses.

This work aims to address such challenges in comparing neural networks via neuron functionality. We present CAN, or concept-aligned neurons, a visualization design that aims to support users in comparing modern deep neural networks, specifically computer vision models. CAN builds on prior works [OW22] that aim to interpret neurons along human-interpretable concepts. However, rather than treating concepts as a single unstructured set, CAN organizes concepts in a hierarchical manner, by hypernyms [Mil95], thus providing a multi-level analysis of neuron functionality. Such a hierarchy permits us to compare neural networks along two main axes: (1) concept granularity, and (2) neuron level of detail. The visualization design of CAN is informed by these axes, as summarized in Fig. 1. We visually encode concept granularity horizontally in our interface, allowing for comparisons of models at different levels of detail, e.g., alignment of models along the concept of "dog", or subcategories of dogs. We depict the level of detail of a neuron vertically, showing an aggregated, focus+context view of neuron alignment with concepts (A), a finer-grained summary showing concept-conditioned neuron distributions and neuron-conditioned concept distributions (B), as well as a detailed view of individual

neurons (C). CAN further provides linked-and-coordinated views to support comparing models along a subset of concepts, as well as model comparisons based on a chosen subset of neurons.

By leveraging the proposed visualization design, we demonstrate its usability in a number of comparison tasks of practical importance in computer vision. Specifically, we aim to answer the following questions that often arise in ML research: 1) Do models trained with different types of data learn similar visual representations? 2) How do models of different architectures compare in terms of concepts that are captured by neurons (e.g., ViT vs. CNN)? 3) How do models trained with different objectives (e.g., adversarially trained) compare in their semantic coverage of neurons? Lastly, by answering these questions originated from machine learning researchers, we facilitate in-depth look at the practical usage of concept-level model comparison and evaluate the efficacy of the tool by the obtained insight and direct feedback from the experts. Through a series of case studies, we observed 1) functional difference between neuron groups when comparing models trained on different image domains, 2) sparsity difference in their neuron-representation between different neural network architectures, 3) interpretability difference when comparing models with and without adversarial robustness. The key contributions of the proposed work are summarized below:

- We introduce a novel perspective for model comparison through the concepts captured by individual neurons;
- We introduce a scalable concept-centric visual analytic tool <sup>†‡</sup> for exploring the neuron semantics at different granularity;
- We demonstrate the effectiveness of the proposed method through real-world ML research questions, revealing novel insights and obtaining encouraging feedback from ML experts.

## 2. Related work

CAN draws from two main areas of research: (1) model interpretability, and (2) model comparison. We discuss each in turn, with approaches coming from the machine learning and visualization communities.

### 2.1. Model interpretability

It is a common theory that deep neural networks, optimized to perform visual recognition, contain knowledge about the visual world that is encoded within their learned representations [BJY\*17], usually over a model's set of neurons. To test this theory, prior works often rely on human-annotated datasets of concepts, either at the categorical level of an image [KWG\*18] or object-based segmentation masks [BZK\*17], to test whether neurons are associated with given concepts [BZK\*17, FV18, BZS\*18, MA20]. By aligning neurons with concepts, one may assess how many unique concepts are detected by a set of neurons, indicating the semantic coverage of a model. This provides a means of comparing models in an interpretable manner, e.g. as summarized over concept detection counts.

<sup>†</sup> <https://github.com/tiga1231/can-concept-aligned-neurons>

<sup>‡</sup> <https://observablehq.com/@tiga1231/can-concept-aligned-neurons>

Probing neurons with human-annotated datasets, however, is often costly, as it requires extensive human effort to produce image-level labels or detailed image segmentation masks. More recent works [HSB\*21, OW22, BKN\*23] have sought to address this burden by replacing human annotations with multimodal vision-language models, e.g. CLIP [RKH\*21]. Although such models can sometimes be incorrect, these works nevertheless demonstrate robustness in aligning neurons with concepts that can originate from an open vocabulary of concepts [MGS\*22]. Such a zero-shot approach has the promise of scaling up model interpretability to, in principle, an unbounded set of concepts, with the only human supervision being (1) an unlabeled collection of images on which to probe neurons, and (2) a specification of concepts. Yet the increase in scale presents its own set of interpretability challenges: if we wish to go beyond merely summarizing concept detection, then we must analyze how hundreds-to-thousands of neurons relate to thousands of concepts. It is this problem that we aim to address in CAN.

The visualization community has developed numerous approaches for analyzing the representations learned by neural networks. Often these methods are distinguished by whether they are supervised, e.g. images with class labels/semantic segmentation masks, or completely unsupervised. Unsupervised methods for interpreting neurons often rely on projecting neurons and/or data instances into a 2D embedding space, using either the raw neuron activations [PHVG\*17, KAKC17] or by measuring the overlap between feature maps produced by CNNs [PDD\*21]. By not relying on supervision, these methods are quite general, but at the cost of interpretability: understanding precisely why neurons are related can be a challenge without a means of expressing neurons along human-interpretable concepts.

In contrast, supervised methods for visually analyzing representations can more easily provide insight into why neurons are related to one another, e.g. as shown in Blocks [BJY\*17] and Summit [HPRC19]. Other works leverage per-instance concept-based explanations [GWZK19] to analyze model behavior across a collection of images [HMKB22], while Zhao et al. [ZXS21] employ interactive visualization as a means of incrementally finding interpretable concepts. Moreover, Hoque et al. [HHS\*22] demonstrate the use of discovered concepts to build customized classifiers with minimal human effort. Our approach similarly takes a concept-based approach for analyzing and comparing models, but does so at a much larger scale than prior works, in that we aim to support neuron interpretability with open vocabulary concept sets. We note that our approach to leveraging external hierarchies bears similarity to Bilal et al. [BJY\*17], but we assume arbitrary concepts, rather than relying on class labels used for the prediction task.

## 2.2. Model comparison

Comparisons of deep learning models have been explored both in terms of model predictive performance (i.e., external behavior) and their learned representations (i.e., internal behavior). The most straightforward way to compare models is based on their performance summary statistics, e.g., prediction accuracy. Despite being the de facto standard in the ML domain, however, the summary nature hinders a detailed understanding of their fine-grain behavior. Visualization can provide powerful and interactive so-

lutions to compare model predictions beyond summary statistics. Squares [RAL\*16] adopted a parallel coordinate-like visual encoding to capture class prediction score distribution of different models and how they wrongly classify samples. More recently, manifold [ZWM\*18] presents a more scalable interface that facilitates the comparison of multiple models' predictive performance spontaneously across different slices of the data space. Similarly, boxer [GBYH20] allows users to assess classifiers' performance by composing different views for specific subsets and data. Beyond simple class labels, Neo [GHM\*22] generalizes the confusion matrix visual encoding for understanding the hierarchical and multi-label output of classifiers.

