# NEURFLOW: INTERPRETING NEURAL NETWORKS THROUGH NEURON GROUPS AND FUNCTIONAL INTERACTIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Understanding the inner workings of neural networks is essential for enhancing model performance and interpretability. Current research predominantly focuses on examining the connection between individual neurons and the model's final predictions, which suffers from challenges in interpreting the internal workings of the model, particularly when neurons encode multiple unrelated features. In this paper, we propose a novel framework that transitions the focus from analyzing individual neurons to investigating groups of neurons, shifting the emphasis from neuron-output relationships to the functional interactions between neurons. Our automated framework, NeurFlow, first identifies core neurons and clusters them into groups based on shared functional relationships, enabling a more coherent and interpretable view of the network's internal processes. This approach facilitates the construction of a hierarchical circuit representing neuron interactions across layers, thus improving interpretability while reducing computational costs. Our extensive empirical studies validate the fidelity of our proposed NeurFlow. Additionally, we showcase its utility in practical applications such as image debugging and automatic concept labeling, thereby highlighting its potential to advance the field of neural network explainability.

## 1 INTRODUCTION

The explainable AI (XAI) field has seen significant advancement in understanding the mechanisms of deep neural networks (DNNs). This field emerges from the growing need in decoding the internal representations, in hope of reverse engineering deep models into human interpretable program. Prior works have initiated on breaking down convolutional neural networks (CNNs) into interpretable neurons, understanding the models in the most fundamental units (Nguyen et al., 2016; Zeiler & Fergus, 2014; O'Mahony et al., 2023; Bykov et al., 2024). Extending further, one can examine the relation between neurons to gain insights on how the model works, within one layer (Vu et al., 2022), and between multiple layers (Cammarata et al., 2020). Ultimately, recent works try to generate circuits (Cammarata et al., 2020; Bykov et al., 2024; Wang et al., 2022c; Conmy et al., 2023) that create exhaustive explanations of how features are processed and evolve throughout the model.

The majority of existing methods focuses on individual neurons Oikarinen & Weng (2024b); La Rosa et al. (2023a) and their relationship to the model's final predictions Ghorbani & Zou (2020b); Wang et al. (2022b); Ghorbani & Zou (2020a), while giving less attention to exploring and quantifying the relationships and interactions between neurons across different layers. These approaches are not only constrained by scalability challenges arising from the extensive number of neurons, but it also hinders a comprehensive understanding of underlying mechanisms of DNNs. A notable example is the polysemantic phenomenon Mu & Andreas (2020); O'Mahony et al. (2023); Olah et al. (2020), where a single neuron is activated by several unrelated concepts. This phenomenon complicates the task of associating each neuron with a distinct feature and hampers the interpretation of how a model processes concepts based on the relationships among neurons. Drawing inspiration from human inference, which synthesizes information from a variety of sources, we contend that, in addition to individual neuron encoding multiple concepts (as demonstrated in prior studies O'Mahony et al. (2023); Olah et al. (2020)), groups of neurons within each layer also collectively encode the same concept. Furthermore, the decision-making process in neural networks is

shaped not solely by the interactions between individual neurons, but rather by interactions among neuron groups.

This study seeks to explore the roles and interactions of neuron groups in shaping and developing concepts, enabling the execution of specific tasks. Due to the complex connections between large number of neurons, identifying those functions and there interactions is a daunting task. To overcome this, we demonstrate that for a particular task, only a subset of neurons—referred to as *core concept neurons*—play a crucial role as influential and concept-defining elements in neural networks. These neurons, when deactivated, significantly alter the associated concepts.

Focusing on core concept neurons allows us to view the intricate network in a simplified way, revealing the most important interactions between the groups of neurons. Therefore, we propose NeurFlow framework that (1) identifies core concept neurons, (2) clusters these neurons into groups, and (3) investigates the functions and interactions of these groups. To enhance interpretability, we represent each neuron group by the set of visual features it encodes (i.e., named as neuron group's concept). Focusing on classification models, we construct, for each class of interest, a hierarchical tree in which nodes represent neuron groups (defined by the concepts they encode), and edge weights quantify the interactions between these groups.

Our key contributions are summarized as follows:

i) We introduce an innovative framework that systematically builds a circuit to elucidate the mechanisms by which core concept neuron groups operate and interact to achieve specific tasks. This entire process is automated, necessitating no human intervention or predefined concept labels. To our knowledge, we are the first to employ neuron groups as the fundamental units for explaining the internal workings of deep neural networks.

ii) We perform empirical studies to validate the proposed framework, demonstrating the optimality and fidelity of core concept neurons, and the reliability of interaction weights between core concept neuron groups.

iii) We provide experimental evidence showing that our framework can be applied to various tasks, including image debugging and automatic neuron concept labeling. Specifically, we confirm the biases found by Kim et al. (2024) on ImageNet (Russakovsky et al., 2015), which have not been proven, by masking the core concept neurons related to the biased features.

## 2 Related work

In an effort to understand the inner mechanism of DNNs, several branches of research have emerged:

**Concept based.** Kim et al. (2018) show that a model can be rigorously understood by assigning meaning to the activations, referred to as concept activation vectors. Subsequent works (Ghorbani et al., 2019; Zhang et al., 2021) have explored more complex methods for extracting these meanings, however, the relationships between concepts remain understudied. Fel et al. (2023); Kowal et al. (2024) address this limitation by constructing a graph of concepts with edges that quantify the relations. Their main intention is to see the evolution of concepts throughout the network layers. Nevertheless, they are unable to explain which parts of the model are responsible for these concepts.

**Neuron based.** Nguyen et al. (2016); Mu & Andreas (2020); O'Mahony et al. (2023); Bykov et al. (2024) invest effort in studying the meaning of neurons, in parallel, Vu et al. (2022); Ghorbani & Zou (2020c); Khakzar et al. (2021b) propose different approaches in identifying important neurons to the model output. These researches shed light on the function of individual neurons and their impact on the prediction of the model. Recently, (Cammarata et al., 2020; Bykov et al., 2024; Achtibat et al., 2023) connect the neurons to form circuits that explain the behavior of a model throughout the layers, nevertheless, the circuits are constructed manually. Furthermore, a major limitation of all previous works is that they only analyze one neuron at a time. This approach is prone to the complex nature of neuron, namely polysemantic neurons, where neurons may encode multiple distinct features, making model interpretation via neurons challenging (Cammarata et al., 2020; O'Mahony et al., 2023). Lastly, Wang et al. (2022a); Kalibhat et al. (2023) find the group of neurons that encode the same concept, however, the relations among the groups and the influence of a group on the model's outputs are left unexplored.

