

## A NOTATIONS

We summarize the notations used in this work in Table 1.

Table 1: Notations summarization.

Notation	Meaning
$a$	A neuron in a model at layer $l + 1$
$S$	A set of neurons at layer $l$
$\mathcal{V}_a$	Concept of $a$ : the top- $k$ highest image patches that activate $a$
$\mathcal{V}_a^S$	Concept of $a$ when knocking out $S$
$\mathcal{V}_{a,j}$	The $j$ -th semantic group of $a$
$\tau$	The number of core concept neurons of $a$
$\mathbb{S}_a$	The set of core concept neurons of $a$
$\phi^{1,l}$	The function that maps from the dataset to the activation at layer $l$ of the model
$T(a, s_i, \mathcal{V}_a)$	The importance score of $s_i \in \mathbb{S}_a$ w.r.t $a$ on $\mathcal{V}_a$
$w(a, s_i, \mathcal{V}_{a,j})$	The normalized importance score of $s_i$ w.r.t $a$ on $\mathcal{V}_{a,j}$
$r(v)$	The activation vector of an input $v$
$\mathcal{V}_{s_i,j}$	The representative activation vector of the $j$ -th semantic group
$G$	A neuron group
$\mathbb{S}_G$	The set of neurons of $G$
$\mathbb{V}_G$	The concept of $G$
$W(G_i, G_j)$	The edge weight between $G_i$ and $G_j$

## B RELATED WORKS SUMMARIZATION

Table 2 compares our proposed method and existing approaches.

Table 2: Comparison of NeurFlow and existing approaches.

Method	Objectives	Level of granularity	Interaction quantification
Vu et al. (2022)	Finding critical neurons to the model’s output	Neuron	N/A
Ghorbani & Zou (2020c)			
Khakzar et al. (2021b)			
O’Mahony et al. (2023)	Individual neuron explanation		
Mu & Andreas (2020)			
La Rosa et al. (2023b)			
Oikarinen & Weng (2024a)			
Mu & Andreas (2020)			
Bykov et al. (2024)			
Kalibhat et al. (2023)			
Wang et al. (2022a)			
	Group of neurons		
Kowal et al. (2024)	Determining concept connectivity	Concept	Concept interaction
NeurFlow (Ours)	Determining groups of neurons’ function and interaction	Group of neurons	Neuron group interaction

## C LIMITATIONS AND DISCUSSION

While we view NeurFlow as a significant step toward understanding the function and interaction of neuron groups, it is not without limitations. Our approach defines the concept of neurons as the top- $k$  most activated visual features, a common practice in the field (O’Mahony et al., 2023; Mu & Andreas, 2020; Nguyen et al., 2016). However, other researchers have broadened this definition to include concepts spanning a wider range of activation patterns (La Rosa et al., 2023b; Oikarinen & Weng, 2024a). This limitation highlights a promising direction for future research: developing more

flexible frameworks that incorporate both top- $k$  activation and more distributed neural activation patterns.

Furthermore, our research primarily focus on CNNs, which follows the main focus of a range of previous works in the field (Cammarata et al., 2020; O’Mahony et al., 2023; Nguyen et al., 2016; Mu & Andreas, 2020). However, we can apply our framework onto different DNN architectures by following several steps: 1) define the granularity level of neurons (i.e. individual units, feature maps, attention heads etc.); 2) iteratively identify the target neuron concept and the core concept neurons; 3) cluster the core concept neurons into groups and construct the concept graph. While exploring the differences in the inner workings of various architectures is valuable, we leave this promising direction for future works.

