Rethinking Crowd-Sourced Evaluation of Neuron Explanations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Interpreting individual neurons or directions in activations space is an important component of mechanistic interpretability. As such, many algorithms have been proposed to automatically produce neuron explanations, but it is often not clear how reliable these explanations are, or which methods produce the best explanations. This can be measured via crowd-sourced evaluations, but they can often be noisy and expensive, leading to unreliable results. In this paper, we carefully analyze the evaluation pipeline and develop a cost-effective and highly accurate crowdsourced evaluation strategy. In contrast to previous human studies that only rate whether the explanation matches the most highly activating inputs, we estimate whether the explanation describes neuron activations across all inputs. To estimate this effectively, we introduce a novel application of importance sampling to determine which inputs are the most valuable to show to raters, leading to around $30\times$ cost reduction compared to uniform sampling. We also analyze the label noise present in crowd-sourced evaluations and propose a Bayesian method to aggregate multiple ratings leading to a further $\sim 5\times$ reduction in number of ratings required for the same accuracy. Finally, we use these methods to conduct a large-scale study comparing the quality of neuron explanations produced by the most popular methods for two different vision models.

1 Introduction

Despite their transformative capabilities, deep learning models remain fundamentally opaque, which limits their reliability and trustworthiness, especially in high-stakes domains. To address this, the field of mechanistic interpretability aiming to understand the mechanisms inside neural networks has quickly increased in popularity. A key part of mechanistic interpretability is to understand small components of neural networks such as neurons or latents in a sparse autoencoder (SAE), which can be achieved via automated neuron descriptions [3, 19, 14, 20, 21, 7, 17, 26, 1].

To assess the quality of these neuron explanations, researchers often rely on human studies, particularly crowd-sourced evaluations via platforms such as Amazon Mechanical Turk (AMT). However, despite their importance, the design and analysis of these studies have received relatively little attention. Most evaluations rely on subjective measures of whether the explanation matches the highest activating inputs of that neuron. As identified by [22], this corresponds to measuring only *Recall*, and ignores many important factors such as whether the explanation also describes lower activations of that neuron, or whether all inputs matching the explanation actually activate the neuron.

In this paper our goal is to conduct the first crowdsourced evaluation using a more principled evaluation metric: *Correlation coefficient* between neuron activations and explanation concept presence. However evaluating the correlation presents us with new challenges:

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

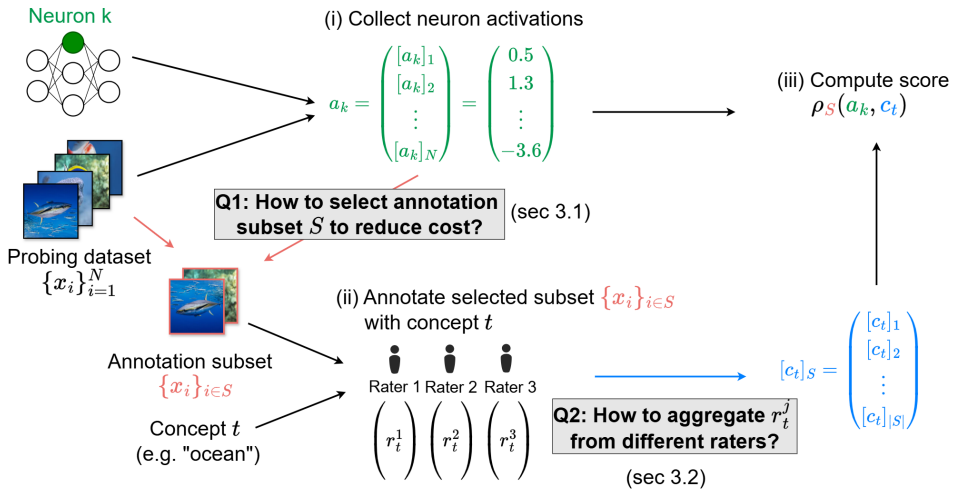


Figure 1: Crowd-sourced evaluation pipeline for neuron explanations. We focus on two critical questions: **Q1:** "*how to efficiently select inputs to annotate?*" and **Q2:** "*how to effectively handle label noise?*". Our proposed solutions are discussed in Section 3 and validated in Section 4.

1. *Cost* and *scalability*. Perfectly evaluating correlation coefficient requires collecting annotations for the concept on *every* input on the dataset. Estimating this for a single (neuron, explanation) pair would cost us around $600 USD with three independent raters, making large scale comparisons with hundreds of neurons and multiple explanation methods impossible. This means we need to effectively estimate the correlation from a subset of annotated data.
2. *Label noise* and *uncertainty*. Crowd-sourced platforms like AMT are quite noisy. Even a small error rate can ruin correlation coefficients when evaluating rare concepts that are only present in a few inputs as there will be more False Positives than True Positives. Increasing the number of raters per input can help reduce uncertainty but also inflates the overall cost.

In this paper we propose novel approaches to solve these problems, and show our methods can reduce the cost of evaluating *correlation coefficient* by approximately $150\times$ while maintaining accuracy, taking our total evaluation costs down substantially from around $\$108,000$ to $\$720$. Our contributions are as follows:

- We conduct the first crowd-sourced neuron explanation evaluation utilizing an evaluation metric that passes the sanity checks proposed by [22], the *correlation coefficient*.
- We show how to effectively utilize Importance Sampling to select most important inputs to show raters, leading to $\sim 30\times$ reduction in labeling cost over uniform sampling.
- We develop a Bayes-based method to aggregate predictions of different raters to deal with noisy labels, further reducing the cost of reaching a certain accuracy by $\sim 5\times$.
- With our proposed methods, we conduct a large scale study to compare different neuron explanation methods, and we discover Linear Explanations [21] overall produce the best vision neuron explanations, outperforming recent LLM-based explanations [1, 26] and other baselines.

2 Related Work

2.1 Automatic Explanation of Vision Model Neurons

Methods based on labeled concept data: Perhaps the first automated method to generate text descriptions for individual neurons in vision models is Network Dissection (ND) [2]. They use the Broden dataset with dense pixel-wise annotations and try to find concepts with high Intersection over Union (IoU) with binarized neuron activations. However it is limited to searching explanations using concepts where dense concept annotated data is available. Compositional Explanations [19] extends Network Dissection to deal with polysemantic neurons, which may activate on multiple unrelated

concepts, by searching for logical compositions of concepts such as "cat OR dog". Clustered Compositional Explanations(CCE)[17] extends this even further, addressing the problem that Compositional Explanations only explains the highest activations of a neuron. CCE divides the activation range of a neuron into 5 buckets, and generates a separate compositional explanation for each. [3] propose a version of Network Dissection utilizing a segmentation model instead of labeled data. Finally INVERT[7] proposes explaining neurons with logical compositions of dataset class(or superclass) labels that maximize AUC, which removes the reliance on Broden but still requires labeled data.

Generative LM based methods: Another popular approach involves generative language model based descriptions. The first such method was MILAN [14], which trains a small generative neural network to describe the most highly activating inputs of a neuron. Describe-and-Dissect(DnD) [1] is a more recent method that utilizes pre-trained language models instead of training their own to generate detailed neuron descriptions based on highest activating inputs. Multimodal Automated Interpretability Agent (MAIA)[26] is another LLM based explanation pipeline where the explaining LLM agent can interact with many tools such as looking at highly activating inputs, generating new images or editing existing ones to generate its description.

CLIP based methods: Finally, papers such as CLIP-Dissect(CD) [20] and FALCON [16] have proposed methods that don't require labeled concept information by relying on supervision from multimodal models such as CLIP [24]. Linear Explanations (LE) [21] proposes to explain polysemantic neurons as a linear combination of concepts, such as "$3 \times$ dog $+ 2 \times$ cat". To learn these explanations, they either utilize class labels from the dataset - LE(label) - or pseudo-labels from SigLIP [27] - LE(SigLIP).