**Graph based.** Ren et al. (2023); Zhou et al. (2024) try to approximate the mechanism of a model by considering the causal relations between the inputs and outputs. Another notable method (Zhang

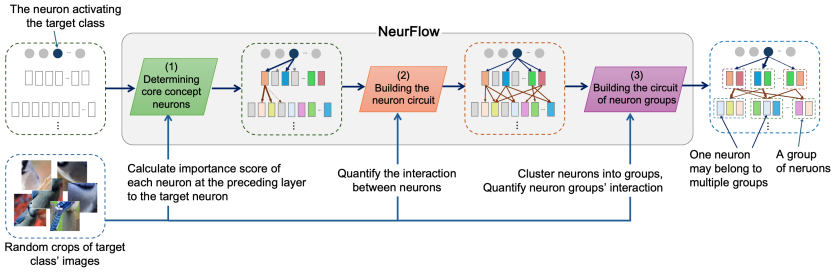


Figure 1: **Workflow of NeurFlow**, consisting of three main components: identifying core concept neurons in each layer, building the neuron circuit, and constructing the circuit of neuron groups.

et al., 2018) generates a graph that highlights what visual features activate a feature map, for multiple layers. While this approach can be modified to form a circuit, the graph lacks meaningful edge weights. Consequently, it cannot quantify the contribution of each CNNs component to others and to the final prediction, unable to explain the inner mechanism (They use and-or-graph (Zhang et al., 2017) to form relations between components. However, unlike circuit, this new graph disregards the original structure of the model, where "concepts" of the first convolution could interact directly with "concepts" of the last convolution.). Subsequent work (Zhang et al., 2019) fixes this issue by building a decision tree to quantify the contribution of each feature map to the final predictions.

Our work aligns the most with explaining neurons and forming circuits. We address the common limitations of manual neuron labeling and circuit construction. We also propose a way to look at neurons not individually but in groups to overcome the common problem of polysemantic neurons. Additionally, we prioritize exploring the interactions between neuron groups across layers rather than focusing solely on the relationship between individual neurons and the model's output. Table 2 in Appendix B provides a comparison of our method with the most relevant existing studies.

# 3 NeurFlow Framework

## 3.1 Problem Formulation

Our goal is to explain the internal mechanisms of deep neural networks (DNNs) by investigating how groups of neurons function and interact to encapsulate concepts, thereby performing a specific task. In particular, we focus on the classification problem, exploring how groups of neurons process visual features to identify a class. Given the exponential number of possible neuron groups, we focus only on core concept neurons. In addition, to facilitate human interpretation, we group these neurons through the common visual features they encode. In essence, we propose a comprehensive framework to address the following questions: *(i) Which neurons play a crucial role in each layer? (ii) How can these neurons be clustered, and what visual features does each neuron group encapsulate? (iii) How do groups of neurons in adjacent layers interact?*

Our problem can be formulated as follows: Given a pretrained classification network $F$ and a dataset $\mathcal{D}_c$ composed of exemplars from a specific class $c$, the goal is to construct a hierarchical tree whose vertices represent groups of core concept neurons in each network layer, and the edges capture the relationships between these groups. Figure 1 illutrates the workflow of our framework which comprises the following key components: (1) identifying core concept neurons (Section 3.3), (2) determining inter-layer relationships among neurons (Section 3.4), (3) clustering the core concept neurons into groups, and analyzing the interactions between these neuron groups (Section 3.5).

## 3.2 Definitions and Notations

In this paper, the term *neuron* refers to either a unit in a linear layer or a feature map in a convolutional layer. As suggested by Cammarata et al. (2020); Bykov et al. (2024); O'Mahony et al. (2023), each neuron is selectively activated by a distinct set of visual features, and by interpreting the neuron as a representation of these features, we can gain insights into the internal representations of a DNN. We refer to these visual features as the concept of the neuron.

In the following definitions, let $a$ represent an arbitrary neuron located in layer $l$ of the pretrained network $F$. In this study, we do not rely on any predefined concepts. Instead, we enhance the original dataset $\mathcal{D}_c$ by cutting it into smaller patches with varying sizes. These patches serve as visual features for probing the model. We refer to this augmented dataset as $\mathcal{D}$, and denote $v$ as an arbitrary element of $\mathcal{D}$.

**Definition 1** (Neuron Concept)**.** The neuron concept $\mathcal{V}_a$ of neuron $a$ is defined as the set of the top-$k$ image patches[1] that most strongly activate neuron $a$. Formally, the neuron concept of $a$ is expressed as $\mathcal{V}_a := \underset{\mathcal{V}\subset\mathcal{D};|\mathcal{V}|=k}{\arg\max} \sum_{v\in\mathcal{V}} \phi_a(v)$, where $\phi_a(v) : \mathbb{D} \to \mathbb{R}$ represents the activation of neuron $a$ for a given input $v \in \mathcal{D}$, and $k$ is a hyperparameter.

An empirical analysis of the impact of $k$ (Appendix D.7) reveals that NeurFlow's performance is relatively insensitive to the selection of $k$.

**Definition 2** (Neuron Concept with Knockout)**.** Let $M$ be the computational graph of the network $F$, $S$ be an arbitrary subset neurons of $M$, and $M \setminus S$ be the sub-graph of $M$ after removing $S$; let $\phi_a^{\overline{S}}$ be the activation of neuron $a$ computed from $M \setminus S$. The neuron concept of $a$ when knocking-out $S$ (denoted as $\mathcal{V}_a^{\overline{S}}$) is defined as $\mathcal{V}_a^{\overline{S}} := \underset{\mathcal{V}\subset\mathcal{D};|\mathcal{V}|=k}{\arg\max} \sum_{v\in\mathcal{V}} \phi_a^{\overline{S}}(v)$.