## D ABLATION STUDIES

### D.1 COMPARISON OF ATTRIBUTION METHODS

In this section, we run an ablation study on different choices of attribution method apart from our integrated gradient (IG) approach, verifying that IG-based score is the most suitable for the quantification of edge weights. We assess four additional common pixel attribution methods, including LRP (Bach et al., 2015), Guided Backpropagation (Springenberg et al., 2014), SmoothGrad (Smilkov et al., 2017), Saliency (Simonyan et al., 2014), Gradient Shap (Lundberg, 2017). Notably, SmoothGrad and Gradient Shap are a follow-up versions of IG. Furthermore, we also evaluate attribution method used in Vu et al. (2022), which also find important neurons and attributing scores to them, referred to as Knockoff (Candes et al., 2018). We run on the same setup as in Section 4 for  $\tau$  ranging from 0 to 50. For easier comparison, we report the mean correlations of all values of  $\tau$ . Figure 11 show the mean correlations across the last 10 layers of ResNet50 (He et al., 2016) and GoogLeNet (Szegedy et al., 2015). The Integrated Gradient consistently yields higher correlations compared to other attribution method, surpassing its follow-up version SmoothGrad, while being comparable with Gradient Shap. Furthermore, Knockoff shows a poor performance in ranking the importance of neurons compared to other attribution methods.

Additionally, we also assess the running time of each method. Specifically, we recorded the run time of each method on 50 images on CPU (we implement Knockoff on KnockPy library (Spector & Janson, 2021+) which does not run on GPU, hence, we evaluate all others on CPU for a fair comparison) across all layers of GoogLeNet. The results in Figure 10 show that IG maintain a small running time compared to the follow-up method (i.e. SmoothGrad and Gradient Shap), while yielding the best correlations among the attribution methods. Hence, we choose IG-based score to assign the edge weights in NeurFlow.

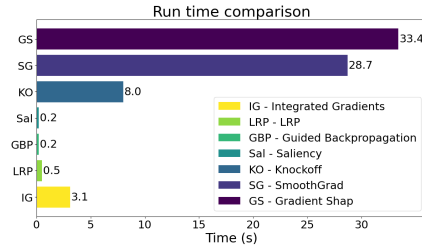


Figure 10: The comparison of average inference time across all layers in GoogLeNet on CPU.

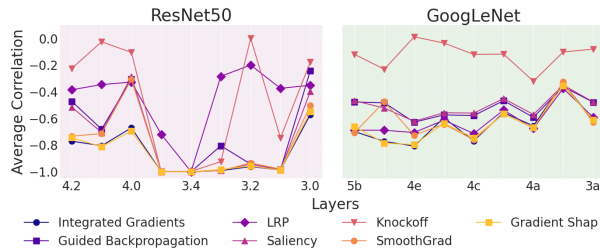


Figure 11: The comparison of different attribution methods for edge weight quantification.

### D.2 NEURON GROUP RELATION WEIGHTS AGGREGATION

In this experiment, we compare our choice of summing the edge weights with averaging the edge weights in forming  $W(G_i, G_j)$  in Section 3.5. Our aim is to verify that: *groups of neurons with higher sum of scores will have higher impact on a target neuron, regardless of the number of neurons in the group.*

We randomly sample 500 groups of neurons of varying sizes, ranging from  $\{1, 5, 10, 20, 50\}$ . For a target neuron in the upper layer, we analyzed the correlation between the loss function (defined in 4 and two metrics: the average edge weights within each group and our original scoring method, which sums the edge weights of neurons in the group. Higher absolute correlation values indicate a more effective scoring method. The results in figure 12 are the average of 10 neurons of different labels in both GoogLeNet and ResNet50.

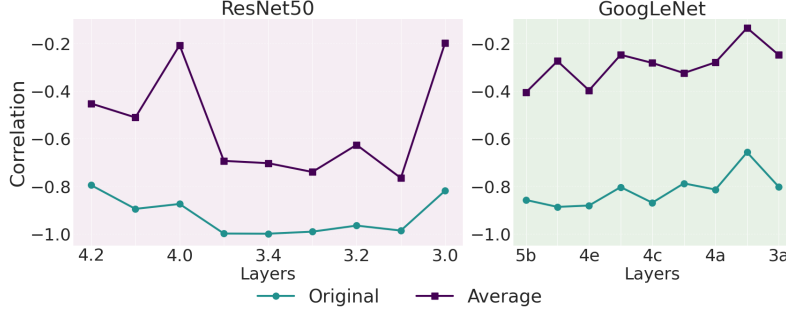


Figure 12: The correlations across 10 layers of our proposed aggregation (denoted as Original) and average aggregation (denoted as Average) on GoogLeNet and ResNet50.