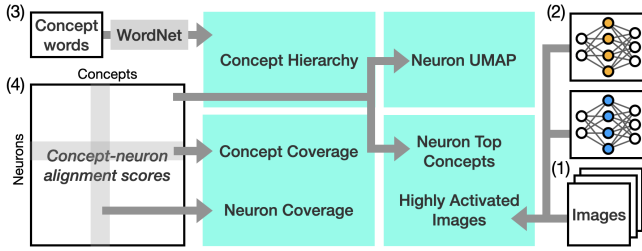
Apart from performance-only comparison, substantial interests have also focused on comparing model internal representations. By looking beyond model performance one can shed light on why models behave differently. Understanding and comparing feature representation help us understand learning mechanisms, and enable us to obtain a deeper understanding of model architecture. Some of the most notable works from the ML community includes metric for comparing feature space similarity, e.g., Central Kernel Alignment (CKA) [KNLH19], where a quantitative distance can be obtained between high-dimensional feature space by evaluating them on the same of input samples. By utilizing such metrics, we can not only study similarities between different layers or how internal representations evolve during model training but also explore the difference between learned representations of different architectures [RUK\*21].

Despite their effectiveness, the global quantitative metric-based approaches are likely ignoring many interesting or important local variations, which is where visualization approaches shine. Several recent works have been proposed to visually compare different feature representations in deep learning models. The Embedding comparator [BCS22] facilitates neighborhood and localized comparison by adopting a small multiple visual encoding for comparing word embedding spaces. The *embcomp* [HKMG20] work, is tailored for a more general vector space comparison task. By utilizing a collection of metrics that summarized the different type relationships between encoding and set algebra operations for selection, the proposed method introduces a flexible novel visual encoding to compare different aspects of embedding at various scales. Besides directly comparing feature representation, another way to study model differences is through peaking into the decision-making process. The VAC-CNN [XZKM22] is proposed to utilize the saliency map as the basis for comparing CNN models.

The proposed work is fundamentally different from all the previously discussed approaches from both ML and visualization domains, as it approaches the comparison from a concept-centric perspective. It aims to provide a comparison of the model's internal representation by exploring concepts of individual neurons and their aggregation.

## 3. CAN Design Objectives

In this section, we describe the problem domain of interest, outline the objectives of CAN, and identify the necessary tasks we need to support in the visualization design to achieve these goals.

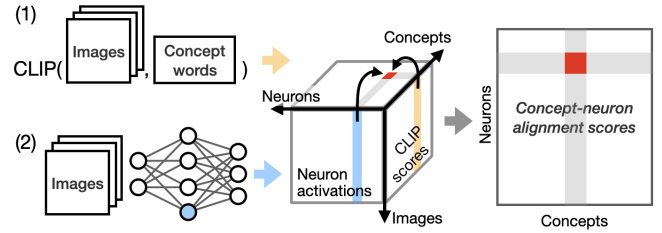


**Figure 2:** CAN illustrated. CAN requires four ingredients: (1) a probing image dataset, (2) a pair of neuron sets to compare (3) a hierarchy of concepts, and (4) the association between neurons and concepts. The arrows point to the interface layout where the ingredients are used.

Our work is focused on helping users conduct comparisons between models, specifically deep neural networks. We assume users performing the comparison have a basic understanding of model architectures, and specifically the layered representation of neural network models and the notion of neurons in each layers representing visual features captured in the images. For each model, we presume the user has selected a particular layer that they wish to analyze. With the common understanding that the model captures low-level features in early layers and high-level concepts in later layers [ZF14], we expect user choose the layer based on the granularity of features they wish to compare. We assume that each layer can be represented by its set of neurons, and thus, model comparison amounts to the comparison of two sets of neurons. Though it is possible to derive measures of similarity between sets of neurons [KNLH19, RUK\*21], such similarity measures lack sufficient context to understand *why* a pair of neurons, or a pair of neuron subsets, might be related, or different, from one another. Therefore, methods that seek to align neurons with *concepts* [OW22, BKN\*23] can provide a way to help users reason about why models are related, and ultimately, draw more meaningful conclusions. In particular, methods that rely on multimodal vision-language models [RKH\*21] to conceive of concepts give, in principle, an unbounded set of concepts with which to compare models, thus ensuring sufficient coverage for contextualizing model similarities/differences.

Nevertheless, the sheer size of the data presents challenges for analysis. Specifically, the number of neurons under inspection can be in the thousands [HZRS16]; likewise, the number of concepts to use in understanding a model can also be in the order of thousands. Thus, in designing a comparison-oriented visualization, it's essential to enable users to understand data at varying levels of detail. By detail, we mean the granularity of concepts, as well as the granularity at which we analyze neurons. Specific to tasks of comparison [Gle17], we wish to perform the following:

- (G1) Understand the alignment of a single neuron with just a single concept.
- (G2) Discover subsets of neurons that are well aligned to a single concept, suggestive of redundancy in knowledge learned by a network.
- (G3) Discover individual neurons that are aligned with multiple concepts, suggestive of multi-semantic neurons.



**Figure 3:** CLIP-Dissect illustrated. CLIP-Dissect estimates concept-neuron alignment by: (1) Given a list of concept words, the CLIP model estimates likelihood of concept presence across an image set. (2) Neuron activations is generated from passing images to the model. Through these two sets of data, CLIP-dissect estimates concept-neuron similarity based on estimations of mutual information between concepts and neurons.

- (G4) Obtain a general understanding of how groups of neurons relate to groups of concepts, and in particular, at different levels of concept granularity.

To support these analysis goals, the design of CAN aims to address the following tasks:

- (T1) Identify concepts associated with neurons, this can be a many-to-one, or one-to-many relationship.
- (T2) Summarize neural network functionality through an aggregate measure of concepts associated with neurons within a concept hierarchy.
- (T3) Explore the relationship between neurons through their concept-based descriptions.

## 4. Neuron-Concept Alignment

In this section, to fulfill T1, we describe the processing methods used to acquire neuron-concept alignment data for CAN's visual analytic interface. The CAN visualization requires four main ingredients (Fig. 2) to be specified beforehand: (1) a set of images with which to probe neurons, (2) a pair of models and the layers that one aims to compare, (3) a hierarchy of concepts used for explaining/annotating neurons, and (4) the association between the neurons and the concepts in the hierarchy, and the ways to enhance the robustness of such estimation. We discuss each in turn.

### 4.1. Probing Dataset

A probing dataset serves the purpose of gathering neuron activations, and ultimately, forms the basis of concept-neuron alignment scores. The dataset is a simple collection of images without any additional data. If the intention is to probe a model for broad coverage of visual concepts, a probing dataset that faithfully represents the visual world should be used. On the other hand, if the intention is to study certain types of data, e.g. of a particular scene, certain types of objects, or images that are known to be out-of-distribution [HMD18] or adversarial [UKE\*20] to standard deep networks, then the probing dataset should be created with the specific task in mind.