We hypothesize that for each neuron $a$, only a small subset of neurons from the preceding layer exert the most significant influence on $a$. In particular, knocking out these neurons would lead to a substantial change in the concept associated with $a$. We refer to these neurons as *core concept neurons* and provide a formal definition below.

**Definition 3** (Core Concept Neurons)**.** Given a neuron $a$ at layer $l$, core concept neurons of $a$ (denoted as $\mathbb{S}_a$) is the sub-set of neurons at the previous layer $l-1$ satisfying the following conditions:

$$\mathbb{S}_a := \underset{S\subseteq\mathbb{S};|S|\leq\tau}{\arg\min} \left|\mathcal{V}_a^{\overline{S}} \cap \mathcal{V}_a\right|, \tag{1}$$

where $\mathbb{S}$ is set of all neurons at layer $l-1$ and $\tau$ is a predefined threshold. Intuitively, the core concept neurons for a target neuron $a$ are those that play an important role in defining the concepts represented by $a$. In practice, the value of $\tau$ may vary across the network layers, its impact will be elaborated upon in Sections 4.

In the following, we denote by $\phi^{1,l-1}(v) : \mathbb{D} \to \mathbb{R}^{m\times w\times h}$ the function that maps the input $v$ to the feature maps at the $(l-1)$-th layer of the model, where $m$ represents the number of channels, and $w \times h$ indicates the dimensions of each feature map. Furthermore, we adopt the notation $|.|$ to indicate the cardinality of a set, while $\|.\|$ is employed to represent the absolute value. We summarize all the notations in Table 1 (Appendix A).

### 3.3 Identifying Core Concept Neurons

Given a neuron $a$, we describe our algorithm for identifying its core concept neuron set $\mathbb{S}_a$. This process consists of two main steps: determining $a$'s concept $\mathcal{V}_a$ according to Definition 1, and identifying core concept neurons following Definition 3.

Firstly, we generate a set of image patches $\mathcal{D}$ by augmenting the original dataset $\mathcal{D}_c$, which consists of images that the model classifies as class $c$. Since neurons can detect visual features at different levels of granularity, we divide each image in $\mathcal{D}_c$ into smaller patches using various crop sizes, where smaller patches capture simpler visual features and larger patches represent more complex ones. We subsequently evaluate all items in $\mathcal{D}$ to identify the top-$k$ image patches that induce the highest activation in neuron $a$, thereby constructing $\mathcal{V}_a$.

With $\mathcal{V}_a$ identified, one could determine the core concept neurons through a brute-force search over all possible candidates. However, this naive approach is computationally infeasible. To this end, we define a metric named *importance score* that quantifies the attribution of a neuron $s_i$ to $a$. The importance score can be intuitively seen as integrated gradients (Sundararajan et al. (2017)) of $a$ to $s_i$ calculated across all elements of $\mathcal{V}_a$, calculated as follows:

$$T(a, s_i, \mathcal{V}_a) = \sum_{v\in\mathcal{V}_a} \sum_{\substack{x\in\phi_{s_i}^{1,l-1}(v);\\ y\in\phi_a^{l-1,l}(\phi^{1,l-1}(v))}} x \times \frac{1}{N}\left(\sum_{n=1}^{N} \frac{\partial y(\frac{n}{N}x)}{\partial x}\right), \tag{2}$$

[1] each image patch is a piece cropped from image set.

where $\phi_{s_i}^{1,l-1}$ is the element of $\phi^{1,l-1}$ corresponding to neuron $s_i$, $\phi_a^{l-1,l}$ depicts the function mapping from the activation vector of layer $l-1$ to the activation of neuron $a$, and $N$ is the step size. Utilizing the *importance scores* of all neurons in the preceding layer, the set of core concept neurons is identified by selecting the top $\tau$ neurons that exhibit the highest absolute scores. To justify the use of integrated gradients, we empirically show a strong correlation between the absolute values of $T(a, s_i, \mathcal{V}_a)$ and the change in $a$'s concept after knocking out $s_i$, as demonstrated in Section 4. Additionally, we compare our method with other attribution techniques in Appendix D.1.

### 3.4 Constructing Core Concept Neuron Circuit

For each class of interest $c$, the neuron circuit $\mathcal{H}_c$ is represented as a *hierarchical hypertree*[2], with the root $a_c$ being the neuron in the logit layer (ouput) associated with class $c$. The nodes in each layer of the tree $\mathcal{H}_c$ are the core concept neurons of those in the layer above, and branches connecting a parent node $a$ and its child $s_i \in \mathbb{S}_a$ represents the contributions of $s_i$ to $a$'s concept.

As mentioned in (Cammarata et al., 2020; O'Mahony et al., 2023), neurons often exhibit polysemantic behavior, meaning that a single neuron may encode multiple distinct visual features. In other words, the visual features within a concept $\mathcal{V}_a$ of neuron $a$ may not share the same meaning and can be categorized into distinct groups, which we term *semantic groups*. We hypothesize that each core concept neuron $s_i$ makes a distinct contribution to each semantic group of neuron $a$. To model this relationship, we represent the interaction between $s_i$ and $a$ through multiple connections, where the $j$-th connection reflects $s_i$'s influence on $\mathcal{V}_{a,j}$, the $j$-th semantic group of $a$.

At a conceptual level, the algorithm for constructing the hypertree $\mathcal{H}_c$ proceeds through the following steps: (1) employing our core concept neuron identification algorithm to determine the children of each node in the tree (Section 3.3), (2) clustering the neuron concept of each parent node into semantic groups, and (3) assigning weights to each branch connecting a child node to the semantic groups of its parent. Figure 2 illustrates our algorithm. The complete algorithm for constructing the core concept neuron circuit is presented in Appendix E. We provide a detailed explanation of these steps below.