### D.3 QUALITATIVE COMPARISON OF IMAGE DEBUGGING WITH NEUCEPT

We conduct a qualitative experiment to compare the set of critical neurons identified by Vu et al. (2022) (the core concept neuron w.r.t the output logit of the model) and our set in the image debugging experiment. Specifically, following the setups in the experiment in section 5.1, we identify the top  $\tau = 16$  core concept neurons at layer 4.2 of ResNet50 for both methods, which are used to determine the top-2 groups of core concept neurons for a given misclassified image. Groups of neurons were identified following the methodology described in section 3.5, where the groups with the highest metric scores (defined in equation 5) are selected. Furthermore, to quantify the contributions of the selected groups to the model output, we mask all of neurons in each groups and measure the changes of probability of the final predictions. The higher the changes, the more “critical” the groups of neurons. We select three classes, without cherry-picking, namely: Bald Eagle, Great White Shark, and Bee (corresponding to the classes in figure 7, 8, and 9). The results are presented in figure 13, 14, and 15.

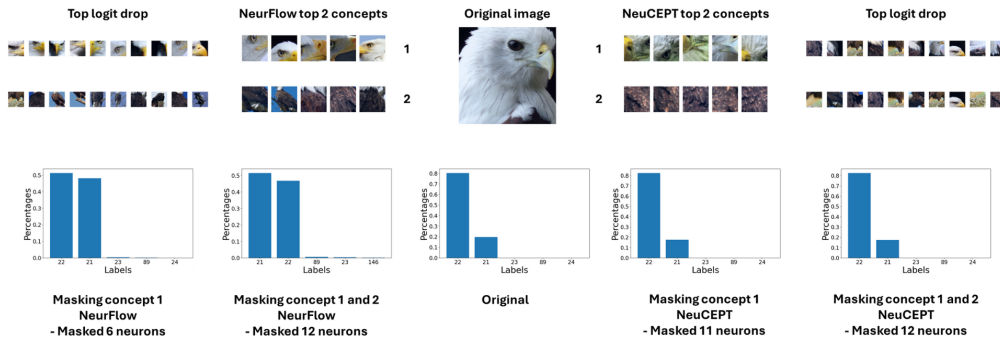


Figure 13: The comparison of the top-2 groups of neurons with the highest metric score of our method and Vu et al. (2022) on class *Bald eagle*. The top logit drop images of NeurFlow are more resemble the original concept (i.e. NeurFlow concept 1 vs NeuCEPT concept 1). And the prediction probability changes when masking our core concept neurons are more significant while masking fewer neurons.

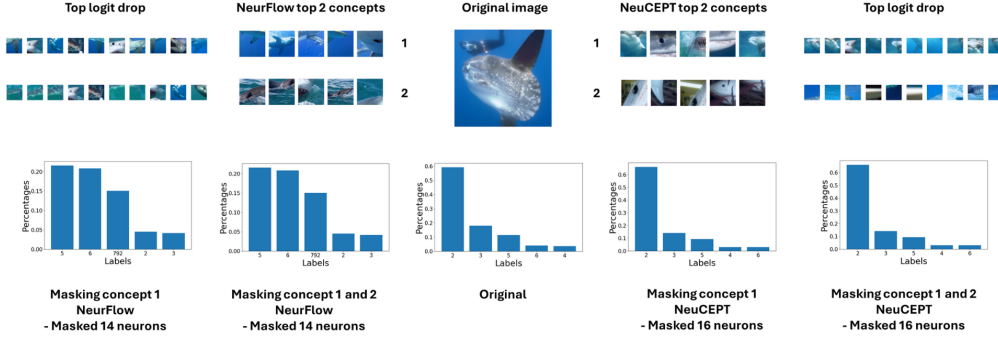


Figure 14: The comparison of the top-2 groups of neurons with the highest metric score of our method and Vu et al. (2022) on class *Great white shark*. The top logit drop images of NeurFlow are more resemble the original concept (i.e. NeurFlow concept 2 vs NeuCEPT concept 2). And the prediction probability changes when masking our core concept neurons are more significant while masking fewer neurons.