## 4.2. Neuron Sets

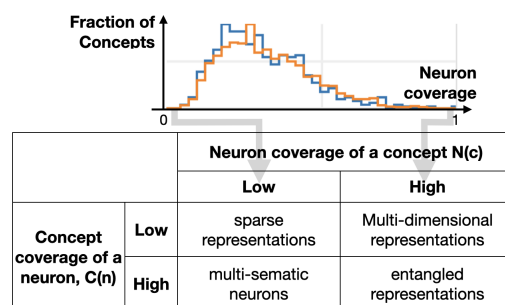
We start from a pair of models that one aims to compare, each of which are deep neural network that is tasked to perform visual recognition. We assume that a user selected a layer from each network, one that is comprised of a set of neurons. Provided an image, the layer forms 2D grids of activations as part of the model's computation, specifically, the learned representations necessary in making a prediction. We use the maximum value over a neuron's grid of activations, following CLIP Dissect [OW22], as the representative value of a neuron. CAN does not make any further assumptions about the types of models for comparison, e.g., the neurons between models need not be aligned in any way, the number of neurons in each set can vary, and we do not make any assumptions about the network architecture, e.g., CNN and ViT.

## 4.3. Concept Hierarchy

Using a large, unstructured concept set for downstream visual analysis can present problems in spotting salient patterns between neurons and concepts. For instance, we expect related concepts, e.g. similar dog breeds, to be associated with similar sets of neurons, but without an organization of concepts that reflects this relationship, we may never find such a pattern. To this end, we propose the use of a **concept hierarchy** to better anchor one's analysis. Specifically, we map the provided set of concepts into WordNet [Mil95] and build a hierarchy based on hypernym relations. Namely, for each concept, we look to see if its hypernym corresponds to a concept already provided in the set. If it does not, then we add this word to the concept set, and continue traversing hypernym relations, until we reach the root concept ('entity'). Upon completion, we obtain an augmented set of concepts, arranged in a hierarchy; we then run CLIP-Dissect on this collection of concepts. See the supplementary material for details. Using a concept hierarchy enables us to perform analysis at different levels of detail: nodes near the root of the hierarchy correspond to abstract concepts, while nodes deep in the hierarchy often refer to rather specific concepts. Moreover, concepts will be naturally grouped by their semantics, e.g., dog breeds will live in a particular subtree. This type of organization is useful in spotting patterns at different levels of detail, e.g. whether a neuron can capture an abstract visual concept, or identifying if a group of neurons captures a semantically related group of concepts.

## 4.4. Associating Concepts with Neurons

Equipped with a set of neuron activations over a probing dataset, the main processing method underlying CAN is the alignment of neurons with a predefined set of concepts. The concept set is simply a list of objects, object properties, scenes, etc., in general, an unbounded, open vocabulary of visual concepts identified by text descriptions. CAN requires a way to compute an alignment score between a given concept and a neuron. In principle, numerous existing methods can be used for this purpose [HSB\*21, OW22, BKN\*23]. We use the scoring mechanism of CLIP-Dissect in this paper, wherein a variant of pointwise mutual information (PMI), soft weighted pointwise mutual information (SoftWPMI) between a concept and a neuron is computed. For a set of given concepts, a collection of CLIP [RKH\*21] scores are computed over the probing images. As a feature of CLIP, the cosine similarity between

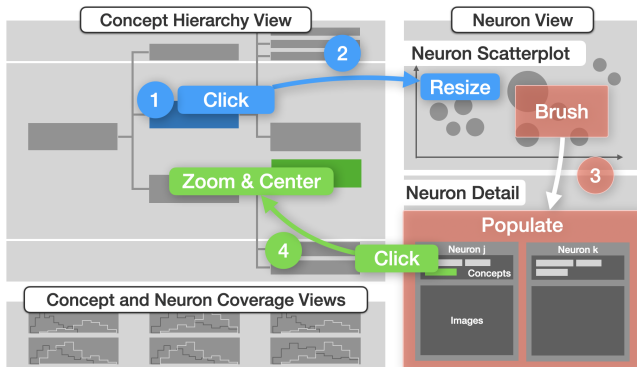


**Figure 4:** Cases of concept and neuron coverage levels and their implications. Together, levels of concept and neuron coverage enumerate four interesting neuron representational behaviors of a model and help users in assessing models in comparative tasks.

the textual embedding of the concept words and the feature of the images (see Fig. 3, top left) are computed. Next, to capture neuron functionality on images, CLIP-Dissect records the activation for all the probing images (Fig. 3, bottom left). Till this end we gathered a pair of image-aligned matrices, one against all concept words and one with all neuron activations, as depicted in the middle of Fig. 3. Finally, by computing SoftWPMI between concepts and neurons, CLIP-Dissect derives concept-neuron alignment scores (Fig. 3, right), which serves as the data feed into CAN's visual analytic interface. SoftWPMI aims to estimate the mutual information between a concept and a neuron. A SoftWPMI value is positive and large magnitude indicates a strong level of co-occurrence between concept and neuron over the probing image dataset. The concept-neuron alignment scores are used in deriving neuron similarities in the neurons view, displaying neuron top concepts in the detail view, and is further processed in estimating neuron and concept coverages.

**Estimating Neuron and Concept Coverage:** As one of CAN's design objectives, we aim to facilitate users in identifying neurons by their type of concept alignment (T1). Toward this purpose, we estimate the neuron and concept coverage over a given layer of a model. Intuitively, the neuron coverage of a concept counts the number of neurons that detect a given concept. Conversely, the concept coverage of a neuron counts the number of concepts that a certain neuron detects. Concepts with a high neuron coverage are captured by a large number of neurons in a model, indicating a certain level of neuron redundancy or multi-dimensionality; a concept with low neuron coverage, on the other hand, indicates the concept being sparsely covered by neurons.

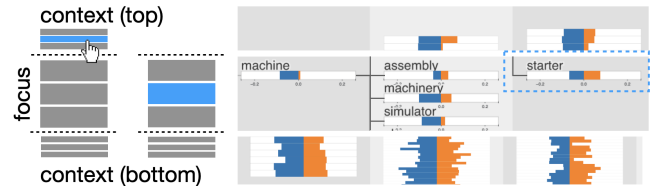
As summarized in Fig. 4, (1) When a concept is uniquely detected by a single neuron, giving low neuron coverage as well as low concept coverage, there is a 1-1 correspondence between the neuron and concept. When this is prevalent among neurons in a model, the neurons sparsely represent concepts; (2) When a concept is densely covered by multiple neurons, giving high neuron coverage, the model requires multiple neurons to encode a concept. For example, a generic concept can have finer, multidimensional representations: instead of the concept of all dogs, the neurons represent individual breeds of dogs; (3) When multiple concepts fire up the same neuron, the model contains multi-semantic neuron(s);



**Figure 5:** Interface layout and linked-and-coordinated interactions in CAN. 1) The user clicks individual concepts in the concept hierarchy view, which will resize the neuron marks based on alliance in the neuron view. The maximal activation images associated with the neuron are revealed by brushing in the neuron view. Additionally, the neuron detail view also lists the associated concepts to each image, allowing a direct lookup of the concept of interests in the concept hierarchy view on click, forming an exploration loop.