2.2 Human Studies for Neuron Explanations

Previous works [2, 20, 1] have conducted crowdsourced evaluations of the quality of their neuron descriptions. These evaluations show the raters the most highly activating inputs for that neuron, as well one or multiple descriptions, and ask the users to rate how well this description matches the set of highly activating inputs. As pointed out by [22], this evaluation method is flawed and fails to measure whether the description matches lower activations of the neuron, or whether all inputs corresponding to description cause the neuron to fire, and in fact corresponds to only measuring *Recall*. In contrast, in this paper we focus on evaluating the correlation coefficient between the explanation and neuron activations(across all inputs), as this is a principled evaluation metric that passes sanity checks proposed by [22].

[5] and [29] conducted more objective crowd-sourced evaluations of neuron interpretability by having raters predict whether an input is highly activating or not based on a feature visualization or a set of highly activating images for that neuron. However these studies did not study text descriptions of neurons, and their methodology may be biased towards overly generic neuron descriptions and does not pass the sanity checks proposed by [22].

3 Methods

In this section, we address two critical challenges in crowd-sourced evaluation of neuron explanations.

3.1 Challenge 1: Small Sample Size

Problem Formulation. Given a neural network f and a neuron of interest k, one can obtain a neuron explanation t manually or automatically from neuron explanation methods. Following the framework introduced by [22], we use the Pearson correlation between the neuron activation vector a_k and the explanation concept presence vector c_t to evaluate the faithfulness of these neuron explanations. We use Pearson correlation as it passes the sanity checks and performed the best overall in the comparison between evaluation metrics conducted by [22].

Here, $a_k \in \mathbb{R}^{|\mathcal{D}|}$ is the activation vector of a neuron k over all the probing images $x_i \in \mathcal{D}$ and $c_t \in \mathbb{R}^{|\mathcal{D}|}$ is the concept vector to indicate if concept t appears on the probing images. The i-th component $[a_k]_i$ and $[c_t]_i$ can be expressed as: $[a_k]_i = f_k^{0:l}(x_i), \quad [c_t]_i = \mathbb{P}(t|x_i), \ \forall\, i \in \{|\mathcal{D}|\}$

where $f^{0:l}$ denotes the neural network up to l-th layer. See Fig. 1 for an Overview.

Our goal is to evaluate Pearson's correlation coefficient ρ, which can be expressed as:

$$\rho(a_k, c_t) = \frac{1}{|\mathcal{D}|} \frac{\sum_{i \in \mathcal{D}} ([a_k]_i - \mu(a_k)) \cdot ([c_t]_i - \mu(c_t))}{\sigma(a_k)\sigma(c_t)} \tag{1}$$

where $\mu(a_k), \sigma(a_k), \mu(c_t), \sigma(c_t)$ are the mean and standard deviation of the vector a_k and c_t.

The activation vector a_k can be easily evaluated with a forward pass of the neural network. However, the concept vector c_t can be expensive and difficult to obtain, requiring for example crowdsourced annotations. Our problem then becomes: How can we best select a subset of inputs $S \subseteq \{|D|\}$ to annotate corresponding concept labels $\{[c_t]_i\}_{i \in S}$ such that we can accurately estimate $\rho(a_k, c_t)$.

3.1.1 Idea 1: Monte Carlo(Uniform) Sampling

One natural idea to select S is to use Monte-Carlo(uniform) sampling, which randomly selects a batch of samples $\{x_i\}_{i \in S}$ from D uniformly. Notice that Eq (1) can be written in the form of expectation:

$$\rho = \mathbb{E}_{x_i \sim \mathcal{P}} \frac{([a_k]_i - \mu(a_k)) \cdot ([c_t]_i - \mu(c_t))}{\sigma(a_k)\sigma(c_t)}, \tag{2}$$

meaning that we can sample x_i from a uniform distribution $\mathcal{P}$, where $p(x_i) = \frac{1}{|D|}$ for all inputs. We can then view correlation as measuring an expected value of $\bar{a}_{ki} \cdot \bar{c}_{ti}$ where $\bar{a}_{ki} = ([a_k]_i - \mu(a_k))/\sigma(a_k)$, $\bar{c}_{ti} = ([c_t]_i - \mu(c_t)))/\sigma(c_t)$. With Monte-Carlo sampling, the estimator will be $\rho_S = \frac{1}{|\mathcal{S}|} \frac{\sum_{i \in \mathcal{S}} ([a_k]_i - \mu(a_k)) \cdot ([c_t]_i - \mu(c_t))}{\sigma(a_k)\sigma(c_t)}$, where $\mu(c_t)$ and $\sigma(c_t)$ are estimated from the sample. However, this approach leads to a problem: most concepts t are rare compared to the size of probing data $|\mathcal{D}|$, so a small sample $S = \{i_1, i_2, ..., i_{|S|}\}$ from $\mathcal{P}$ is likely to have $[c_t]_{i \in S}$ to be all 0's – meaning that we didn't sample any inputs containing the concept t, making it impossible to accurately estimate ρ. This leads to our second idea, importance sampling.

3.1.2 Idea 2: Importance Sampling

To address this, we propose using importance sampling to estimate ρ in Eq. (2), i.e. using $x_i \sim \mathcal{Q}$ where $\mathcal{Q}$ is a more favorable probability distribution than the original uniform distribution $\mathcal{P}$. Importantly, Importance Sampling does not change the expected value of the function we're evaluation, i.e. $\mathbb{E}_{x \sim \mathcal{P}}[h(x)] = \mathbb{E}_{x \sim \mathcal{Q}}[\frac{p(x)}{q(x)} h(x)]$ for any distribution Q. According to [25](Sec 3.3.2, Theorem 3.12), when estimating the expected value of a function h, i.e. $\mathbb{E}_{x \sim \mathcal{P}}[h(x)]$, the optimal sampling distribution q^* that minimizes the variance of the estimator should have probability density function (pdf) $q^*(x) \propto |h(x)|p(x)$. See Appendix B.3 for formal statement and proof.

Based on the Theorem 1 and Eq (2), we can estimate the ρ by sampling x_i from $\mathcal{Q}$ with pdf $q(x)$:

$$q(x_i) \propto |h(x_i)| \cdot \frac{1}{|\mathcal{D}|}, \quad h(x_i) = \frac{([a_k]_i - \mu(a_k)) \cdot ([c_t]_i - \mu(c_t))}{\sigma(a_k)\sigma(c_t)}. \tag{3}$$

In other words, to minimize the variance of estimator ρ, we should sample x_i more from the probing images that have high product values $\bar{a}_{ki} \cdot \bar{c}_{ti}$ – meaning that the images that have high neuron activation $\bar{a}_{ki}$ and contain concept t.

Since we do not know c_t before running the test (e.g. human study or expensive simulation pipeline), we instead use a cheaper (and less accurate) method to approximate c_t for the purposes of sampling. In particular, we use SigLIP [27] to predict c_t^{siglip}. Thus, our final sampling distribution q is:

$$q(x_i) \propto \frac{1}{|\mathcal{D}|} |[\bar{a}_k]_i \cdot [\bar{c}_t^{siglip}]_i + \epsilon| \tag{4}$$

where $[\bar{c_t}^{\text{siglip}}]_i = \frac{[c_t^{\text{siglip}}]_i - \mu(c_t^{\text{siglip}})}{\sigma(c_t^{\text{siglip}})}$ and $\epsilon = 0.001$ which needed to ensure converge and non-zero sampling probability for all inputs.