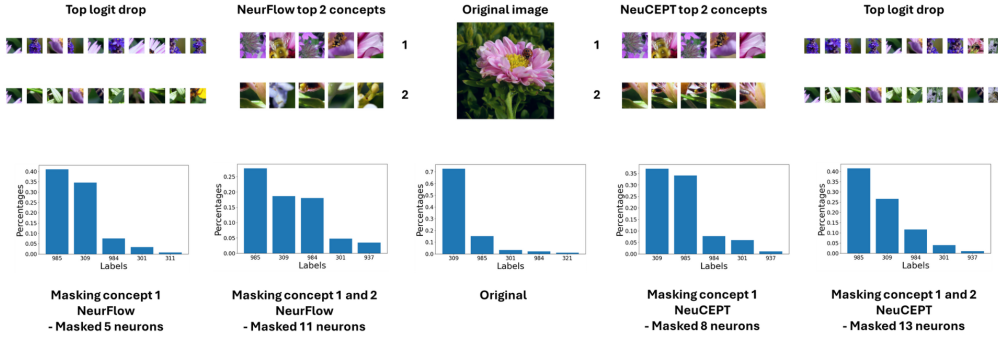


Figure 15: The comparison of the top-2 groups of neurons with the highest metric score of our method and Vu et al. (2022) on class *Bee*. The top logit drop images of both methods are similar to the exemplary image of the concept. And, both methods are able to alter the prediction of the model.

Qualitatively, we observed that our method identified the top-2 concepts more closely resembling the original images. Additionally, our top logit drop images (i.e., "images showing the largest decrease in the target logit value" as described in 5.1) better matched the representative examples of the identified concepts. Furthermore, masking the core concept neuron groups identified by our method resulted in more significant changes to the prediction probabilities, using fewer neurons, compared to the groups identified by NeuCEPT (Vu et al., 2022). For instance, with the labels Bald Eagle and Great White Shark, masking NeuCEPT’s core concept neurons had no effect on prediction probabilities, whereas masking the neurons identified by our method substantially altered the predictions. These findings suggest that our approach identifies more impactful neurons and concepts directly related to the model’s predictions compared to NeuCEPT.

#### D.4 QUANTITATIVE COMPARISON OF CORE CONCEPT NEURONS OF THE MODEL OUTPUT

We run an experiment to further verify: although our method focuses on the set of core concept neurons w.r.t a specific target neuron, our identified neurons also have strong influence to the performance of the model.

Specifically, we evaluate the overlaps between our core concept neurons and the critical neurons (which are specifically designed to find important neurons for the model output) determined by Khakzar et al. (2021a) and Vu et al. (2022), then average the results across all layers of ResNet50 and GoogLeNet of 10 random classes. The numbers of core concept neurons are set to be the same for all three methods. We measure the  $F_1$  scores of the overlaps, which are shown in table 3. The

Table 3: Overlapping ratio of critical neurons between NeuronMCT (Khakzar et al., 2021a), NeuCEPT (Vu et al., 2022), and core concept neurons of NeurFlow

Overlap	NeuronMCT-NeurFlow	NeuronMCT-NeuCEPT	NeurFlow-NeuCEPT
ResNet50	0.72	0.48	0.49
GoogLeNet	0.79	0.55	0.56

Table 4: Average subtraction of the losses. Negative means our loss is better and vice versa

Model	Average Subtraction of the Losses
ResNet50	-0.082
GoogLeNet	-0.013

results imply that NeurFlow contains mostly similar core concept neurons to NeuronMCT while not directly identifying core concept neurons of the output.

#### D.5 QUANTITATIVE COMPARISON OF CORE CONCEPT NEURONS OF A TARGET NEURON

We assess our method of identifying core concept neurons given a specific target neuron with the method used in Cammarata et al. (2020). In Cammarata et al. (2020), neurons are ranked based on the top neurons with the highest  $L_2$  weights connected to the target neuron. Note that this method is not applicable in other experiments since calculating weight magnitude is limited to consecutive layers.

For this comparison, we identify the top  $\tau = 16$  core concept neurons in two consecutive layers (separated by one convolution layer, as per the setup in Cammarata et al. (2020)) using both methods. We then knock out these core concept neurons to observe how the target neuron’s concept is affected. The extent of this change is quantified by the loss function defined in 4, where a lower loss indicates better performance. We randomly selected 100 neurons across 10 different convolution layers from both models and calculated the average difference in losses between the two methods. A negative result indicates our method produces a better loss, while a positive result indicates otherwise.