(4) When a large group of neurons jointly captures the large group of concepts, giving both high concept and high neuron coverage, a compressed and possibly entangled neuron representation may have transpired in the model.

One approach to derive a measure of neuron or concept coverage involves applying a threshold to the neuron-concept alignment scoring function. For a given concept, we collect neurons with alignment scores surpassing a threshold, then normalize the retained neuron count by the total, yielding the fraction capturing the concept in a neural network layer. The sensible threshold is inspired by CLIP-Dissect [OW22]. Through a user study, CLIP-Dissect found an interpretability cutoff value for the concept-neuron alignment scoring function (SoftWPMI). Interpretability cutoff value aims to filter out uninterpretable neurons. A neuron is deemed uninterpretable if evaluators, on average, found virtually none of the top 10 activated images correspond well to the top concept assigned to the neuron. Their optimal threshold largely ensures that, on average, more than 75% of the top-10 activated images for a neuron that is well-explained by the top concept. In general, one should likewise calibrate the threshold over multiple models and match up the model’s predictive performance, which we considered out of scope in this work. We found the interpretability cutoff point reasonable as the threshold for estimating neuron and concept coverage. A higher threshold in general leads to a more conservative assessment of whether a neuron detects a concept. This results in higher precision but lower recall when identifying neurons capturing a specific concept. CAN incorporates an interactive slider for adjusting this threshold. In practice, users can increase the threshold if the captured concepts fail to explain highly activated images and decrease it when no neurons explain any concept, contradicting model’s high performance. In our case studies, we maintain the threshold at the optimized interpretability cutoff (SoftWPMI > 0.16) determined by the user study in CLIP-Dissect.



**Figure 6:** Features in the concept hierarchy view. **Left:** Clicking on concepts in context view centers the concept. **Right:** When space permits, the closest concept in context view is snapped to the focus, giving a sample preview of the context out of focus.

## 5. Design

In this section, we elucidate the visualization design of CAN. Following the goals/tasks analysis in Sec. 3, specifically T2, T3, our visual design aims to help users:

- Explore a hierarchy of concepts captured by the models, and associate these concepts with their corresponding neuron groups;
- Identify similar and unique neuron(s) from two models and explain the similarity and uniqueness through the concepts they capture.

To facilitate these explorations, the interface bifurcates into two components, concept hierarchy view and neuron view, as illustrated in Fig. 5. The bottom of the interface displays the concept and neuron coverage, summarizing and comparing at each concept level the semantic coverage of neurons in the two neural network models.

### 5.1. Concept Hierarchy View

In the concept hierarchy view (Fig. 6, right), we graphically represent the concept hierarchy as a tree structure (T2), with high-level concepts positioned on the left and low-level concepts on the right. From the concept view, the user can pan over the concept hierarchy and examine and compare the semantic coverage of neurons between the two models. Under the concept names in the hierarchical structure, the view compares *neuron coverage* of concepts between the two models using bar plots. Examining neuron coverage of concepts enables the users to discern and compare the semantic coverage of model layers.

Due to the sheer size of the concept set, one is constrained from observing the entirety of the concept hierarchy. Due to this constraint, we utilize a focus+context embedding of the tree structure. The in-focus concepts are displayed as a tree in the center; out-of-focus concepts are densely packed on the top and bottom context view. The user can scroll through the concept around the focus view, or click concepts in the top or bottom context view to jump to the clicked concept (Fig. 6, left). To further facilitate a sense of context, when space permits, the closest concepts on both context views are snapped to the edge of the tree view in focus, giving a preview of the hierarchy that is out of focus. (Fig. 6, right).

### 5.2. Neuron View

The neuron view compares neurons from two models. The goal of the neuron view is to systematically discern commonalities and

distinctions in neurons between two models, elucidating both the shared attributes and unique characteristics through the concepts they capture. Specifically, the user would be able to explore neurons via their concept-based descriptions (T3) and validate their findings by tracing back to the images that highly activate the neurons.

As an overview for the comparison, we project neurons in a two-dimensional space and plot them in a scatter plot (top right, Fig. 5), with color encoding the model a neuron comes from. The projection is given by UMAP [MHSG18] where between-neuron distances are defined as the Euclidean distances in their concept alliance scores. By design, neurons that detect a comparable set of concepts are grouped in the projection scatter plot. In the context of comparing models, users would seek to examine two distinct cases:



As neurons with comparable functionalities are clustered, the alignment of two models predominantly manifests as a composition of colored dots in the scatter plot.

Conversely, neurons furnishing distinct functionalities in one model, while absent in the other, will coalesce into a cluster of uniformly colored dots.

To facilitate an examination of these groupings and to validate the inferred similarity among neurons as derived from the projection plot, the bottom of the neuron view comprises a details-on-demand display of concepts captured by individual neurons in text, as well as a small sample of images which elicit a high level of activation in each neuron (bottom right, Fig. 5). When a set of neurons is brushed (step 3, Fig. 5) in the neuron projection plot, neurons in the detail view are sorted according to the relevance to the concept if a concept is clicked (step 1, Fig. 5) in the concept view.

### 5.3. Concept and Neuron Coverage View

Concept and neuron coverage views summarize the semantic coverage of neurons and duplicity of neurons respectively. Recall that the neuron coverage of a concept counts the number of neurons that detect a certain given concept. Similarly, the concept coverage of a neuron counts the number of concepts that the neuron detects. Together, combinations of neuron and concept coverage cover four interesting neuron representational scenarios in neural network models, as summarized in Fig. 4.

To compare models in terms of their semantic coverage of neurons, we juxtapose vertically a histogram of neuron coverage over all concepts and a histogram of concept coverage over all neurons in the two models. As, in theory, concepts on a single level of granularity partition the space of all concepts in the visual world, the concept coverage should be computed per level in the concept hierarchy. Therefore, we plot a histogram for each level of concept granularity. That is, the concept and neuron coverage plot on each column is restricted to the set of concepts on that level.

When comparing the distribution of neuron coverage between two models, the difference in the number of neurons between two models can bias the judgment: a model layer with a larger number of neurons is more likely to have functional redundancy or multidimensional representations. We define neuron coverage as the proportion of neurons in a given layer, that detects a given concept,

giving a fraction of neurons ranging from 0 to 1. For a similar reason, the concept coverage counts could be biased towards levels which has more concepts in the hierarchy, therefore, we also normalize the concept coverage counts by the number of concepts in a concept hierarchy level.