**Determining semantic groups.** Let the concept $\mathcal{V}_a$ corresponding to $a$ be composed of $k$ elements $\{v_a^1, \ldots, v_a^k\}$. For each visual feature $v_a^i$ $(i = 1, \ldots, k)$, we define its representative vector $r(v_a^i) \in \mathbb{R}^m$ as:

$$r(v_a^i) = \left[\texttt{mean}\left(\phi_1^{1,l-1}(v_a^i)\right), \ldots, \texttt{mean}\left(\phi_m^{1,l-1}(v_a^i)\right)\right], \tag{3}$$

where $\phi_j^{1,l-1}(v_a^i)$ $(j = 1, \ldots, m)$ represents the $j$-th feature map and $\texttt{mean}\left(\phi_j^{1,l-1}(v_a^i)\right)$ denotes the average value across its all elements. Next, we use agglomerative clustering (Murtagh & Legendre, 2014) to divide the set $\{v_a^1, \ldots, v_a^k\}$ into clusters, where the distance between two visual features $v_a^p$, $v_a^q$ is defined by the distance between their corresponding representative vectors $r(v_a^p)$, $r(v_a^q)$. The Silhouette score (Rousseeuw, 1987) is employed to ascertain the optimal number of clusters. The complete procedure is detailed in Algorithm 2.

**Calculating edge weight.** The weight $w(a, s_i, \mathcal{V}_{a,j})$ of the branch connecting a child $s_i$ and its parent $a$'s $j$-th semantic group $\mathcal{V}_{a,j}$ is defined as:

$$w(a, s_i, \mathcal{V}_{a,j}) = \frac{T(a, s_i, \mathcal{V}_{a,j})}{\sum_{s \in \mathbb{S}_a} \|T(a, s, \mathcal{V}_{a,j})\|}, \tag{4}$$

where $T(a, s_i, \mathcal{V}_{a,j})$ is the importance score of $s_i$ to $a$ calculated over $\mathcal{V}_{a,j}$.

### 3.5 Determining Groups of Neurons and Constructing Concept Circuit

This section describes our algorithms to (1) cluster the set of core concept neurons $\mathbb{S}_a = \{s_1, ..., s_k\}$ into distinct groups, (2) identifying the concept associated with each group, and (3) quantifying the interaction between the groups.

**Clustering neurons into groups.** As mentioned in the previous section, a single neuron can encode multiple distinct visual features, while several neurons may also capture the same visual feature

[2]A hypertree is a tree in which each child-parent pair may be connected by multiple edges.

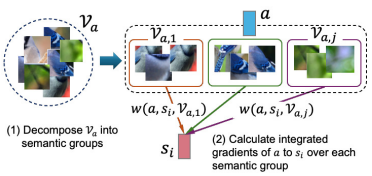


Figure 2: The interaction between a neuron $s_i$ and its parent $a$.

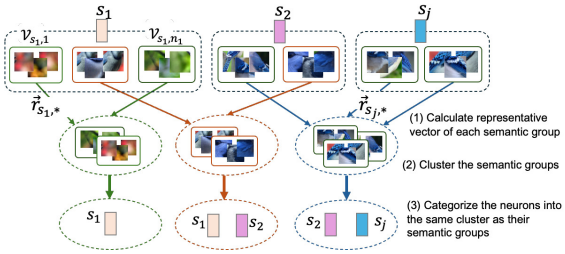


Figure 3: Illustration of our algorithm to determine groups of neurons.

Cammarata et al. (2020). We hypothesize that, due to the polysemantic nature of neurons (Cammarata et al., 2020; O'Mahony et al., 2023), a model may struggle to accurately determine whether a concept is present in an input image by relying on a single neuron. As a result, the model processes visual features not by considering individual neurons in isolation but rather by operating at the level of neuron groups. Intuitively, a group of neurons consists of those that capture similar visual features. This can be interpreted as *two neurons belonging to the same group if they share similar semantic concept groups*.

Building on this intuition, we develop a neuron clustering algorithm based on the semantic groups of each neuron's concept (Figure 3). Specifically, let $\mathcal{V}_{s_i}$ represent the concept of neuron $s_i$ (i.e., the primary visual features it encodes), which can be decomposed into several semantic groups $\{\mathcal{V}_{s_i,1}, ..., \mathcal{V}_{s_i,n_i}\}$ (see Section 3.4), where $n_i$ is the number of semantic groups encoded by $s_i$. For each semantic group $\mathcal{V}_{s_i,j}$, we calculate its representative activation vector $\overrightarrow{r_{s_i,j}}$ by averaging the feature maps of all its visual features, i.e., $\overrightarrow{r_{s_i,j}} := \frac{1}{|\mathcal{V}_{s,j}|}\sum_{v_s \in \mathcal{V}_{s,j}} mean(\phi^{1,l-1}(v_s))$. We then apply the agglomerative clustering algorithm to group the semantic groups $\mathcal{V}_{s_i,j}$ $(i = 1, ..., k; j = 1, ..., n_i)$, where the distance between any two groups $\mathcal{V}_{s_i,u}$ and $\mathcal{V}_{s_j,w}$ is determined by the distance of their respective representative activation vectors, $\overrightarrow{r_{s_i,u}}$ and $\overrightarrow{r_{s_j,w}}$. Finally, we assign neurons $s_1, ..., s_k$ to the same groups based on their semantic concept groups. Specifically, neurons $s_i$ and $s_j$ are clustered together if there exists a semantic group $\mathcal{V}_{s_i,u}$ (of $s_i$) and a semantic group $\mathcal{V}_{s_j,w}$ (of $s_j$) belonging to the same group.

**Finding neuron group concept automatically.** We define the concept associated with a group of neurons as the union of all visual features from the corresponding semantic groups. Specifically, let $\{\mathcal{V}_{G,1}, \ldots, \mathcal{V}_{G,k}\}$ represent the semantic groups categorized into a cluster, with their corresponding neurons $\{s_{G,1}, \ldots, s_{G,k}\}$ grouped together in the same set, denoted as $G$. The concept of this group, denoted as $\mathbb{V}_G$, is then defined as the union of the sets $\{\mathcal{V}_{G,1}, \ldots, \mathcal{V}_{G,k}\}$, i.e., $\mathbb{V}_G := \bigcup_{i=1}^{k} \mathcal{V}_{G,i}$. We leverage a Multimodal LLM to automatically assign labels to the concept, thereby eliminating the need for a predefined labeled concept dictionary. Further details on the design of the prompts are provided in the Appendix G.