Sampling Correction. To get an unbiased estimate of $\mu(c_t)$ and $\sigma(c_t)$ with importance sampling, we need to apply the importance sampling correction sequentially. First, we estimate the mean $\mu_S(c_t)$, and then the standard deviation $\sigma_S(c_t)$ from the samples $x_i \sim \mathcal{Q}$:

$$\mu_S(c_t) = \frac{1}{|S|} \sum_{i \in S} \frac{p(x_i)}{q(x_i)} [c_t]_i, \quad \sigma_S(c_t) = \sqrt{\frac{1}{|S| - 1} \sum_{i \in S} \frac{p(x_i)}{q(x_i)} ([c_t]_i - \mu_S(c_t))^2}. \tag{5}$$

Next, we normalize the inputs: $[\bar{c_t}]_S = \frac{[c_t]_S - \hat{\mu}_S(c_t)}{\hat{\sigma}_S(c_t)}$, $[\bar{a}_k]_S = \frac{[a_k]_S - \mu_S(a_k)}{\sigma_S(a_k)}$, and finally obtain the estimated correlation score of ρ_S as:

$$\rho_S = \frac{1}{|S|} \sum_{i \in S} \frac{p(x_i)}{q(x_i)} [\bar{a}_k]_i \cdot [\bar{c_t}]_i \tag{6}$$

3.2 Challenge 2: Label Noise

Another challenge we face is that often the concept labels c_t we get are noisy. This is in particular a problem when using crowd-sourced platforms such as Amazon Mechanical Turk. To deal with this issue, we can collect m binary ratings $r_{ti}^j, j \in \{m\}$ for each input x_i and aggregate them using different methods. Let $R_{ti} = \{r_{ti}^1, ..., r_{ti}^m\}$ be the set of ratings for a particular (input, concept) pair (x_i, t). Below we describe three different methods for Rating Aggregation, i.e. aggregating the m ratings into a single number $[c_t]_i$:

- **Method 1 - Average:** $[c_t]_i = \frac{\sum_{j=1}^m r_{ti}^j}{m}$
- **Method 2 - Majority Vote:** $[c_t]_i = \begin{cases} 1, & \text{if } \frac{\sum_{j=1}^m r_{ti}^j}{m} > 0.5; \\ 0, & \text{if } \frac{\sum_{j=1}^m r_{ti}^j}{m} \leq 0.5. \end{cases}$

Method 1 and Method 2 are commonly used techniques to aggregate multiple ratings that are simple and intuitive. However, as we show in Section 4.2, we can further improve over these by leveraging Bayes rule to estimate $[c_t]_i$ as $\mathbb{P}([c_t^*]_i = 1 | R_{ti})$. Here we use c_{ti}^* to represent the "ideal" concept value without any labeling noise.

- **Method 3 - Bayes:**

$$[c_t]_i = \frac{\mathbb{P}(R_{ti} | c_{ti}^* = 1) \cdot \mathbb{P}(c_{ti}^* = 1)}{\mathbb{P}(R_{ti} | c_{ti}^* = 1) \cdot \mathbb{P}(c_{ti}^* = 1) + \mathbb{P}(R_{ti} | c_{ti}^* = 0)(1 - \mathbb{P}(c_{ti}^* = 1))} \tag{7}$$

We use Bayes rule to expand the posterior $[c_t]_i = \mathbb{P}(c_{ti}^* = 1 | R_{ti})$ as in Eq (7), and calculate each term as follows:

(I) Likelihood: $\mathbb{P}(R_{ti} | c_{ti}^*)$**.** Suppose each rater makes errors uniformly at random with error rate η, i.e. for any input $\mathbb{P}(r_{ti}^j = c_{ti}^*) = 1 - \eta$, where η is a parameter we can estimate experimentally. Let $\alpha_{ti} = \sum_{j=1}^m r_{ti}^j$, we obtain the likelihood in below equations:

$$\mathbb{P}(R_{ti} | c_{ti}^* = 1) = (1 - \eta)^{\alpha_{ti}} (\eta)^{(m - \alpha_{ti})}, \;\; \mathbb{P}(R_{ti} | c_{ti}^* = 0) = (\eta)^{\alpha_{ti}} (1 - \eta)^{(m - \alpha_{ti})} \tag{8}$$

(II) Prior: $\mathbb{P}(c_{ti}^* = 1)$**.** There are multiple ways of choosing priors, reflecting the confidence of whether a concept exists in an input x_i. In our analysis, we consider two different priors:

- **(a) Uniform Prior:** for all concepts t and all inputs x_i, set $\mathbb{P}(c_{ti}^* = 1) = \beta$
- **(b) SigLIP Prior:** we leverage knowledge from a cheap evaluator, namely SigLIP to initialize the prior and set $\mathbb{P}(c_{ti}^* = 1) = [c_t^{siglip}]_i$

Here β is a hyperparameter, which we set to $\beta = 0.01$. For the SigLIP prior, we clip the prior to be between 0.001 and 0.999 to avoid extreme values which may dominate the final result. Our method with SigLIP prior can be seen as a hybrid evaluation that combines both human and model knowledge.

4 Methodology Testing and Validation

In this section, we describe the 2 settings we used to test the solutions we proposed in Section 3. In setting 1 we utilize simulation with artificial label noise and in Setting 2 we perform a real small scale crowd-source study. This section provides quantitative results and guidance on which methods we should use as cost-effective solutions for large scale crowd-sourced study conducted in Section 5.

Setting 1: Simulating Human Study In this setting, we study the accuracy of our estimator ρ_S by using the ground-truth class(or superclass) labels as our concept vector c_t. We can simulate human raters by only observing a subset of these labels with added label noise. We use c_t^* to denote the ground truth labels, while c_t represents our noisy observation.

We first describe the neurons in layer4 of ResNet-50[13] trained on ImageNet[8] by selecting the concept t_k from ImageNet class and superclass labels that maximizes correlation with neuron k's activations, using ground truth concept labels c_t^* and the full probing dataset $\mathcal{D}$:

$$t_k = \text{argmax}_{t \in \mathcal{C}} \ \rho(a_k, c_t^*) \tag{9}$$

The correlation coefficient of between this concept labels and neuron activations over the entire dataset then serves as our ground truth correlation:

$$\rho_{gt} = \rho(a_k, c_{t_k}^*). \tag{10}$$

We then simulate a crowdsourced study by estimating correlation with a subset of the inputs x_i from the probing data, $i \in S, S \subset \mathcal{D}$ to estimate the correlation:

$$\rho_S = \rho([a_k]_S, [c_t]_S) \tag{11}$$

To simulate rater noise, we randomly flipped some percentage of c_t^* to the opposite label. Here we use a noise rate of η=13%, which is the empirical error rate we measured on Mechanical Turk in Setting 2. For each evaluation (i.e. each data point in Fig 2a), we averaged the estimation error over 10 different trials.

Setting 2: Small Scale Crowd-sourced Study In this setting, we conducted a *real* crowd-sourced evaluation on AMT to annotate the concept label c_t for samples x_i in the probing dataset, on inputs where we also have access to the ground-truth labels c_t^* from the dataset for comparison.

Specifically, we performed a crowdsourced evaluation of 5 neurons in Resnet-50 layer4 with explanations selected from ImageNet classes and superclasses following Eq. (9), and estimated the correlation coefficient as

$$\rho_S = \rho([a_k]_S, [c_t]_S). \tag{12}$$

This setup allows us to evaluate the error rate on Mechanical Turk and the effectiveness of different sampling strategies by measuring the difference between estimated correlation coefficient in Eq. (12) and the "ground truth" correlation coefficient in Eq. (10). We selected neurons with clear unambiguous concepts such as *lizard*, and labeled 300 inputs per neuron with 9 AMT raters labeling each input. To estimate the evaluation accuracy with smaller number of raters/inputs we randomly sampled subsets of the ratings for our plots in Figure 2b and 4 (Note: each point is the average of 1000 random samples of subsets of the same size). We also used this setting estimate error rate of MTurk raters as $\eta = \mathbb{P}(r_{ti}^j \neq [c_t^*]_i)$, giving us an estimate of 13%.

Evaluation Metric: Our main evaluation metric to compare between different sampling strategies is the Relative Correlation Error (RCE) between the ground truth correlation ρ_{gt} and estimated correlation ρ_S in setting 1 and 2:

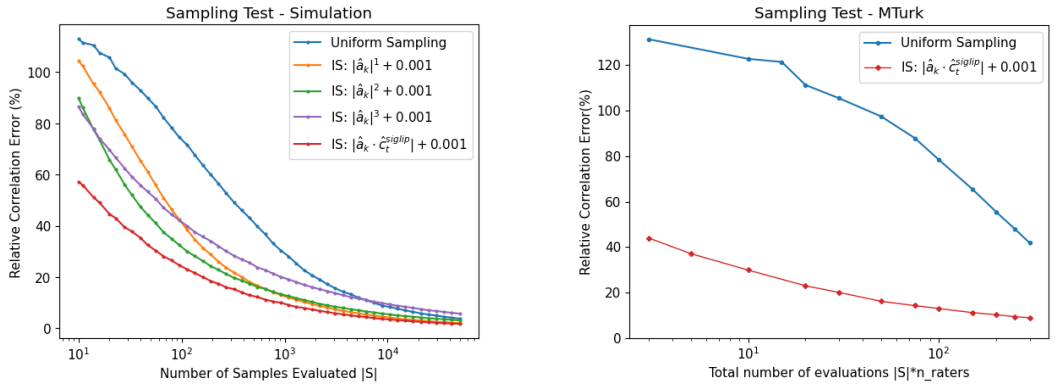
$$RCE = \frac{1}{K} \sum_{k \in K} \left| \frac{\rho_S(a_k, c_{t_k}) - \rho_{gt}(a_k, c_{t_k}^*)}{\rho_{gt}(a_k, c_{t_k}^*)} \right| \tag{13}$$

For example, an RCE of 20% indicates on average our estimated correlation is 20% away from the true correlation value.