The results are summarized in table 4. These findings demonstrate that our method is more effective at identifying core concept neurons. Additionally, gradient-based approaches are more versatile, as they can be applied to non-consecutive layers (e.g., ResNet Block 4.2  $\rightarrow$  ResNet Block 4.1 in our experiments), whereas the  $L_2$ -weight-based approach is limited to consecutive layers.

#### D.6 DEPENDENCE ON THE CHOICES OF $\tau$

**The trade-off of the parameter  $\tau$ :** In this experiment, we aim to study the choices of parameter  $\tau$  on the set of core concept neurons of a model. Specifically, in the experiment “Fidelity of core concept neurons”, the choice of  $\tau$  can be seen as a trade-off between simplicity (the number of core concept neurons) and performance (the accuracy of the prediction when retaining only the core concept neurons). However, for  $\tau = 4, 8$  the results are vary across our tested models. We conduct additional experiment to highlight that for sufficiently large  $\tau$ , the results are less dependent on the parameter.

We evaluate on 10 different labels with the same setups as in the experiment “Fidelity of core concept neurons” for  $\tau = 20, 24$ . The results in figure 16 show that with these higher  $\tau$  values, the performance drops of the model become negligible. Furthermore, the differences between retaining for  $\tau = 20$  and  $\tau = 24$  at all layers are minimal, suggesting that the dependence on  $\tau$  decreases as we increase the value.

**Completeness of core concept neurons on the output:** Additionally, we run an experiment to assess the completeness of NeurFlow in identifying the important neurons for the model’s output. By greedily adding 50% more neurons in each layer, of which the neurons are ranked by the importance scores defined in Khakzar et al. (2021a). The higher the scores, the stronger the influence on the

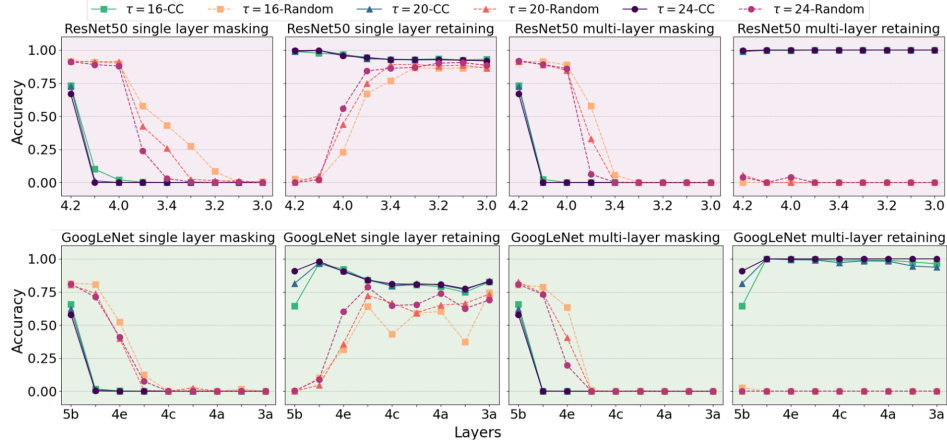


Figure 16: Effects of neuron groups on model’s performance for  $\tau = 16, 20, 24$ . The effect of increasing  $\tau$  are negligible for most of the layers in both models.

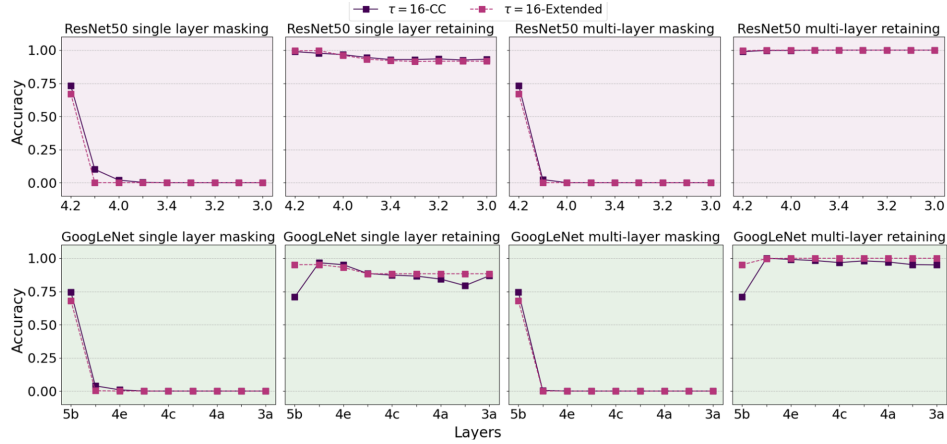


Figure 17: The comparison of the influences on models’ performances of core concept neurons and the extended set of core concept neurons