**Constructing concept circuit.** For a given class $c$, the concept circuit $\mathcal{C}_c$ is a hierarchical tree where each node represents a neuron group concept (*NGC*), and each edge illustrates the contribution of the child neuron group to its parent. For a node $G$, we denote by $\mathbb{V}_G = \{\mathcal{V}_{G,1}, ..., \mathcal{V}_{G,|\mathbb{V}_G|}\}$ the set of semantic groups associated with $G$, and $\mathbb{S}_G = \{s_{G,1}, ..., s_{G,|\mathbb{S}_G|}\}$ represent the neurons corresponding to the semantic groups in $\mathbb{V}_G$, i.e., $s_{G,j}$ is the core concept neuron possesses the semantic group $\mathcal{V}_{G,j}$ $(j = 1, ..., |\mathbb{V}_G|)$. Let $G_i$ and $G_j$ be a child-parent pair in the tree, then, the relationship between $G_i$ and $G_j$ (quantified by $W(G_i, G_j)$) is represented by two aspects: the number of edges connecting elements of $G_i$ and $G_j$, and the weights of those connecting edges. The more the edges and the higher the edge weights, the stronger the relationship between $G_i$ and $G_j$. Accordingly, we define the weight of branch connecting a child $G_i$ to its parent $G_j$ as sum of the attribution of each neuron in $\mathbb{S}_{G_i}$ with each neuron in $\mathbb{S}_{G_j}$: $W(G_i, G_j) := \sum_{\substack{s_{G_i,q} \in \mathbb{S}_{G_i}; \\ s_{G_j,p} \in \mathbb{S}_{G_j}}} w(s_{G_j,p}, s_{G_i,q}, \mathcal{V}_{G_j,p})$.

## 4 Experimental Evaluation

We perform an extensive empirical study to investigating three aspects, including: optimality of core concept neurons, fidelity of core concept neurons, and fidelity of neuron interaction weights.

Our experiments are performed on ResNet50 (He et al., 2016) and GoogLeNet Szegedy et al. (2015) using the ILSVRC2012 validation set (Russakovsky et al., 2015). The models are pretrained in Pytorch (Paszke et al., 2019), and layer names follow Pytorch's conventions (e.g., *layer4.2* for ResNet50). Unless otherwise specified, the input parameters are $\tau = 16$, $N = 50$, and $k = 50$,

where the top 50 images with the highest activation on the target neuron are considered as its concept. We will release the source code once the paper is published.

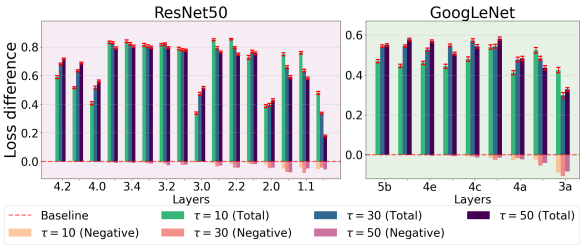


Figure 4: **The difference in losses between core concept neurons and random neuron combinations.** The blue-toned bars represent the average losses, while the pink-toned bars indicate instances where random neuron combinations result in smaller losses compared to core concept neurons.

**Optimality of core concept neurons.** According to Definition 3, the core concept neuron set $\mathbb{S}_a$ for neuron $a$ is the set that minimizes the objective function $\left|\mathcal{V}_a^{\overline{\mathbb{S}_a}} \cap \mathcal{V}_a\right|$ without exceeding the cardinality $\tau$. To evaluate our heuristic solution, we define a loss function $\mathcal{L}(S, a) := \left|\mathcal{V}_a^{\overline{S}} \cap \mathcal{V}_a\right| / |\mathcal{V}_a|$, balancing these two objectives. In this experiment, our objective is to demonstrate that the core concept neurons, $\mathbb{S}_a$, identified by our algorithm are near-optimal. Ideally, a brute-force search over all possible combinations of $\tau$ neurons would be conducted to demonstrate that these combinations yield a higher loss function value compared to $\mathbb{S}_a$. However, such an approach is computationally infeasible due to its prohibitive cost. Consequently, we perform experiments using a large set of randomly selected combinations. Specifically, we use three different values of $\tau$, specifically $10, 30, 50$. For each setting, we randomly select $50$ target neurons (denoted by $a_i$) from $10$ distinct classes (five neurons for each class). For each target neuron $a_i$, we determine its core concept neuron set $\mathbb{S}_{a_i}$ using our algorithm and generate 100 random neuron combinations, with the same cardinality as $\mathbb{S}_{a_i}$, from the preceding layer of $a_i$. In total, the experiments are performed over $15,000$ cases per layer for each model. We compare the loss differences between $\mathbb{S}_{a_i}$ and the random neuron combinations. These average differences along with 99% the confidence intervals are shown in Figure 4. Additionally, we report cases where the random combinations resulted in a smaller loss than our core concept neurons. As observed, the average differences are positive in all cases, indicating that replacing the core concept neurons identified by our algorithm with random ones generally leads to a significant increase in the loss for both models. Furthermore, only a few cases show a random combination achieving a smaller loss than our core concept neurons, and in those instances, the gap is negligible.

**Fidelity of core concept neurons.** We evaluate the impact of the identified core concept neurons on the model's performance by comparing two variants: (1) *Retaining version*–all neurons masked except for the core concept neurons, and (2) *Masking version* version–only the core concept neurons are masked. Intuitively, a higher performance in the *Retaining version* and a lower performance in the *Masking version* would indicate that the core concept neurons play a significant role in the model's performance. We compare the performance of these two versions against models obtained by performing retraining and masking on equal numbers of random neurons. We select 50 random classes and apply the retaining and masking operations at two levels: on a single layer or across multiple layers. In the multi-layer scenario, masking or retaining is applied from the linear classifier down to a specified layer. Figure 5 presents the results for $\tau = 4, 8$, and $16$. The $y$-axis indicates changes in model accuracy, where a value of $1$ implies that masking neurons does not affect predictions. It is evident that masking core concept neurons consistently results in a more pronounced decline in performance compared to masking random neuron combinations. Moreover, the rate of decline in accuracy, moving from higher to lower layers, is considerably steeper for the core concept neurons than for random neurons. The most significant discrepancy occurs at layer *5a* of the GoogLeNet model, where masking core concept neurons at this layer reduces model accuracy to nearly $0$, while masking random neurons has a minimal effect on performance. *Retaining version*, preserving only the core concept neurons allows the model to maintain its performance substantially better than when random neurons are retained.