4.1 Results: Sampling Method

Figure 2 shows the results of of different sampling strategies across our two settings. To focus only on the effect of sampling, we didn't use any label noise in Fig. 2a. We can see that overall our importance sampling as defined in Eq. (4) performs the best (red line) as lower left represents low error and low cost. Compared to uniform sampling (blue line), we can reach similar correlation error with around 30× less samples, leading to 30× reduction in labeling cost. This holds for both our simulation results (Fig. 2a), as well as with real raters on our MTurk test (Fig. 2b).

In addition to uniform sampling and the optimal importance sampling defined in Eq. 4, we also tested importance sampling where the sampling distribution only depends on neuron activation a_k.



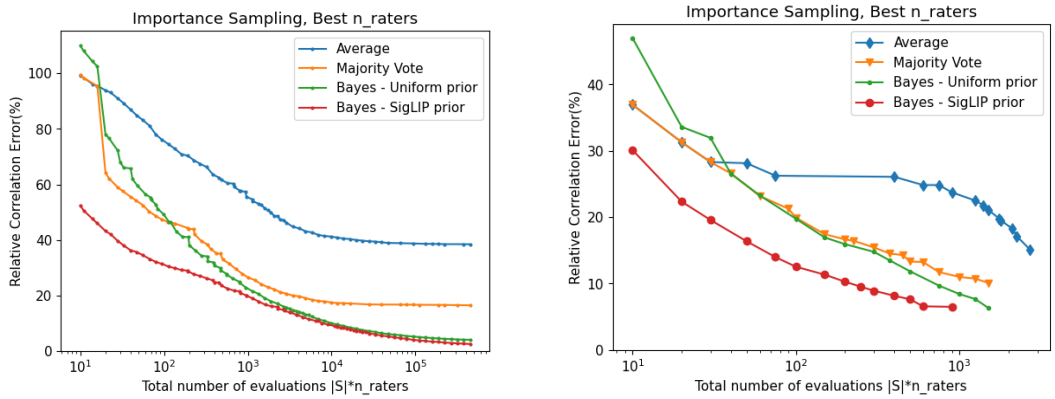
(a) Simulated results in Setting 1

(b) Test results on MTurk in Setting 2

Figure 2: Comparing different sampling strategies in two settings from Sec. 4. We can see Importance Sampling using SigLIP estimates (red) performs significantly better than other methods.

While it performed worse than our SigLIP guided sampling, this sampling also provided a significant improvement over uniform sampling, and based on our experiment using $q(i) \propto [\hat{a}_k]_i^2 + \epsilon$ can be a good alternative if no cheap estimator of c_t is available. *Based on this analysis, we chose to use **importance sampling (Eq. 4)** for our large scale crowd-sourced evaluation in Section 5.*

4.2 Results: Rating Aggregation Method



(a) Simulated results in Setting 1

(b) Test results on MTurk in Setting 2

Figure 3: Comparing rating aggregation strategies in the two settings from Sec. 4. The x-axis represents the cost of evaluations as number of inputs per neuron multiplied by number of raters per input. Each point uses the optimal number of raters that leads to lowest error for that cost.

Next we evaluated the different rating aggregation strategies proposed in section 3.2 to address label noise. As we can see from Figure 3, Method 3b (Bayes with SigLIP prior, red line) resulted in the lowest RCE overall across both settings, reaching similar error at around $5\times$ reduced cost compared to other methods. Method 3a (Bayes with uniform prior, green line) and Method 2 (majority vote, orange line) perform similarly, with Method 3a slightly outperforming with larger number of evaluations. Each line in Figure 3 represents the lowest error achievable by the aggregation method by using the best number of raters for a given cost. *Based on these results, we choose to use **Method 3b: Bayes with SigLIP prior** for rating aggregation in our large scale crowd-sourced evaluation of Section 5.* In addition, we report the results of Bayes with uniform prior to show pure human evaluation results.

We can also use these experiments to decide the optimal parameters for our human study, as discussed in detail in Appendix A.3. Based on our findings and our budget, we choose to use 2 raters per input

and to evaluate 90 inputs per neuron. As we can evaluate 15 inputs in one task for the cost of \$0.06, this evaluation will cost us $\frac{\$0.06}{15} \cdot 90 \cdot 2 = \$0.72$ per (neuron, explanation) pair.

5 Large-Scale Crowdsourced Study

Setup. Our study focuses on neurons in two layers of two different models: Layer4 of ResNet-50 trained on ImageNet, and MLP activations in Layer11 of ViT-B-16 [9] trained on ImageNet. We use the full ImageNet Validation set as probing dataset, see Appendix B.1 for additional details.

5.1 Selecting Methods for the study via Automated Evaluation

Many different methods have been introduced to explain neurons in vision models. Due to limited budget, we are unable to compare all these method on our crowdsourced study, so we first perform an automated evaluation to find the best methods to include in the crowdsourced study. For automated evaluation we use the SigLIP based simulation with correlation scoring introduced by [21], see Appendix B.2 for details.

Explanation Complexity: Some methods explain a single neuron as a combination of concepts. For example [19, 7] use a logical composition of explanations, such as "dog OR cat", while [21] proposes describing neurons as a linear combination of concepts such as "$2.7 \cdot \text{dog} + 1.5 \cdot \text{cat}$" A more complex explanation is typically more accurate, but it is harder to understand, and more expensive to evaluate as it requires labels for each of the concepts involved in the explanation. We use the length l to indicate explanation complexity, where l is the number of unique concepts in the explanation.

Simple Exp. (l=1)	ND [3]	MILAN [14]	CD [20]	INVERT l=1 [7]	DnD [1]	MAIA [26]	LE(label) l=1 [21]	LE(SigLIP) l=1 [21]
RN-50 (Layer4)	0.1242 ± 0.002	0.0920 ± 0.002	0.1904 ± 0.002	0.1867 ± 0.002	0.1534 ± 0.002	0.1396 ± 0.009	0.1793 ± 0.002	**0.2413** ± **0.002**
ViT-B-16 (Layer11 MLP)	0.0335 ± 0.002	0.0194 ± 0.002	0.1849 ± 0.004	0.1343 ± 0.004	0.1049 ± 0.003	0.1497 ± 0.021	0.2704 ± 0.004	**0.2968** ± **0.004**

Table 1: SigLIP based simulation with correlation scoring comparing different simple explanation methods ($l = 1$). We can see Linear Explanation (LE) performs the best even with at length 1.

Complex Exp. (l>1)	Comp Exp [19] l=3	INVERT [7] l=3	CCE [17], l= 5	CCE [17], l=15	LE(label) [21] l=4.37/1.97	LE(SigLIP) [21] l=4.66/1.82
RN-50 (Layer4)	0.1399 ± 0.002	0.2341 ± 0.002	0.0993 ± 0.002	0.1510 ± 0.003	0.2924 ± 0.002	**0.3772** ± **0.002**
ViT-B-16 (Layer11 MLP)	0.0468 ± 0.002	0.1101 ± 0.003	0.0534 ± 0.003	0.0570 ± 0.006	0.3243 ± 0.005	**0.3489** ± **0.005**

Table 2: SigLIP based simulation with correlation scoring comparing complex explanations ($l > 1$). We can see Linear Explanation (LE) overall performs the best.