### 5.4. Linked-and-coordinated Interactions Between Views

The layout of CAN aims to help users relate concepts to neurons. Therefore, we implemented coordinated views where the left portion is dedicated to exploring concepts and the right to exploring neurons (c.f. Fig. 5). A typical workflow goes back and forth between identifying a concept of interest and looking up neurons that capture it, and identifying neurons of interest and looking up the concepts they capture. The concept hierarchy view and the neuron view are linked and coordinated for this purpose. In a typical workflow, CAN supports interactions as numbered in Fig. 5:

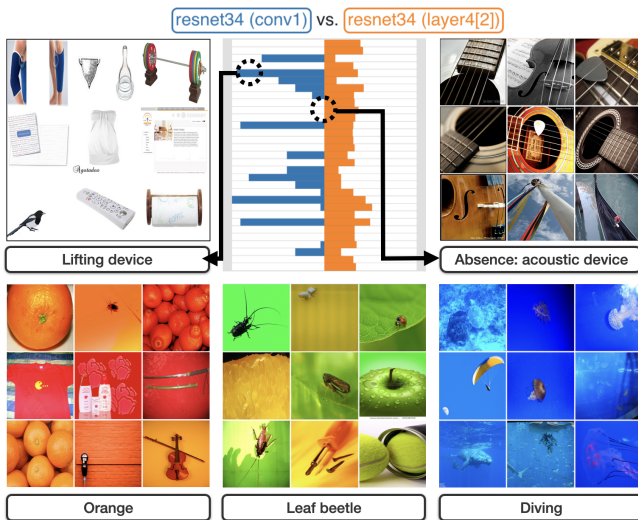
1. Semantic difference is observed from the concept hierarchy;
2. Clicking on a concept resizes neuron marks in the neurons view by the concept-neuron alliance scores of the clicked concept. Neurons that are more aligned with the given concept are shown as larger marks in the neuron plot;
3. Brushing over neurons of interest in the neuron view populates the neuron detail view with top concept words and high-activating images for each neuron, where neurons are ranked by alliance to the concept clicked in step 2;
4. Clicking on concept words in the neuron detail view re-centers the concept hierarchy to the selected concept. This helps the user contextualize the detected concept with other related concepts in the hierarchy.

Note the interactions complete a loop between concept and neuron exploration. A variant of the workflow starts from the exploration of neurons view in step 3, where the user first identifies groups of neurons of interest, either from the mixture or isolation of neurons in the view or from simple exploratory browsing of neuron regions (c.f. Sec. 5.2), then looks up concepts in the hierarchy.

## 6. Case Studies

We demonstrate the effectiveness of CAN in four use cases in relationship to the ML questions raised at the end of the introduction. First, as a sanity check, we compare different layers of the same model, and we expect this example to showcase the tool's capability to capture the evolution of concepts in the neural network. Next, we compare models trained on different data. We split images from ImageNet into two sets based on the class labels and the wordnet hierarchy [Mil95], and trained two CNN models (ResNet-50) [HZRS16] independently. One set contains 550 classes of artificial objects (e.g., cooked food, and furniture), and the other contains 450 classes of natural objects (e.g., animals, plants, and fungi). For this example, we speculate on how the training data affect the model's internal representation and the concepts they capture. After that, we compare models with different architectures. Specifically, we compare a pre-trained convolutional neural network with residual connections (ResNet-50) to a vision transformer (ViT-B/16) [DBK\*20]. Finally, we compare models with different adversarial robustness. Specifically, we compare a ResNet-50 trained on the regular image set to an adversarially-trained one [UKE\*20].





**Figure 7:** Concept evolution from [early layer \(conv1\)](#) to [later layer \(layer4\[2\]\)](#) within the same network. We show that early layers are more irregular in terms of the concepts they learn. Such irregularity is caused by the limited association of the early layer with only a specific set of concepts and the complete absence of certain concepts. We find that early layers learn representation limited to color-related concepts. We also find that early layers oftentimes fail to learn representation of instance-level objects such as acoustic device.

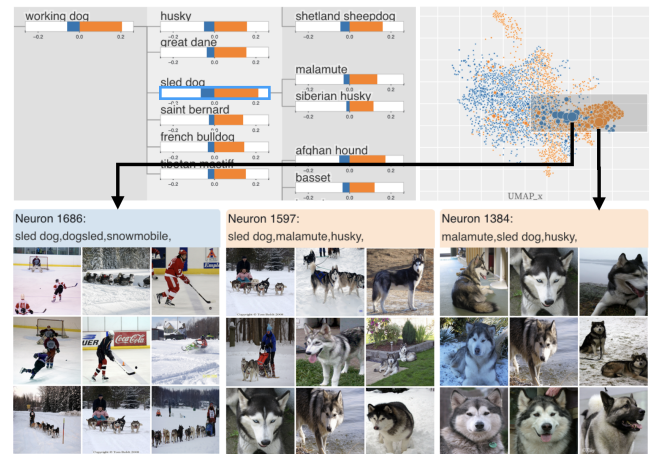
### 6.1. Explore concept evolution in the same model

In this case study, we look at different layers and how concepts emerge or disappear deeper down the network. To accomplish this objective, we conduct a comparative analysis across distinct layers within a singular neural network, aiming to discern potential disparities in the conceptual coverage they encapsulate.

We showcase a comparison of an early and late layer of ResNet-34. While comparing learned concepts at different depths of the network, we find that concepts learned in the early layers show irregularity. Such irregularity divides into two specific observations. First is that neurons in the early layer have a very high association with a very limited set of concepts — in the ResNet case, these are colors. As can be seen in Fig. 7, most of the neurons in the early layers are associated with color or color-related concepts. The second observation is that there are oftentimes concepts that early layers do not pick up at all. These concepts tend to be specific objects such as names of specific instruments or machinery. This is probably more strategic for the later layers to learn as the representation of such specificity will be more beneficial for downstream tasks for neural networks, such as classification tasks. These observations align well with our existing knowledge regarding the functionality of different layers of CNN models, and thus provides a meaningful baseline check to see whether the proposed method fulfilled its design objectives.

### 6.2. Comparing models trained on different data

In this case study, we compare two ResNet-50 models, one trained on artificial objects and the other one on natural objects. We hy-



**Figure 8:** For the concept “sled dog”, CAN indicates that neurons from a [model trained on artificial objects](#) detect working sled dogs and the sleds, while a [model trained on natural objects](#) captures sled dogs, Alaskan malamutes, and huskies in a residential area.