This experiment also demonstrates that the value of $\tau$ represents a trade-off between the simplicity of the circuit and the comprehensiveness of capturing the core concept neurons. A smaller $\tau$ results in greater instability in the model's performance during the retaining experiment, leading to a more pronounced performance drop. For more discussion on the impacts of $\tau$, please refer to Appendix D.6. The most substantial disparity is observed in the multi-layer scenario of ResNet50, where retaining $16$ core concept neurons nearly preserves the model's full accuracy (close to $1$), whereas retaining $16$ random neurons reduces accuracy to nearly $0$.

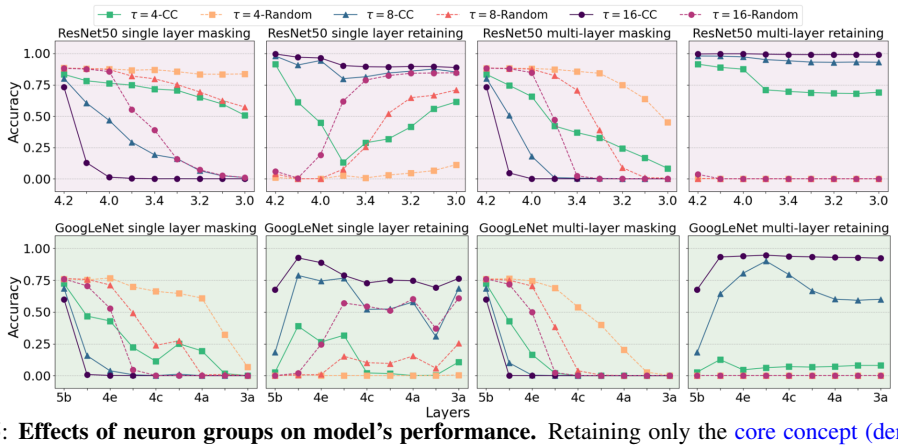


Figure 5: **Effects of neuron groups on model's performance.** Retaining only the core concept (denoted as CC) neurons preserves high accuracy, whereas masking them leads to a significant drop in performance. In contrast, random neuron combinations show the opposite trend.

We conduct an experiment to show that adding non-core neurons to the concept core neurons set identified by NeurFlow has minimal impact the model's performance. Specifically, we perform the Fidelity experiment with $\tau = 16$, incorporating $50\%$ more non-core neurons (i.e., those that are not concept core neurons), and evaluated their impact on model accuracy. These neurons were selected greedily, prioritizing those with the highest scores as ranked by NeuronMCT Khakzar et al. (2021a). The results in Figure 17 (Appendix D.6) indicate that for ResNet-50, adding non-core neurons had little to no effect on improving model performance, confirming that when $\tau$ is sufficiently large, our algorithm ensures completeness.

**Fidelity of neuron interaction weights.** The edge weight representing the interaction between core concept neurons (or groups of core concept neurons) is defined using Integrated Gradients (IG) (Definition 3). Without the ground truth, we evaluate the fidelity of edge weights based on the following rationale: if the weights assigned by our definition are meaningful, they should accurately rank the importance of neurons in the preceding layer in detecting the concept represented by a target neuron in the subsequent layer. We demonstrate that our IG-based scores exhibit a strong correlation with the loss, not only for single neuron setup but also for groups of neurons. Specifically, we randomly select 10 target neurons from 10 distinct classes (denoted as $a_i$, where $i = 1, \ldots, 10$). For each target neuron $a_i$, we generate random combinations of neurons from the preceding layer. We then measure the correlation between the losses caused by these random neuron combinations and the sum of the absolute values of their IG-based importance scores with respect to $a_i$. The experiments are conducted using $500$ neuron combinations, with cardinality ($\tau$) varying from $1$ to $50$. Figure 6 presents the average correlation across all combinations. The results indicate that for $\tau < 50$, IG-based scores maintain a high correlation across all layers.

Notably, for $\tau = 1$, the correlation consistently exceeds $0.6$ in both models, and up to almost perfect correlations for several layers in ResNet50. While the correlation diminishes as $\tau$ increases, our focus is on a small subset of core concept; thus, for a sparse sub-graph of core concept neurons, these results are considered satisfactory. We further compare our defined IG-based score with other attribution methods, including the one used in Vu et al. (2022), in the Appendix D.1.

**Quantitative comparison of NeurFlow with existing approaches.** While our approach focuses on identifying core concept neurons relative to a specific target neuron, we demonstrate that the neurons identified by our method also significantly influence the model's final output. To validate this, we analyzed the overlap between our core concept neurons and the critical neurons identified by Vu et al. (2022), and NeuronMCT (Khakzar et al., 2021a). The $F_1$ scores for these overlaps are presented in Table 3 (Appendix D.4). The results indicate that NeurFlow identifies core concept neurons largely similar to those found by NeuronMCT, even though it does not explicitly find critical neurons to the model's output. Additionally, we compare our approach for identifying core concept neurons for a specific target neuron with the method proposed in Cammarata et al. (2020). In their approach, neurons are ranked based on the top $L_2$ weights connected

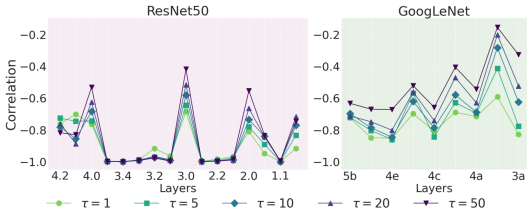


Figure 6: Correlation between loss and our defined IG-based importance scores.

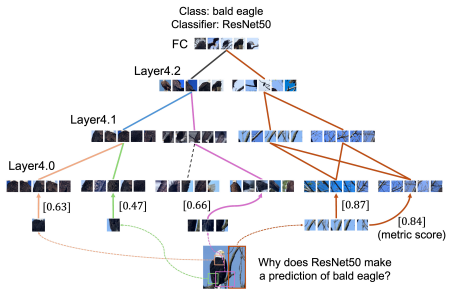


Figure 7: **Using NeurFlow to reveal the reason behind model's prediction.** The top concepts can be traced throughout the circuit.