For fairness we split the explanations into two groups, simple explanations where $l = 1$ and complex explanations with $l > 1$. Table 1 shows comparison of different simple explanations, while Table 2 compares complex explanation methods. Overall we can see Linear Explanations [21] produce the highest correlation scores. Other well performing methods include INVERT [7] and CLIP-Dissect [20], followed by recent language model based explanations MAIA [26] and DnD[1]. We observed that Broden based methods [2, 19, 17] overall performed relatively poorly, likely because their concept sets do not include the relevant higher level concepts such as animal species needed to describe later neurons of ImageNet models.

5.2 Crowdsourced evaluation

Setup. Based on automated evaluation results, we chose to conduct our crowdsourced study on the 5 best performing simple explanations, namely LE(SigLIP)[21], CLIP-Dissect[20], INVERT[7],

MAIA[26] and DnD[1]. We evaluated 100 randomly selected neurons for each of the two models, and we evaluated 90 inputs per neuron with 2 raters each. This gives us a total of 1000 (neuron, explanation) pairs to evaluate for a total cost of $720. Our study was deemed Exempt from IRB approval by the IRB review board at our institution.

User Interface. The main objective of our crowdsourced study is to have users annotate which inputs have concept t present. We do this by presenting raters with explanation t and 15 inputs per task, and asking them to select inputs where concept is present. The full study interface is shown in Figure 8.

5.2.1 Results

Model	Aggregation	CD [20]	INVERT l=1 [7]	DnD [1]	MAIA [26]	LE(SigLIP) l=1 [21]
RN50 (Layer4)	Bayes (Uniform Prior)	0.1500 ± 0.013	0.1258 ± 0.009	0.1576 ± 0.013	0.1215 ± 0.013	**0.1783 ± 0.011**
	Bayes (SigLIP Prior)	0.1817 ± 0.010	0.1637 ± 0.006	0.1661 ± 0.011	0.1309 ± 0.009	**0.2107 ± 0.010**
ViT-B-16 (Layer11 mlp)	Bayes (Uniform Prior)	0.1230 ± 0.016	0.0770 ± 0.016	0.0670 ± 0.015	0.0938 ± 0.017	**0.1513 ± 0.018**
	Bayes (SigLIP Prior)	0.1860 ± 0.020	0.1210 ± 0.021	0.0895 ± 0.017	0.1516 ± 0.021	**0.2382 ± 0.022**
Avg:		0.1601	0.1219	0.1201	0.1245	**0.1946**

Table 3: Results of our large scale MTurk evaluation. Each row represents average correlation across 100 random neurons, as well as the standard error of the mean.

The overall results from the crowdsourced evaluation are shown in Table 3. In Figure 6, we show some example descriptions and their scores. We can see Linear Explanation performs clearly the best overall despite being limited to using a predefined concept set. We think this is likely because it is the only explanation method tested that is optimized to explain the entire range of activations instead of focusing only on highest activations. The second best performing method was CLIP-Dissect [20] while the other methods performed roughly equally with some variance based on the model evaluated. It is somewhat surprising that recent methods based on large language models [1, 26] did not outperform simpler baselines despite being able to provide very accurate and complex description in some cases. We think this is caused by a few reasons:

1. Focus on highly activating inputs only, leading to overly specific explanations that do not describe lower activations.
2. Inconsistency. While all 5 methods mostly produce relevant descriptions, more complex methods such as those relying on LLMs often have more variance in their description quality, leading to poor explanations for some neurons.

Finally, overall we see that methods that perform better on automated SigLIP scoring (Table 1) also performed well on the human study (Table 3). While correlation scores decreased overall (perhaps due to label noise), the order between methods remained stable, highlighting the reliability of SigLIP based explanation evaluation.

6 Conclusion

In this paper, we introduced new methods to conduct a principled crowdsourced evaluation of neuron explanations. First, we proposed to use importance sampling to determine which inputs to show raters, reducing our labeling cost by $\sim 30\times$. Second, we introduced a new Bayes rule based method to aggregate multiple ratings, which further reduces the amount of ratings needed to reach a certain level of accuracy by $\sim 5\times$. Finally, using these methods, we were able to conduct a large-scale human study with reasonable cost (lowering cost from estimated $108,000 USD to $720 USD) without sacrificing accuracy – We evaluated 5 most high performing method for explaining vision neurons and discovered that Linear Explanations [21] performed the best overall with a sizeable lead over other methods.

References

[1] Nicholas Bai, Rahul A Iyer, Tuomas Oikarinen, Akshay Kulkarni, and Tsui-Wei Weng. Interpreting neurons in deep vision networks with language models. *TMLR*, 2025.

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017.

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *PNAS*, 2020.

[4] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.

[5] Judy Borowski, Roland Simon Zimmermann, Judith Schepers, Robert Geirhos, Thomas S. A. Wallis, Matthias Bethge, and Wieland Brendel. Exemplary natural images explain {cnn} activations better than state-of-the-art feature visualization. In *International Conference on Learning Representations*, 2021.

[6] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[7] Kirill Bykov, Laura Kopf, Shinichi Nakajima, Marius Kloft, and Marina MC Höhne. Labeling neural representations with inverse recognition. In *NeurIPS*, 2023.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[10] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting the second-order effects of neurons in clip. *arXiv preprint arXiv:2406.04341*, 2024.

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[12] Yoav Gur-Arieh, Roy Mayan, Chen Agassy, Atticus Geiger, and Mor Geva. Enhancing automated interpretability with output-centric feature descriptions. *arXiv preprint arXiv:2501.08319*, 2025.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *ICLR*, 2022.

[15] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024.

[16] Neha Kalibhat, Shweta Bhardwaj, C Bayan Bruss, Hamed Firooz, Maziar Sanjabi, and Soheil Feizi. Identifying interpretable subspaces in image representations. In *ICML*, 2023.

[17] 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, 2023.

[18] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

[19] Jesse Mu and Jacob Andreas. Compositional explanations of neurons. In *NeurIPS*, 2020.

[20] Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. In *ICLR*, 2023.

[21] Tuomas Oikarinen and Tsui-Wei Weng. Linear explanations for individual neurons. In *International Conference on Machine Learning*, 2024.

[22] Tuomas Oikarinen, Ge Yan, and Tsui-Wei Weng. Evaluating neuron explanations: A unified framework with sanity checks. In *International Conference on Machine Learning*, 2025.

[23] OpenAI. GPT-4o system card. `https://openai.com/index/gpt-4o-system-card/`, 2024.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[25] Christian Robert and George Casella. *Monte Carlo Statistical Methods, second edition*. Springer New York, 2004.

[26] Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning*, 2024.

[27] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[28] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[29] Roland S. Zimmermann, Thomas Klein, and Wieland Brendel. Scale alone does not improve mechanistic interpretability in vision models. In *NeurIPS*, 2023.

A Discussion

A.1 Limitations:

Due to time and budget constraints, our crowdsourced evaluation focus on comparing descriptions on a few models/layers e.g. later layer neurons of ImageNet trained models. While we mostly find similar trends between the ResNet and ViT models, different description methods may have advantage on different types of neurons. For example, it is possible that Network Dissection based methods would perform better than they did on our current evaluation if we focus on lower layer neurons or on models trained on Places365 [28], as the labels in the Broden dataset are more suitable for these tasks. Similarly, we think LLM based methods [1, 26] might perform better when describing neurons of a sparse autoencoder [6] as these are more monosemantic and can be better described by highly activating inputs only.