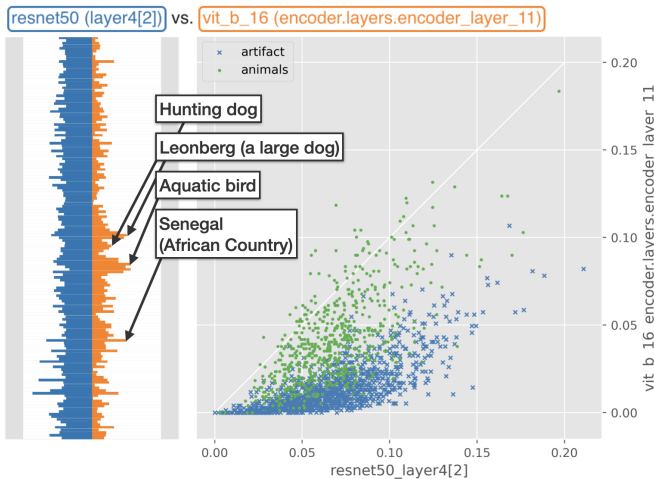
pothesize that models trained on different datasets should exhibit different representational power. According to our design goal (G3), the visualization should be effective in verifying the aforementioned hypothesis.

At a glance, the concept hierarchy shows a diverging trend of the represented concepts within two different models. In Fig. 1, the collapsed bottom of the hierarchy, we can see that neurons inside the model trained on natural objects represent natural concepts significantly more compared to the other model. Similarly, the neuron projection shows a color grouping of neurons from the two distinct models. Even though some neurons from the two models seem to detect the same concepts, as evidenced by the overlap in the neuron view, a closer examination reveals more nuanced differences. In Fig. 1, neurons under the brushed area are related to underwater objects. Neurons from the model trained on artificial objects captures concepts of “snorkel” and “oxygen mask” - man-made artifacts. On the other hand, neurons from the model trained on natural objects capture underwater creatures (c.f. Fig. 1). As another example, among neurons activated by dogs in both models, on the concept “sled dog”, CAN indicates that neurons from a model trained on artificial objects detect working sled dogs and the sleds, while the model trained on natural objects more directly captures dog concepts such as sled dogs, Alaskan malamutes, and huskies that are mostly photographed in a residential area. (c.f. Fig. 8)

### 6.3. Comparing models with different architectures

CAN is also useful in studying different representational power of two different neural network architectures. Previous studies [RUK\*21] find that ViT represents spatial or location-based concepts better compared to ResNet models. Also, from the superior accuracy that the former has over the latter, one may hypothesize that ViT has a better overall representation of concepts. From the collapsed bar plot in CAN, we observe that the most salient difference between the vision transformer (ViT) and convolutional model (ResNet-50) is their neuron representational sparsity. As ev-



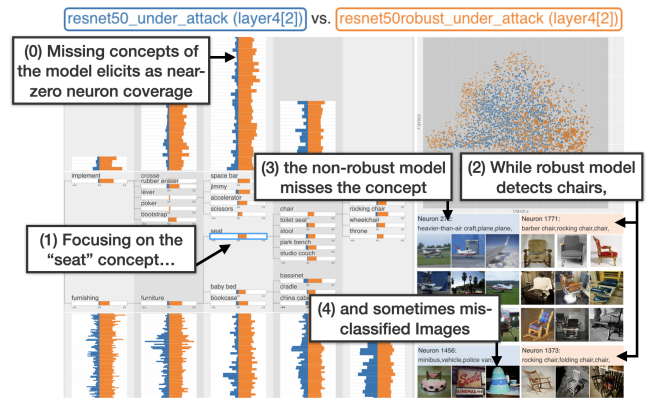


**Figure 9:** Between CNN (*ResNet50*) and Vision Transformer (*ViT*), we found that ViT, despite a much lower neuron coverage on average, dedicates a comparable fraction of neurons to living objects. As an example, compared to man-made items such as sports equipment, ViT dedicates more neuron to concepts such as dogs (e.g., Leonberg), aquatic birds, and lively African countries such as Senegal. The scatter plot compares the neuron coverage over concepts between the two models, where color represent concept type ( $\times$  artifact vs.  $\bullet$  animals).

ident from the concept and neuron plots (Fig. 9), we find that compared to CNN models with skip connections (ResNet-50), vision transformer (ViT) comprises neurons that sparsely cover the concept space. Despite an overall lower neuron coverage among concepts, ViT seems to dedicate a comparable portion of neurons to living objects with respect to a convolutional model, such as animals, insects, and fruits. To validate this hypothesis, in Fig. 9 we compare the neuron coverage between the two models in a scatter plot where dots are colored by concept category (artifact or animals). Compared to previous comparison effects for ViT and CNN that focus on global observation, this sheds new light on differences between these models that has not been documented before [ZHP\*17, LSL\*16] (to the best of our knowledge). The combination of sparse concepts and more natural concept concentration may spark further hypotheses and experiments that lead to a deep understanding of these model architectures.

#### 6.4. Comparing models with different adversarial robustness

In this case study, we examine the effectiveness of CAN in comparing models of different adversarial robustness. We used the fast gradient sign method [GSS14] to generate a set of adversarial examples in the ImageNet validation set. On each of the 1000 classes on ImageNet, we reassigned the image label of the correct class  $i$ , for  $i \in \{0, \dots, 999\}$  to a distant, unrelated wrong class  $i + 500 \bmod 1000$ , and look for adversarial examples on the scrambled labels. Under this set of adversarial attacks, the non-robust pre-trained ResNet-50 model drops to 31.79% classification accuracy, while the robust counterpart retains an accuracy of 69.49%. With CAN, we find that due to adversarial attacks, non-robust model misses, and sometimes misinterprets concepts, as depicted



**Figure 10:** Compared to an *adversarially robust model*, adversarial examples confuse the *non-robust model* by muting some of its concept detectors. As an example, under adversarial attack, no neuron from the non-robust ResNet-50 activates on the “seat” concept, while the adversarially robust model detects concepts such as “barber chair” and “rocking chair”. Moreover, the non-robust model sometimes mis-interprets the concepts, such as interpreting image of a cake as “minibus”, “vehicle” or “police van”. Upon clicking on the “seat” concept, selecting and sorting all neurons in two models by the neuron alliance to “seat”, neurons from the non-robust model do not detect chairs, and sometimes mis-classified the concepts in the images.

in Fig. 10. The neuron coverage bar plots under and on top of the hierarchy view serve the purpose of identifying missing concepts (Fig. 10, left). Looping through the concepts that report very low neuron coverage on the non-robust model, we are able to identify, and verify, that a large number of neurons in the non-robust model fail to recognize certain concepts.

#### 7. Usability Study

We interviewed three individuals (E1-E3) from our institute with varying machine learning and visualization experience to assess our interface’s usability. Of the three interviewees, E1 had 5 years of experience in graph machine learning and data mining, while E2 and E3 specialized in data visualization. E1 has working knowledge in visualization, and actively uses visual idioms and charts for learning and reporting purposes. E2, despite less than one year of experience in machine learning, is interested in utilizing AI for visual analytics. E3, with 4 years of experience in machine learning, specializes in explaining deep learning through visual analytics. Overall, all interviewees recognized the importance of model comparison, found the interface well designed and intuitive, while having mixed views on the design of concept hierarchy.