prediction of the model. We then re-run the “Fidelity of core concept neurons” for  $\tau = 16$  (denoted as “CC”) and its extended version (50% more neurons - denoted as “Extended”). The results in figure 17 show that, for ResNet 50, adding non-core-concept neurons had almost no effect on improving model performance. For GoogleNet, only in the most critical case (where the retaining operation is applied up to layer 5b), adding 50% more non-core-concept nodes led to an improvement in model performance by 25% only at layer 5b in the retaining setup. These results show that when  $\tau$  is sufficiently large, our algorithm ensures completeness.

#### D.7 DEPENDENCE ON THE CHOICES OF $k$

To evaluate the dependence of the results on the choice of  $k$ , we conducted additional experiments with various values of  $k$  and measured the number of core concept neurons overlapping with the baseline setup of  $k = 50$ . Greater overlap indicates less dependence on the choice of  $k$ .

Table 5 summarizes the results with  $\tau = 16$  (i.e., the maximum number of core concept neurons per target neuron is 16) and  $k \in \{30, 40, 50, 60, 70, 90, 110, 130, 150, 170, 190\}$ , evaluated across 50 random neurons. The results show that for all tested values of  $k$ , the overlap ratio is always at least  $14/16$  ( $> 86\%$ ), demonstrating that the results of our proposed algorithm are independent of the choice of  $k$ .

Table 5: The overlap of sets of core concept neurons of different  $k$  compared to the baseline  $k = 50$ 

K	30	40	50	60	70	90	110	130	150	170	190
GoogLeNet	15.0	15.4	<b>16.0</b>	15.5	15.3	15.3	15.0	15.0	14.9	14.9	14.9
ResNet50	14.9	15.6	<b>16.0</b>	15.6	15.3	14.7	14.5	14.3	14.1	14.0	14.0

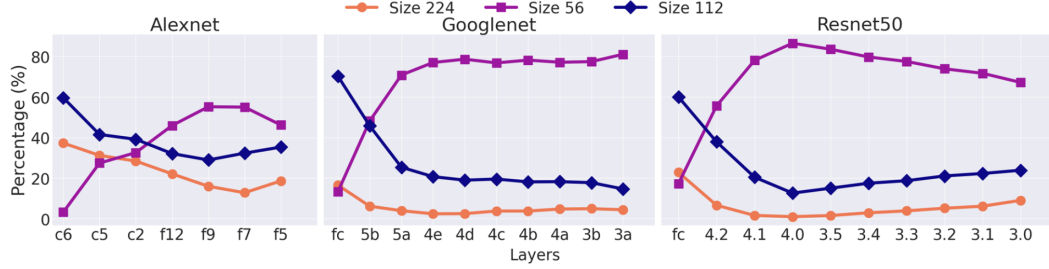


Figure 18: Illustration of the percentages of crop sizes in the concepts of core concept neurons. 50 random classes are assessed for three models and three different crop sizes. The layer names are abbreviated (e.g. “feature.12” to “f12”).

## D.8 MULTIPLE CROP SIZES AUGMENTATION

For a target class  $c$ , our input dataset is created by randomly cropping the images that the DNN classified as class  $c$ , similar to Fel et al. (2023). However, since each neuron can detect feature at different granularity, we crop the images into multiple crop sizes in order to capture features at different levels. Intuitively, small crop sizes indicate low level while large crop sizes indicate high level features. In our experiments, we crop the original images into patches of three different sizes—100%, 50%, and 25% of the original dimensions. The cropping is performed using a sliding window with a 50% overlap, resulting in roughly 2500 patches in total.