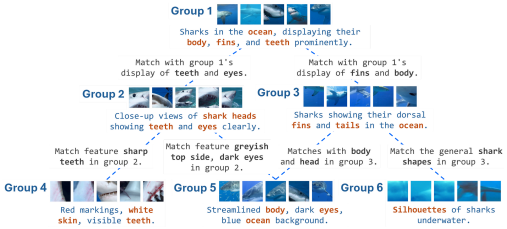


Figure 8: Demonstration for automatically labelling and explaining the relation of NGCs on class "great white shark" using GPT4-o (OpenAI, 2024). The captions and the names of the NGCs are highlighted in blue, while the relations are in black.

to the target neuron. Details of this experiment can be found in Section D.5 (Appendix D.5). The results, summarized in Table 4 (Appendix), show that our method is more effective in identifying core concept neurons.

## 5 Applications

We outlines some applications of NeurFlow. We hypothesize that, as one neuron can have multiple meanings, a DNN looks at a group of neurons rather than individually to determine the exact features of the input. Hence, we propose a metric that assesses a model's confidence in determining whether the input contains a specific visual feature. For a group $G$ with core concept neurons $\mathbb{S}_G = \{s_{G,1}, \ldots, s_{G,|\mathbb{S}_G|}\}$, the metric denoted as $M(v, \mathbb{S}_G, \mathcal{D}) = \exp(\frac{1}{|\mathbb{S}_G|} \sum_{s \in \mathbb{S}_G} \log(\|\phi_s(v)/\max(\phi_s, \mathcal{D}))\|)$, where $v \in \mathcal{D}$ and $\max(\phi_s, \mathcal{D})$ is the highest value of activation of neuron $s \in \mathbb{S}_G$ on dataset $\mathcal{D}$. This returns high score when all neurons in $G$ have high activation (indicating high confidence), while resulting in almost zero if any neuron in the group has low activation (indicating low confidence). We can use this metric to determine how similar the features in the input image are to the predetermined neuron groups concept. The specific setup can be found in the Appendix F. Figures 7 and 9 demonstrate the usage of the metric and the concept circuit. We use the term *NGC* to denote the concept of a neuron group.

### 5.1 Image Debugging

We aim to use the concept circuit to identify concepts contributing to false prediction, which we call *image debugging*. If a concept contributes to a class when it should not, we say that the prediction (or equivalently, the model) is *biased* by that concept. Kim et al. (2024) propose a framework for detecting biases in a vision model by generating captions for the predicted images and tracking the common keywords found in the captions. With this method, they concluded that the pretrained ResNet50 is biased by "flower pedals" in the class "bee". However, correlational features do not imply causation and can lead to misjudgments. We verify and enhance the causality of their claim by examining the concept circuit of class "bee", and conducting experiments on the probabilities of the final predictions with and without neurons that related to "flowers". Additionally, we discover that the model also suffers from "green background" bias (resemble "leaves"), which is not mentioned in Kim et al. (2024).

Figure 9 shows the process of debugging false positive images. Three different concepts are presented in *layer4.2* of ResNet50, representing "pink pedals", "green background", and "bee" respectively (we choose this layer as it has a small set of NGCs, however, our following experiment is consistent for multiple layers and with different classes). We discover that most of the false positive images have high metric score for "pedal" and "green background". To further verify the impact of these biased features, we mask all neurons in the groups of the respective concepts and find that the probability of the predictions are distorted drastically (and predictions is no longer "bee"), as opposed to masking random neurons, which yield negligible changes. This implies the dependence on the biased concept. *But how do we know that the groups reflect the respective visual features?* If these groups indeed represent the visual features, then masking them should hinder the classification probability for images that include those features. We highlight the top images that have the largest decrease in the value of the logit neuron (corresponding to class "bee") on both validation set of the target class and augmented dataset (see Section 3.3). As shown in Figure 9, this process indeed yields the images that contain the respective features.

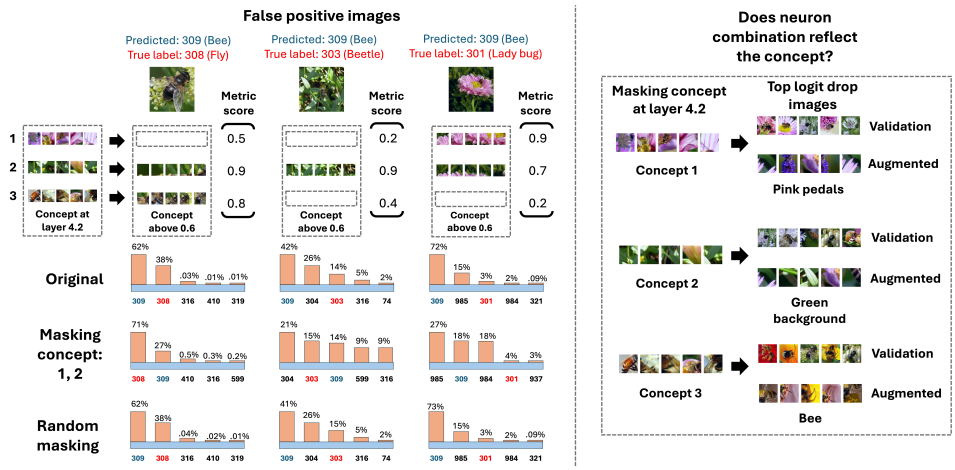


Figure 9: (left) The metric scores of false positive images for each concept in *layer4.2* of ResNet50. (right) Showing the images that have the greatest drop in the activation of the logit neuron when masking each group concept. Verifying that the neuron groups indeed reflect the concepts.

To demonstrate how NeurFlow's findings differ from those of existing methods, we conduct a qualitative experiment comparing the core concept neurons identified by NeurFlow with those identified by NeuCEPT Vu et al. (2022). Detailed information about this experiment is provided in Appendix D.3. Our observations indicate that NeurFlow identifies concepts more closely resembling the original images. Additionally, the top logit drop images identified by NeurFlow align better with the representative examples of the corresponding concepts. Moreover, masking the core concept neuron groups identified by NeurFlow resulted in more significant changes to prediction probabilities while utilizing fewer neurons compared to the groups identified by NeuCEPT.