During the walk-through, including CLIP-Dissect and the CAN interface, all interviewees recognized the importance of model comparison. E1 highlighted comparing methods in the expert domain, focusing on model quality and performance. E1 noted challenges in interpreting quality differences, often due to a lack of domain knowledge or visual encoding strategies. E2 shared the experience visualizing neural network weights with heatmaps. E3 emphasized the importance of model selection, expressing interest in

identifying the best model for specific tasks. Overall, interviewees found CLIP-Dissect valuable, with E1 and E3 seeing its potential in concept-level interpretability for guiding model pruning.

All interviewees found the interface intuitive. E1 and E2 particularly found the neuron UMAP view intuitive. During the exploration they selected different regions of the scatter plot, checking concepts and images in the detail view, and summarizing neuron functions. For example, when brushing over the dog neurons, E1 recognized the region as “animals”. E1 also noted the non-expert friendly nature of the interface compared to verbal summaries. Interviewees had mixed views on the concept hierarchy. E1 viewed it as complementary to the neuron view, suggesting enhancements for exploring same-level concepts. E2 raised concerns about the fairness of vertical space allocation in the compact view for deeper concept levels. In contrast, E3 found concept view and neuron coverage views most useful, taking a top-down approach from the most generic concept (“entity”) to specific concepts (e.g. “dogs”) only when necessary. E3 recognized the semantic coverage of neurons and various neuron-concept relationships. Additionally, E3 desired a brush-over feature for the neuron coverage histogram to examine representational sparsity in the concept hierarchy.

## 8. Limitations and Future Directions

As demonstrated in Sec. 6, by utilizing the proposed tool we are able to answer a set of very specific and detailed questions from the ML community. Despite its effectiveness, it is also important to explore and discuss potential limitations, and the mitigation strategy for them. Additionally, we also want to discuss future directions.

**CLIP-based concept detection:** CAN relies on CLIP-based alignment model, namely CLIP-Dissect, to automatically detect and align concepts with neurons. As a result, the quality of findings presented by CAN depends heavily on the accuracy and specificity of concepts as well as how concepts are interpreted by CLIP. While many of the concepts captured by CLIP appear to be logical, we found some quality issues with CLIP and CLIP-Dissect. For example, we found some overly generic concept words reported by CLIP, such as “juvenile”, “female”, and “adult” reported on a lot of animals and birds images. With CLIP-Dissect, we found certain names (e.g., Tyson, Leo, Bailey) highly associated with neurons highly activated by dog pictures. We speculated this was influenced by the training data of CLIP (dog pictures captioned with their pet names). See figures in supplementary material for a sample of concepts captured by CLIP and CLIP-Dissect. Nevertheless, with the ongoing advancements in artificial intelligence research, we anticipate the development of more sophisticated models and concept probing techniques that can better handle ambiguity.

**Sensitivity regarding thresholds:** One important concern when carrying out comparison tasks of vastly different entities through an intermediate representation is whether the mapping from each of their original space to the shared space is comparable and stable. This is particularly true when the number of neurons may differ drastically across architecture and layers. We specifically explored the sensitivity of using different thresholds to determine the type of concepts captured by a given neuron and incorporated various designs to mitigate its impact (see Sec. 4). However, ideally, we may want to develop methods with built-in self-calibration.

**Layer selection:** During exploration, we need to select a layer from each neural network for comparison. This can turn into a non-trivial task when the number of layers is high in both of the networks we intend to compare. However, the same challenge does also exist for other methods as well. For comparison in the activation space using CKA or other high-dimensional space comparison methods, we also need to identify the layer, i.e., where to measure the activation, for comparison. Interestingly, the concept-centric approach we propose can provide more flexibility and guidance, as we can select the layer based on the concepts they capture, which allows us to narrow down the comparison task to more meaningful pairs.

**Future Directions:** We believe the concept-centric approach can provide a fundamentally new perspective on various network-related comparison tasks. As a robust and reliable way to estimate concept captured by neurons is at the heart of the challenge, one important future direction is to improve the reliability of the concept estimation techniques. One avenue for improvement is through adopting more recent developments [DNR\*23] in multimodal models beyond CLIP, as they may provide a better foundation for concept neuron alignment. Additionally, in the current implementation, we use the word itself as our concept set to guide the discovery of concepts. A full word definition likely contains more nuance semantics that a single word can not convey, so we plan to use these descriptions as an input to CLIP-dissect method. This theoretically could provide better concept association. The other area we plan to improve is the readability of the hierarchy concept abstraction. Although the WordNet provides a principal approach for obtaining a hierarchy, the organization at times can compose more instance-level concepts. To make the discovery of trends more streamlined between the two networks in terms of high-level concepts, we plan to refine our current hierarchy and optimize the structure to highlight more distinct high-level concepts.

## 9. Conclusion

We presented CAN, a visual analytic framework that compares neural networks from the perspective of the functionality of individual neurons. To the best of our knowledge, it is the first visualization work that focuses on the concept-level comparison of neuron networks, which provides a unique perspective for understanding the difference among complex models through a more human-accessible medium. From our case studies, we demonstrate the capability of the proposed approach for investigating challenging questions from ML communities. Apart from confirming existing knowledge of the network behavior (Sec. 6.1), we also obtain new insights regarding the difference in concept representations across model architecture as well as in relationship to model robustness. By summarizing the concept-level interpretation of all neurons in the network, we provide ML researchers with a powerful tool for the efficient exploration of a large collection of neurons and concepts at the same time, which facilitates hypothesis generation.

## Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-CONF-764021). The project is supported by LLNL LDRD (23-ERD-029).