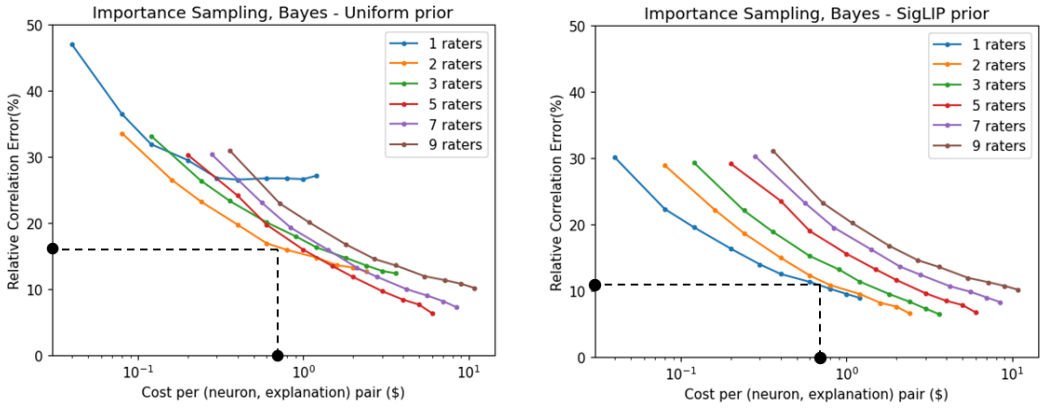
Second, our crowdsourced evaluation relies on Amazon Mechanical Turk workers, who are not experts and often make errors in labeling. While we introduced principled measures to estimate the errors as well as mitigations for error, we cannot improve their domain knowledge, which means a crowdsourced evaluation might favor simpler descriptions over more complex concepts requiring domain knowledge. To compare more complex descriptions or neuron descriptions in a specific domain it may be necessary to recruit domain expert raters.

Finally, our evaluation is focused on evaluating *input-based* neuron explanations that aim to explain the "input $\rightarrow$ neuron activation" function. Some recent work such as [10, 12] instead focus on *output-based* neuron explanations that aim to explain the "neuron activation $\rightarrow$ model output" relationship. Rigorously evaluating these *output-based* explanations will require different methodology and is an interesting problem for future work.

A.2 Broader Impact

This paper is focused on better understanding neural networks via interpretability, and as such we expect it's impact to be largely positive as better understanding of neural networks can help us for example identify failure modes before deploying and enable better control of models. As our focus is in particular on rigorous evaluation of neuron explanations, this can help avoid interpretability illusions or users over-relying on unreliable explanations.

A.3 Crowdsourced Study Design



(a) Aggregation with Method 3: Bayes - Uniform Prior. (b) Aggregation with Method 3: Bayes - SigLIP prior.

Figure 4: Comparing rating aggregation strategies in Setting 2 (Testing on MTurk) described in Sec 4. The x-axis represents the cost of evaluations as number of inputs per neuron multiplied by number of raters per input times cost per input. The lowest curve represents the optimal number of raters for a particular budget.

In this section we discuss how our methodology testing/validation settings described in Section 4 can be used to help design the finer parameters of our study such as how many raters to use per

input. In our current MTurk setup, the cost of obtaining a rating for a single image by a single rater is $\frac{\$0.06}{15} = \$0.004$ as we show 15 inputs per task. We can then plot the expected error as a function of evaluation cost by plotting $n_{inputs} \cdot n_{raters} \cdot \$0.004$ on the x-axis.

In Figure 4, we plot the cost vs expected error rate for different numbers of raters we have per input. For most evaluation budgets, Bayes with uniform prior gives the best results when using 2 raters per input, while Bayes with SigLIP prior gives the best results when using 1 rater. Given our budget, we are aiming for a cost of around \$0.7 per (neuron, explanation pair), and we can see the optimal number of raters and expected error rate for this budget following the black lines on our plot. Since we wish to get good results using both aggregation methods, and at this range the SigLIP sampling gives almost the same error whether it is 1 or 2 raters, we choose to use 2 raters and 90 inputs per neuron for the crowdsourced study, which gives us an expected correlation error of around 16% with Bayes - Uniform Prior and around 12% with Bayes - SigLIP prior based on the MTurk test.

B Additional Details

B.1 Experimental Details

Our evaluation focused on two models/layers:

1. Layer4 (end of residual block 4) of ResNet-50 trained on ImageNet. For simulation we use the neuron activation after average pooling, giving scalar activations, but for methods designed for 2d-activations, activations before average pooling are given as input.
2. MLP-activations in Layer 11 of ViT-B-16 [9] Encoder trained on ImageNet. We focus only on the activations of the CLS-token as this is the last layer and other tokens do not affect the prediction.

We use the full ImageNet validation dataset as $\mathcal{D}$ for the human study for all methods, and for generating explanations unless the method requires a specific dataset for explanation generation such as Broden [2].

For all methods using SigLIP guidance we use the SigLIP-SO400M-14-386 model.

B.1.1 Baseline Implementation Details

For practical purposes we made slight modifications to some baseline methods. The changes are discussed in detail below:

DnD [1]: The original implementation uses GPT-3.5 Turbo through the OpenAI API. Given the high cost of using the API and that recent open-source LLMs have strong performance compared to even closed-source LLMs like GPT-3.5, we replace GPT-3.5 Turbo with Llama 3.1-8B-Instruct [11]. DnD [1] showed that the older Llama 2 model was already better than GPT-3.5 Turbo and on-par with GPT-4 Turbo for neuron description, so our choice of Llama 3.1 does not degrade DnD's performance w.r.t. GPT-3.5 Turbo.

MAIA [26]: Compared to the original implementation, we replace GPT4-vision-preview with the newer GPT4o-2024-08-06 [23] since it has lower API costs and better performance. The method is still quite expensive, costing us approximately \$65 and \$116 to generate descriptions for 100 randomly selected neurons of ResNet-50 Layer4 and ViT-B-16 Layer11 MLP respectively. Note that this cost also includes repeating the experiment for $\sim$10 and $\sim$20 neurons from ResNet and ViT respectively which did not yield any neuron description in the first run. Initially, we also tried open-source LLMs with support for vision inputs (*i.e.* VLMs) like Llama-3.2-11B-Vision-Instruct [11], Llava-OneVision-Qwen2-7B-ov-hf [18], and CogVLM2-Llama3-Chat-19B [15]. However, these VLMs do not work well with MAIA since they fail to generate executable code, forget the image tokens and focus only on the last few text tokens if given a long prompt, and allow only one image input at a time. This is likely because these VLMs are geared towards visual question answering and do not possess the more generalized capabilities of GPT4/4o.

Methods relying on 2d activations: Many methods are designed to explain entire channels of CNNs with 2d activations [2, 19, 14, 17]. For ResNet-50 layer4 we fed the pre-avg pool activations to these methods for proper 2d input. However, for ViT-B-16 last layer only the CLS-token activations affect

the output, and as such we are explaining neurons with scalar activations. In this case, we broadcasted the scalar activations into a 2d-tensor with the same value in all spatial locations. However this is not the intended way to use these methods and may partially explain poorer performance of some methods on ViT-neurons as observed in Tables 1 and 2.

CCE [17]: For Clustered Compositional Explanations, we tested two different versions: the $l = 15$ version corresponds to the default version with length 3 explanations for each of the 5 activation clusters. For the $l = 5$ version we used explanation length=1 with 5 clusters of activations. We also used the implementation of [17] to reproduce the results of Compositional Explanations [19] by setting explanation length=3 and number of clusters=1.

B.1.2 Subset of Neurons

For most methods in Table 1 and 2, we report the average correlation scores across all 2048 neurons for ResNet-50 layer4 and 3072 neurons for ViT-B-16 layer11 mlp. However for certain methods due to high computational and/or API cost we were only able to explain a subset of these neurons and report the average score of these subsets in Tables 1 and 2. We report the results for a subset of neurons for the following methods:

- MAIA [26]: Randomly selected subset of 100 neurons each for both RN50 and ViT-B-16.
- CCE [17] $l = 5$: For ViT-B-16 we used a subset of 1420 neurons. RN50 evaluated on all neurons.
- CCE [17] $l = 15$: RN50: subset of 984 neurons. ViT-B-16: subset of 422 neurons.

All other methods were evaluated on all neurons of each layer.

B.2 Automated Evaluation Details

For our automated evaluation(Sec. 5.1), we use Simulation with Correlation Scoring as described by [21]. This evaluation was originally proposed for language model neurons by [4].

The basic idea of simulation evaluation to use the *explanation* to predict neuron activations on unseen inputs. With correlation scoring we then evaluate the correlation coefficient ρ between the predicted activations s and actual neuron activations a_k on the entire test split of 10,000 inputs as done by [21].