### 5.2 Automatic identification of layer-by-layer relations

While automatically discovering concepts from inner representation has been a prominent field of research (Fel et al., 2023), automatically explaining the resulting concepts is often ignored, relying on manual annotations. Bykov et al. (2024) utilize label description in ImageNet dataset to generate caption for neurons, however, these annotations is limited and can not be used to label low level concepts. Drawing inspiration from Hoang-Xuan et al. (2024); Kalibhat et al. (2023), we go one step further and not only use MLLM to label the (group of) neurons but also explain the relations between them in consecutive layers. Thus, we show the prospect of completing the whole picture of abstracting and explaining the inner representation in a systematic manner.

Specifically, for two consecutive layers, we ask MLLM to describe the common visual features in a NGC, then matching with those of the top NGC (with the highest weights) at the preceding layer. This can be done iteratively throughout the concept circuit, generating a comprehensive explanation without human effort. We use a popular technique (Wei et al., 2022) to guide GPT4-o (OpenAI, 2024) step by step in captioning and in visual feature matching. Figure 8 shows an example of applying this technique to concept circuit of class "great white shark". We observe that MLLM can correctly identify the common visual features within exemplary images of NGCs. Furthermore, MLLM is able to match the features from lower level NGCs to those at higher level, detailing formation of new features, showing the potential of explaining in automation, capturing the gradual process of constructing the output of the model. The prompt used in this experiment is available in Appendix G.

## 6 Conclusion

We introduced NeurFlow, a framework that systematically elucidates the function and interactions of neuron groups within neural networks. By focusing on the most important neurons, we revealed relationships between neuron groups, which are often obscured by the inherent complexity of neural network structures. Furthermore, we fully automated the processes of identifying, interpreting, and annotating neuron group circuits using multimodal large language models (MLLMs). Our method aims to provide a more efficient and comprehensive approach to the automated interpretation of neural activity. Through rigorous experimentation, we validated the optimality and fidelity of the proposed framework. Additionally, we demonstrated the applicability of NeurFlow across a variety of domains, including image debugging and automatic concept labeling.

## REFERENCES


Reduan Achtibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5(9):1006–1019, 2023.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Kirill Bykov, Laura Kopf, Shinichi Nakajima, Marius Kloft, and Marina Höhne. Labeling neural representations with inverse recognition. *Advances in Neural Information Processing Systems*, 36, 2024.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: Circuits. *Distill*, 2020. https://distill.pub/2020/circuits.

Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold:'model-x'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(3):551–577, 2018.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2711–2721, 2023.

Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5922–5932. Curran Associates, Inc., 2020a.

Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. *Advances in neural information processing systems*, 33:5922–5932, 2020b.

Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. *Advances in neural information processing systems*, 33:5922–5932, 2020c.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019.

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*, pp. 770–778, 2016.

Nhat Hoang-Xuan, Minh Vu, and My T Thai. Llm-assisted concept discovery: Automatically identifying and explaining neuron functions. *arXiv preprint arXiv:2406.08572*, 2024.

Neha Kalibhat, Shweta Bhardwaj, C Bayan Bruss, Hamed Firooz, Maziar Sanjabi, and Soheil Feizi. Identifying interpretable subspaces in image representations. In *International Conference on Machine Learning*, pp. 15623–15638. PMLR, 2023.

A Khakzar, S Baselizadeh, S Khanduja, C Rupprecht, ST Kim, and N Navab. Neural response interpretation through the lens of critical pathways. in 2021 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13523–13533, 2021a.

Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Christian Rupprecht, Seong Tae Kim, and Nassir Navab. Neural response interpretation through the lens of critical pathways. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13528–13538, 2021b.

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*, pp. 2668–2677. PMLR, 2018.

Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. Discovering and mitigating visual biases through keyword explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11082–11092, 2024.

Matthew Kowal, Richard P Wildes, and Konstantinos G Derpanis. Visual concept connectome (vcc): Open world concept discovery and their interlayer connections in deep models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10895–10905, 2024.

Biagio La Rosa, Leilani Gilpin, and Roberto Capobianco. Towards a fuller understanding of neurons with clustered compositional explanations. In *Advances in Neural Information Processing Systems*, volume 36, pp. 70333–70354. Curran Associates, Inc., 2023a.

Biagio La Rosa, Leilani Gilpin, and Roberto Capobianco. Towards a fuller understanding of neurons with clustered compositional explanations. *Advances in Neural Information Processing Systems*, 36:70333–70354, 2023b.

Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31:274–295, 2014.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

Tuomas Oikarinen and Tsui-Wei Weng. Linear explanations for individual neurons. *arXiv preprint arXiv:2405.06855*, 2024a.

Tuomas Oikarinen and Tsui-Wei Weng. Linear explanations for individual neurons. *arXiv preprint arXiv:2405.06855*, 2024b.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Laura O'Mahony, Vincent Andrearczyk, Henning Müller, and Mara Graziani. Disentangling neuron representations with concept vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3770–3775, 2023.

OpenAI. Gpt4-o. `https://openai.com/index/hello-gpt-4o`, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and quantifying the emergence of sparse concepts in dnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20280–20289, 2023.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL `https://www.sciencedirect.com/science/article/pii/0377042787901257`.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014. URL https://arxiv.org/abs/1312.6034.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Asher Spector and Lucas Janson. Powerful knockoffs via minimizing reconstructability. *Annals of Statistics*, 2021+. To Appear.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Minh N Vu, Truc D Nguyen, and My T Thai. Neucept: Locally discover neural networks' mechanism via critical neurons identification with precision guarantee. *arXiv preprint arXiv:2209.08448*, 2022.

Andong Wang, Wei-Ning Lee, and Xiaojuan Qi. Hint: Hierarchical neuron concept explainer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10254–10264, 2022a.

Andong Wang, Wei-Ning Lee, and Xiaojuan Qi. Hint: Hierarchical neuron concept explainer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10254–10264, 2022b.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. Growing interpretable part graphs on convnets via multi-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6261–6270, 2019.

Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11682–11690, 2021.

Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Explaining generalization power of a dnn using interactive concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17105–17113, 2024.