## References

- [BCS22] BOGGUST A., CARTER B., SATYANARAYAN A.: Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. In *27th international conference on intelligent user interfaces* (2022), pp. 746–766. 3
- [BJY\*17] BILAL A., JOURABLOO A., YE M., LIU X., REN L.: Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 152–162. 2, 3
- [BKN\*23] BYKOV K., KOPF L., NAKAJIMA S., KLOFT M., HÖHNE M. M.: Labeling neural representations with inverse recognition. In *Thirty-seventh Conference on Neural Information Processing Systems* (2023). 2, 3, 4, 5
- [BZK\*17] BAU D., ZHOU B., KHOSLA A., OLIVA A., TORRALBA A.: Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 6541–6549. 2
- [BZS\*18] BAU D., ZHU J.-Y., STROBELT H., ZHOU B., TENENBAUM J. B., FREEMAN W. T., TORRALBA A.: Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597* (2018). 2
- [DBK\*20] DOSOVITSKIY A., BEYER L., KOLESNIKOV A., WEISSENBERN D., ZHAI X., UNTERTHINER T., DEGHANI M., MINDERER M., HEIGOLD G., GELLY S., ET AL.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020). 7
- [DK17] DODGE S., KARAM L.: A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)* (2017), IEEE, pp. 1–7. 2
- [DNR\*23] DESAI K., NICKEL M., RAJPUROHIT T., JOHNSON J., VEDANTAM S. R.: Hyperbolic image-text representations. In *International Conference on Machine Learning* (2023), PMLR, pp. 7694–7731. 10
- [FV18] FONG R., VEDALDI A.: Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 8730–8738. 2
- [GBYH20] GLEICHER M., BARVE A., YU X., HEIMERL F.: Boxer: Interactive comparison of classifier results. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 181–193. 3
- [GHM\*22] GÖRTLER J., HOHMAN F., MORITZ D., WONGSUPHASAWAT K., REN D., NAIR R., KIRCHNER M., PATEL K.: Neo: Generalizing confusion matrix visualization to hierarchical and multi-output labels. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (2022), pp. 1–13. 3
- [Gle17] GLEICHER M.: Considerations for visualizing comparison. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 413–423. 4
- [GSS14] GOODFELLOW I. J., SHLENS J., SZEGEDY C.: Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014). 9
- [GWZK19] GHORBANI A., WEXLER J., ZOU J. Y., KIM B.: Towards automatic concept-based explanations. *Advances in neural information processing systems* 32 (2019). 3
- [HHS\*22] HOQUE M. N., HE W., SHEKAR A. K., GOU L., REN L.: Visual concept programming: A visual analytics approach to injecting human intelligence at scale. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 74–83. 3
- [HKMG20] HEIMERL F., KRALJ C., MÖLLER T., GLEICHER M.: emb-comp: Visual interactive comparison of vector embeddings. *IEEE Transactions on Visualization and Computer Graphics* 28, 8 (2020), 2953–2969. 3
- [HMD18] HENDRYCKS D., MAZEIKA M., DIETTERICH T.: Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations* (2018). 4
- [HMKB22] HUANG J., MISHRA A., KWON B. C., BRYAN C.: Conceptexplainer: Interactive explanation for deep neural networks from a concept perspective. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 831–841. 3
- [HPRC19] HOHMAN F., PARK H., ROBINSON C., CHAU D. H. P.: Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 1096–1106. 3
- [HSB\*21] HERNANDEZ E., SCHWETTMANN S., BAU D., BAGASHVILI T., TORRALBA A., ANDREAS J.: Natural language descriptions of deep visual features. In *International Conference on Learning Representations* (2021). 2, 3, 5
- [HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778. 4, 7
- [KAKC17] KAHNG M., ANDREWS P. Y., KALRO A., CHAU D. H.: A cti v is: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 88–97. 3
- [KBN\*21] KIELA D., BARTOLO M., NIE Y., KAUSHIK D., GEIGER A., WU Z., VIDGEN B., PRASAD G., SINGH A., RINGSHIA P., ET AL.: Dynabench: Rethinking benchmarking in nlp. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2021), pp. 4110–4124. 2
- [KNLH19] KORNBLITH S., NOROUZI M., LEE H., HINTON G.: Similarity of neural network representations revisited. In *International conference on machine learning* (2019), PMLR, pp. 3519–3529. 2, 3, 4
- [KWG\*18] KIM B., WATTENBERG M., GILMER J., CAI C., WEXLER J., VIEGAS F., ET AL.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning* (2018), PMLR, pp. 2668–2677. 2
- [LSL\*16] LIU M., SHI J., LI Z., LI C., ZHU J., LIU S.: Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 91–100. 9
- [MA20] MU J., ANDREAS J.: Compositional explanations of neurons. *Advances in Neural Information Processing Systems* 33 (2020), 17153–17163. 2
- [MGS\*22] MINDERER M., GRITSENKO A., STONE A., NEUMANN M., WEISSENBERN D., DOSOVITSKIY A., MAHENDRAN A., ARNAB A., DEGHANI M., SHEN Z., ET AL.: Simple open-vocabulary object detection. In *European Conference on Computer Vision* (2022), Springer, pp. 728–755. 3
- [MHSG18] MCINNES L., HEALY J., SAUL N., GROSSBERGER L.: Umap: Uniform manifold approximation and projection. *Journal of Open Source Software* 3, 29 (2018), 861. 7
- [Mil95] MILLER G. A.: Wordnet: A lexical database for english. *Commun. ACM* 38, 11 (nov 1995), 39–41. URL: <https://doi.org/10.1145/219717.219748>, doi:10.1145/219717.219748. 2, 5, 7
- [OW22] OIKARINEN T., WENG T.-W.: Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965* (2022). 2, 3, 4, 5, 6
- [PDD\*21] PARK H., DAS N., DUGGAL R., WRIGHT A. P., SHAIKH O., HOHMAN F., CHAU D. H. P.: Neurocartography: Scalable automatic visual summarization of concepts in deep neural networks. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 813–823. 3
- [PHVG\*17] PEZZOTTI N., HÖLLT T., VAN GEMERT J., LELIEVELDT B. P., EISEMANN E., VILANOVA A.: Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 98–108. 3

- [RAL\*16] REN D., AMERSHI S., LEE B., SUH J., WILLIAMS J. D.: Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 61–70. [2](#), [3](#)
- [RKH\*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language supervision. In *International conference on machine learning* (2021), PMLR, pp. 8748–8763. [2](#), [3](#), [4](#), [5](#)
- [RUK\*21] RAGHU M., UNTERTHINER T., KORNB�ITH S., ZHANG C., DOSOVITSKIY A.: Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems* 34 (2021), 12116–12128. [3](#), [4](#), [8](#)
- [UKE\*20] UTRERA F., KRAVITZ E., ERICHSOHN N. B., KHANNA R., MAHONEY M. W.: Adversarially-trained deep nets transfer better: Illustration on image classification. In *International Conference on Learning Representations* (2020). [4](#), [7](#)
- [XZKM22] XUAN X., ZHANG X., KWON O.-H., MA K.-L.: Vac-cnn: A visual analytics system for comparative studies of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics* 28, 6 (2022), 2326–2337. [3](#)
- [ZF14] ZEILER M. D., FERGUS R.: Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part 1* 13 (2014), Springer, pp. 818–833. [4](#)
- [ZHP\*17] ZENG H., HALEEM H., PLANTAZ X., CAO N., QU H.: Cn-comparator: Comparative analytics of convolutional neural networks. *arXiv preprint arXiv:1710.05285* (2017). [9](#)
- [ZWM\*18] ZHANG J., WANG Y., MOLINO P., LI L., EBERT D. S.: Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 364–373. [2](#), [3](#)
- [ZXS21] ZHAO Z., XU P., SCHEIDEGGER C., REN L.: Human-in-the-loop extraction of interpretable concepts in deep learning models. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 780–790. [3](#)