For **simple explanations**, the predicted activation s is simply the presence of concept on that input.

$$s(x_i, t) = [c_t]_i \tag{14}$$

For a **linear explanation**, $E = \{(w_1, t_1), ..., (w_l, t_l)\}$ the predicted activation s is calculated following [21] as:

$$s(x_i, E) = \sum_{w_j, t_j \in E} w_j [c_{t_j}]_i \tag{15}$$

For **compositional explanations** [19], we calculate the predicted activation as follows using probabilistic logic. Different basic logical operators are calculated as:

$$s(x_i, t_1 \text{ AND } t_2) = [c_{t_1}]_i \cdot [c_{t_2}]_i \tag{16}$$

$$s(x_i, t_1 \text{ OR } t_2) = 1 - (1 - [c_{t_1}]_i) \cdot (1 - [c_{t_2}]_i) \tag{17}$$

$$s(x_i, \text{NOT } t) = 1 - [c_t]_i \tag{18}$$

Predictions for more complex compositions are then calculated by iteratively applying these rules.

Clustered Compositional Explanations: CCE [17] explanations are of the form $E = \{(l_1, u_1, F_1), ..., (l_r, u_r, F_r)\}$ where r is the number of activation clusters, and l_j, u_j are the lower

and upper bound of activations for that cluster and F_j is a compositional explanation for activations of that cluster. To predict neuron activation based on this explanation, we use the following formula:

$$s(x_i, E) = \sum_{l_j, u_j, F_j \in E} \frac{l_j + u_j}{2} s(x_i, F_j) \tag{19}$$

This means if the concepts according to the formula of a cluster are present, we predict the neuron's activation will be in the middle of the clusters activation range.

For all automated evaluations we use SigLIP-SO400M-14-386 model to predict c_t following [21].

B.3 Theorem 1

Suppose we are estimating the expected value of function $h(x)$ when $x \sim \mathcal{P}$. Let $\mathcal{X}$ be the support of $\mathcal{P}$.

Theorem 1 ([25], Sec 3.3.2, Theorem 3.12)**.** *For importance sampling with sampling distribution q:*

$$\mathbb{E}_{x\sim\mathcal{P}}[h(x)] = \int_{\mathcal{X}} h(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{|S|}\sum_{i=1}^{|S|}\frac{h(x_i)p(x_i)}{q(x_i)}.$$

The choice of q that minimizes the variance satisfies $q(x) \propto |h(x)|p(x)$.

Proof: Reproduced from [25].

$$Var\left[\frac{h(x)p(x)}{q(x)}\right] = \mathbb{E}_q\left[\left(\frac{h(x)p(x)}{q(x)}\right)^2\right] - \left(\mathbb{E}_q\left[\frac{h(x)p(x)}{q(x)}\right]\right)^2 \tag{20}$$

Since the the second term $\left(\mathbb{E}_q\left[\frac{h(x)p(x)}{q(x)}\right]\right)^2 = \left(\int_{\mathcal{X}} h(x)p(x)dx\right)^2$ does not depend on q, in order to minimize variance we only need to minimize the first term.

From Jensen's inequality it follows that:

$$\mathbb{E}_q\left[\left(\frac{h(x)p(x)}{q(x)}\right)^2\right] \geq \left(\mathbb{E}_q\left[\frac{|h(x)|p(x)}{q(x)}\right]\right)^2 = \left(\int_{\mathcal{X}} |h(x)|p(x)dx\right)^2 \tag{21}$$

Giving us a lower bound for the first term. If we set

$$q(x) = \frac{|h(x)|p(x)}{\int_{\mathcal{X}} |h(z)|p(z)dz} \tag{22}$$

Which is a valid probability distribution, we get:

$$\mathbb{E}_q\left[\left(\frac{h(x)p(x)}{q(x)}\right)^2\right] = \left(\int_{\mathcal{X}} |h(z)|p(z)dz\right)^2 \tag{23}$$

This exactly matches the lower bound, proving that minimum variance is attained by setting

$$q(x) = \frac{|h(x)|p(x)}{\int_{\mathcal{X}} |h(z)|p(z)dz} \propto |h(x)|p(x) \tag{24}$$

B.4 Sensitivity to β in Uniform Prior

Our Rating Aggregation Method 3a Bayes - Uniform Sampling described in Section 3.2 depends on the β hyperparameter to select the uniform prior. In this section we conducted a test on Setting 2 (MTurk test) comparing different values for β and show that it has very little effect on the correlation

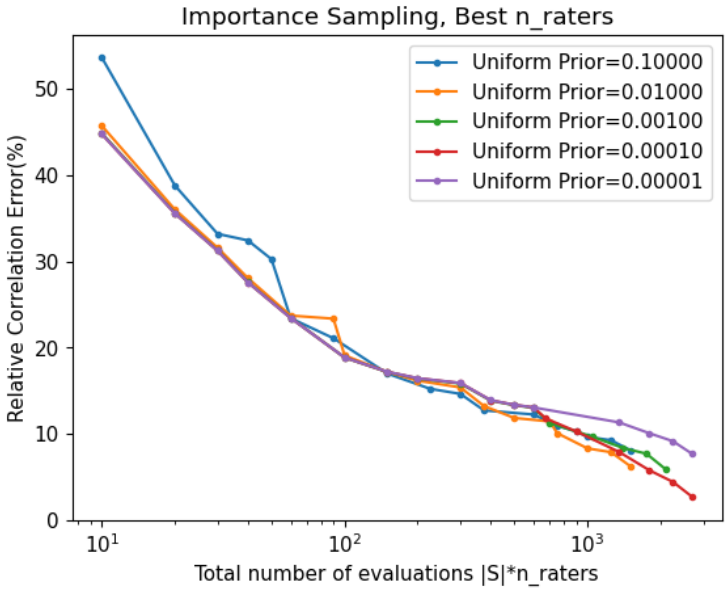


Figure 5: Comparing the error rate of using different β values for uniform prior.

error as shown in Fig 5. In our experiments we used $\beta = 0.01$ which performs well overall but as we can see the exact choice has little effect on results. We think this is mostly because changing prior has a relatively linear effect on predicted c_t, and since correlation coefficient normalizes the c_t the scale of c_t does not change the correlation.

B.5 Compute details

Our main contribution is focused on efficient crowd-sourced evaluation and as such our method is not computationally costly. The main computational cost associated with our method is calculating the SigLIP image encoder outputs for the entire $\mathcal{D}$, as these are needed for both importance sampling and the Bayes - SigLIP prior. However this is a relatively cheap onetime cost of around 20 minutes on a single NVIDIA RTX 6000 Ada Generation GPU.

The main computational expense associated with this paper involved running the baseline methods. In Tables 4 and 5 below, we report the approximate runtime to explain all neurons of a layer using different description methods using a single NVIDIA RTX 6000 Ada Generation GPU.

Simple Exp. (l=1)	ND [3]	MILAN [14]	CD [20]	INVERT l=1 [7]	DnD [1]	MAIA [26]	LE(label) l=1 [21]	LE(SigLIP) l=1 [21]
RN-50 (Layer4)	$\sim$ 1 hr	$\sim$ 1 hr	$\sim$ 5 mins	$\sim$ 1 hr	$\sim$ 55 hrs	$\sim$ 255 hrs	$\sim$ 1 hr	$\sim$ 1 hr

Table 4: Approximate runtime of different $l = 1$ baseline methods to explain neurons.

Complex Exp. (l>1)	Comp Exp [19] l=3	INVERT [7] l=3	CCE [17], l= 5	CCE [17], l=15	LE(label) [21] l=4.37/1.97	LE(SigLIP) [21] l=4.66/1.82
RN-50 (Layer4)	$\sim$ 92 hrs	$\sim$ 24 hrs	$\sim$ 87 hrs	$\sim$ 275 hrs	$\sim$ 1 hr	$\sim$ 1 hr

Table 5: Approximate runtime of different complex explanation baseline methods.

C Additional Figures

Figures 6 and 7 showcases example neurons and the descriptions assigned by different explanation methods, as well as the correlation scores estimated by our crowd-sourced study for those explanations. We colored explanations based on the estimated correlation coefficient, with Green: $\rho \geq 0.25$, Yellow: $0.25 > \rho \geq 0.15$ and Red: $0.15 > \rho$.