Figure 18 shows the percentages of each crop size in the concepts of core concept neurons throughout the networks. As demonstrated, lower layer’s neurons often activated on small crop size images and vice versa. This aligns with the common believe that high level features are detected at the later stages of DNNs. This approach can be improved further by including more complex augmentation methods. However, in this work, our main focus is functional of groups of neurons and their interactions.

## E DETAILED ALGORITHMS

### E.1 IDENTIFYING CORE CONCEPT NEURONS AND CONSTRUCTING NEURON CIRCUIT

Algorithms 1, 2, and 3 provide detailed pseudocode for identifying core concept neurons, determining the semantic groups, and constructing the neuron circuit respectively.

---

#### Algorithm 1 Identifying core concept neurons

---

**Input:** Target neuron  $a$ , dataset  $\mathcal{D}$ , constraint  $\tau$

**Output:** Set of core concept neurons  $S_a$

$\mathcal{V}_a \leftarrow \arg \max_{\mathcal{V} \subset \mathcal{D}; |\mathcal{V}|=k} \sum_{v \in \mathcal{V}} \phi_a(v)$

$T \leftarrow \text{calculate } T(a, s_i, \mathcal{V}_a), \forall s_i \in \mathbb{S}$

$\mathbb{S}_a \leftarrow \text{select the top-}\tau \text{ neurons with the highest } \|T\|$

**return**  $\mathbb{S}_a$

---

**Algorithm 2** Determining semantic groups**Input:** Neuron concept  $\mathcal{V}_a$ **Parameter:** Max number of clusters  $N_{cluster}$ **Output:** Semantic groups  $\mathcal{V}_{a,j}, \forall j$ 


---

```

 $r(v_a^i) \leftarrow$  Calculate the representative vectors  $\forall v_a^i \in \mathcal{V}_a$ 
 $best\_sil\_score \leftarrow -1$ 
 $best\_n \leftarrow$  Initialize
for Number of clusters  $n$  in  $\{2, \dots, N_{cluster}\}$  do
   $\mathcal{V}'_{a,j} \leftarrow$  Agglomerative clustering with  $n$  clusters on  $\{r(v_a^i), \forall v_a^i \in \mathcal{V}_a\}$ 
   $sil\_score \leftarrow$  calculate the Silhouettes score given the results of clustering
  if  $best\_sil\_score < sil\_score$  then
     $best\_sil\_score \leftarrow sil\_score$ 
     $best\_n \leftarrow n$ 
  end if
end for
 $\mathcal{V}_{a,j} \leftarrow$  Agglomerative clustering with  $best\_n$  clusters on  $\{r(v_a^i), \forall v_a^i \in \mathcal{V}_a\}$ 
return  $\mathcal{V}_{a,j}, \forall j \in \{1, \dots, best\_n\}$ 

```

---

**Algorithm 3** Forming neuron circuit**Input:** Logit neuron  $a_c$ , dataset  $\mathcal{D}$ , constraint  $\tau$ **Output:** Neuron circuit  $\mathcal{H}_c$ 


---

```

 $\mathcal{H}_c \leftarrow \{\}; S_L \leftarrow \{a_c\}; \mathcal{H}_c \leftarrow \mathcal{H}_c \cup S_L$ 
for Layer  $l$  in  $\{L-1, \dots, 2, 1\}$  do
   $S_l \leftarrow \{\}$ 
  for Target neuron  $a$  in  $S_{l+1}$  do
     $S_l \leftarrow S_l \cup$  Identify core concept neurons (Alg.1) of  $a$ 
     $\mathcal{V}_{a,j} \leftarrow$  Determine semantic groups (Alg.2),  $\forall j$ 
     $w(s_i, \mathcal{V}_{a,j}) \leftarrow T(a, s_i, \mathcal{V}_{a,j}) / \sum_{s \in \mathbb{S}_a} \|T(a, s, \mathcal{V}_{a,j})\|$ 
  end for
   $\mathcal{H}_c \leftarrow \mathcal{H}_c \cup S_l$ 
end for
return  $\mathcal{H}_c$ 

```

---

**F IMAGE DEBUGGING SETUP**

For an arbitrary input  $v \in \mathcal{D}$ , we want to see which parts of  $v$  are detected by the group of neurons  $G$ . Thus, we crop the image into multiple crops, similar to what we do in Section 3.3. The crops, denoted as  $v_i$  are passed into the model to get the activations, which we can then measure the metric  $M(v_i, \mathbb{S}_G, \mathcal{D}), \forall v_i$ . Then we can set a threshold for each group, so that, the crops with the scores above the threshold can be visualized.