MTurk Experimental Details: Figure 8 showcases the full user interface displayed to the raters. We selected raters based in the US with over 10,000 tasks approved and $> 98\%$ task approval rate. Each rater was paid \$0.05 per task(and we paid another \$0.01 per task in fees to MTurk), which takes around 15 seconds to complete based on our testing, for an estimated \$12/hr earnings.

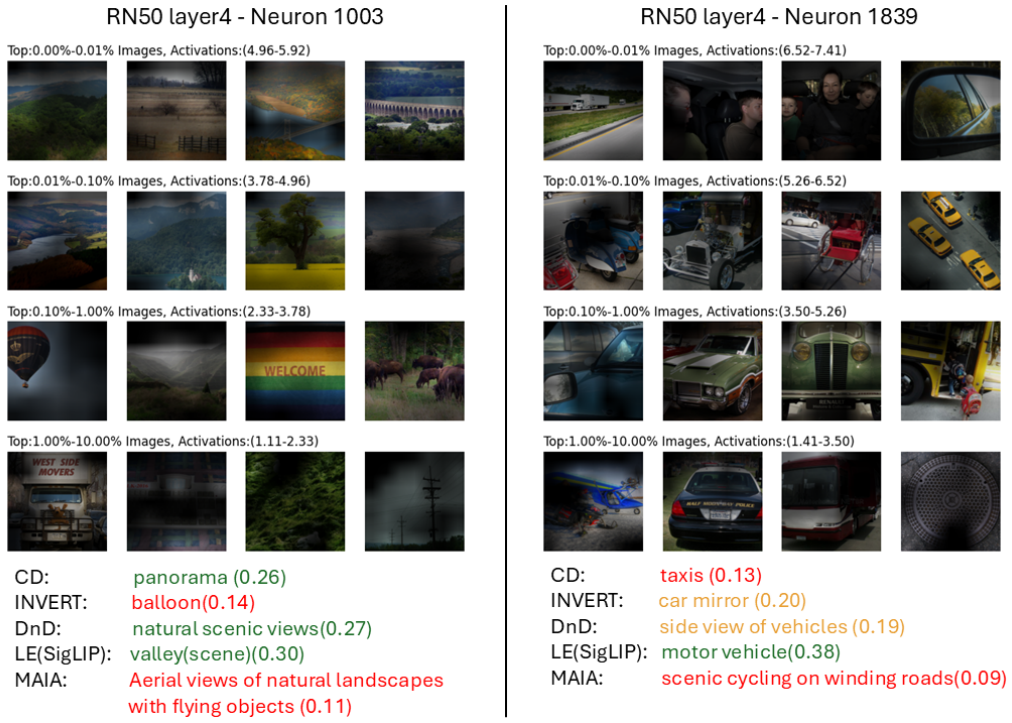


Figure 6: Visualization of example neurons, their descriptions and correlations scores from our crowdsourced evaluation (Bayes with SigLIP prior). We have colored the descriptions based on the correlation score.

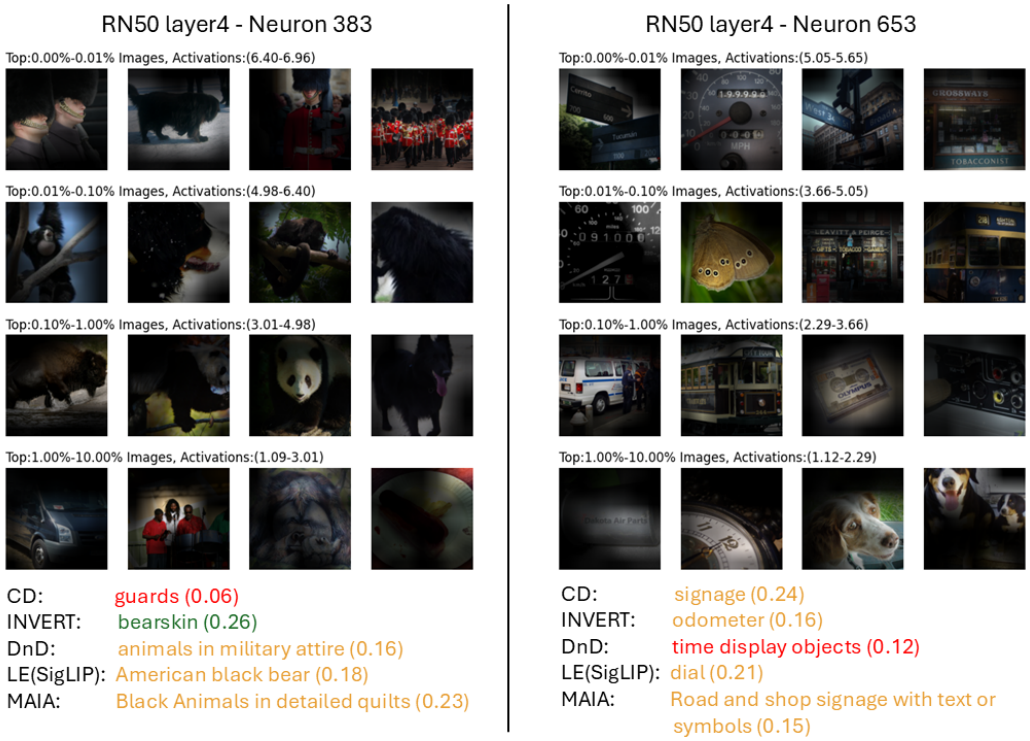


Figure 7: Visualization of example neurons, their descriptions and correlations scores from our crowdsourced evaluation (Bayes with SigLIP prior). We have colored the descriptions based on the correlation score.

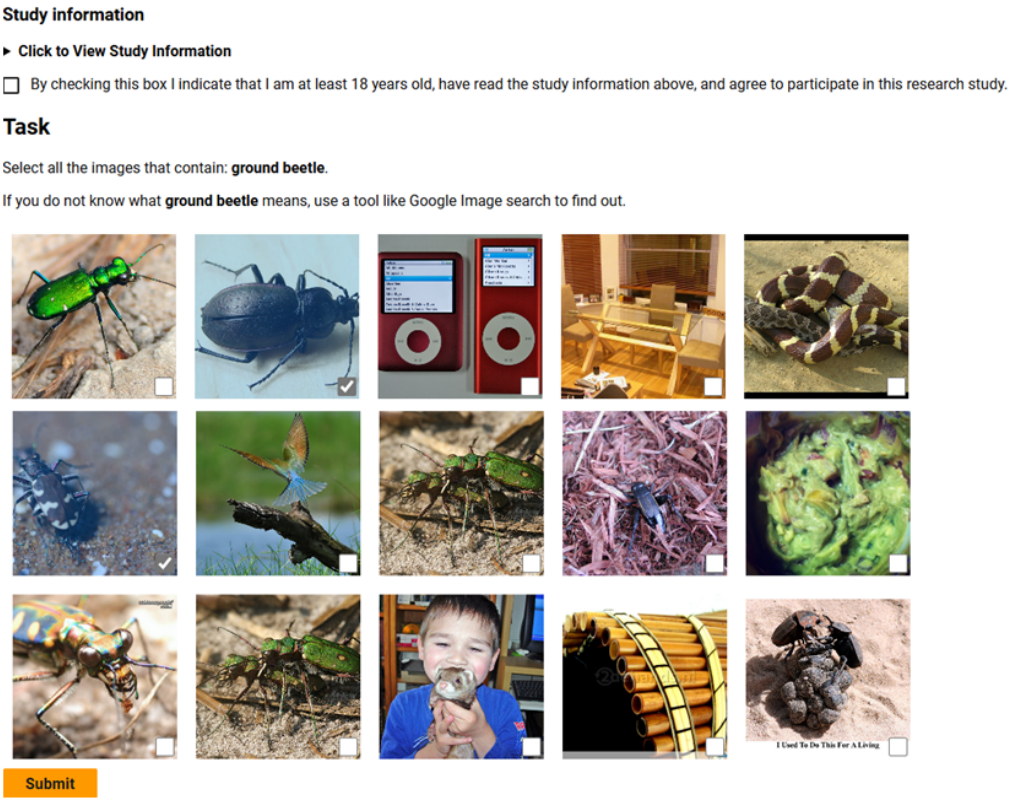


Figure 8: Our user study interface.