However, since the metric can be greatly affected by only one neuron in the group (i.e one neuron with low activation leads to a low metric score), the metric is prone to outliers. Thus, we only assess the metric on the subset  $\mathbb{S}'_G \subseteq \mathbb{S}_G$ . In practice,  $\mathbb{S}'_G$  contains the top-5 neurons that are closest to the group's center, where each neuron is represented as  $\vec{r}_{s_i, j}$  for a neuron  $s_i \in \mathbb{S}_G$  with the semantic group's index  $j$ . The center of the cluster is the average of all representative vectors, and the distance between a pair of neurons is evaluated using  $l_2$  distance.

**G MLLM PROMPT FOR AUTOMATIC CONCEPT LABELLING**

In this section, we provide our prompts for reproducibility. We employ two types of prompt, which are responsible either captioning the common concepts in the exemplary images of a neuron concept, or describing how a NGC formed from NGCs at the preceding layers. Our prompts include three parts. Firstly, we provide a role for MLLM model, marked as *role description*. Secondly, the *main prompt* is presented where it shows the general instruction for the task that MLLM should do. The



*role description* and *main prompt* is the same for all setups. The last part is the *answer form* where we give specific instruction on how to generate appropriate captions and the template of the answer. The structure of the whole prompts are: *Role description* + *Main prompt* + *Answer form*.

### G.1 ROLE DESCRIPTION AND MAIN PROMPT

**Role descriptions:** “Act as an Image Captioning Language Model.”

**Main prompt:**

“# Core Responsibilities:

- Analyze a set of similar images to identify common features.
- Generate descriptive captions that highlight these common features.
- You must adapt to detect both simple and complex features.

# Important notes:

- You don’t have to generate captions for every image, focus on the common features.
- Outliers exist in the images, you could ignore them if they are not relevant to the common theme.
- You should describe the images with objective visual features, not subjective (like powerful or beautiful or scary etc., because these are only your opinion).
- You should only describe visual features, not the context or the story behind the images.
- You should keep a succinct caption, keep it one or two sentences long, that only describe a few most common features.

# Role Summary:

Your role is to provide accurate and coherent captions for a set of similar images by identifying and describing common features. These features can range from simple elements like edges and colors to complex patterns such as a specific object in a particular setting.”

### G.2 ANSWER FORM

**Answer form for single concept captioning:**

“# Answer form:

- Common features: a list of features
- Caption: your caption in one or two sentences”

**Answer form for describing NGC’s formation:**

# Key note of the input:

- There are many different groups of images, make sure you get the number of groups right.
- Each group of images has a common feature.
- The higher level feature is the first group.
- Other groups are lower level features that combine to form the higher level feature of the first group.

# Key note of the output:

- You should not only focus on the common features of the images but also describe how the features from the lower level groups combine to form the higher-level feature of the first group.
- You should focus on the common features that shared among both the high and low level.

“# Step by step:

- Find the lists of common features in Group 2, ..., N.

- For each feature from those lists: match it with the features in Group 1.
- Some of the features in the lists might have no matches: they might be combined with others to form new features, match the features in Group 1 with some simple combination of the features in Group 2, ..., N (e.g. blue and green → blue-green, multiple curve orientations → a circle, two edges with different orientations → an angle, etc.).
- If you don't find any visual features that match, please don't describe features that is not presented, instead, you can say "There is no matches".
- From the matched features, derive the common features in Group 1.
- Generate caption for Group 1.

*# Answer form:*

- Group 1 Common Features: list of common features
- Group 2 Common Features: list of common features
- ...
- Group N Common Features: list of common features

Feature Evolution:

- Group 2: has feature A - match feature A in Group 1 (for Group 2 to N, if there is no matches, please say "There is no matches")
- ...
- Group N: has feature B - match feature B in Group 1

Caption: one or two sentences capturing the common features and their evolution"