Towards Trustworthy Neuron Identification: Faithfulness and Stability

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neuron identification is a popular tool in mechanistic interpretability, aiming to uncover the human-interpretable concepts represented by individual neurons in deep networks. While algorithms such as Network Dissection and CLIP-Dissect achieve great empirical success, a rigorous theoretical foundation remains lacking. In this work, we formalize neuron identification as the *reverse process of learning*, which allows us to import tools from statistical learning theory. From this perspective, we present the fist theoretical analysis of two foundamental challenges: (1) **Faithfulness:** whether the identified concept truly represents the neuron and (2) **Stability:** whether the results are consistent across probing datasets. We derive generalization bounds for widely used similarity metrics (e.g. accuracy, AUROC, IoU) to guarantee faithfulness, and propose a bootstrap ensemble procedure that quantifies stability and provides probabilistic guarantees via prediction sets. Experiments on both synthetic and real data validate our theoretical results and demonstrate the practicality of our method, providing a step toward trustworthy neuron identification.

1 Introduction

Despite the rapid development and application of deep neural networks, their lack of interpretability raises growing concerns[16, 18]. A popular approach to "open the black-box" is to analyze individual neurons and identify human-interpretable concepts that capture their behavior. This process is known as **neuron identification** (or neuron explanation)[3].

Over the past few years, many approaches for neuron identification have been proposed. For example, Bau et al. [3] compare neuron activation with labeled concept datasets to find corresponding concept. Oikarinen and Weng [12] leverage multimodal models to automatically generate neuron explanation.

Despite rapid progress in empirical method, systematic comparison and theoretical understanding of neuron identification remain limited. Oikarinen et al. [14] unify neuron identification methods under a single mathematical framework for fair comparison, but a rigorous theoretical investigation is still lacking. In particular, we find two major challenges on current neuron identification framework: **faithfulness** and **stability**

1. **Faithfulness.** *Does the identified concept faithfully describe the underlying neuron?*
2. **Stability.** *How consistent is the identified concept across different probing datasets?*

To address these challenges, we provide a theoretical analysis based on a key observation: *neuron identification can be viewed as the reverse process of learning*. This perspective highlights the

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

parallels between neuron identification and traditional machine learning, allowing us to apply tools from statistical learning theory on neuron identification. Our contributions are summarized below:

1. We show that **neuron explanation could be viewed as the reverse process of learning**, which explains why there are so many similarities between neuron identification and traditional machine learning. This enables us to import tools from statistical learning theory to answer the questions in neuron identification.
2. We analyze **faithfulness** using generalization theory, proving results for several similarity metrics and showing square-root convergence of test similarity with respect to probing dataset size.
3. We quantify **stability**/uncertainty via bootstrap ensemble over probing datasets.

The remaining of this paper is organized as follows: Sec. 2 formalizes neuron identification and introduce the background. Sec. 3 analyzes the concept faithfulness via generalization bounds. Sec. 4 quantifies algorithm stability using bootstrap ensemble method. Sec. 5 shows empirical results and Sec. 6 summarizes the work and discusses the limitations.

2 Preliminary

In this section, we introduce the background of neuron identification and the notations we use as a preliminaries for our theory. Let $\mathcal{X}$ denote the input space (e.g. images). First, we formally define neuron representation and concept.

1. **Neuron representation** $f(x) : \mathcal{X} \to \mathbb{R}$: A neuron representation is a function mapping an input $x \in \mathcal{X}$ to an activation value. Generally, the output could be more than a real number, e.g. for convolutional neural networks (CNN) $f(x)$ is a 2-D feature map. For the simplicity in similarity calculation, existing works [3, 12, 5] often conduct pooling (average, max) to aggregate the feature into a single real value.
2. **Concept function** $c(x)$: In the literature of neuron identification [3, 12], a concept is usually defined as a human-understandable idea that is described by text. For example, "cat" and "red". Although intuitive, this definition is not a formal math definition. In this work, we define concepts as a function: a concept function $c(x) : \mathcal{X} \to \{0, 1\}$ is a function that takes images[1] as input, and outputs 1 if the concept is present in this image and 0 if not. This definition is compatible with the previous works: for example, Bykov et al. [5] uses human annotation which outputs 1 if the concept presents, otherwise 0. Oikarinen and Weng [12] uses CLIP [15] activations and calculate the cos-similarity of the input image embedding and text embedding of concept, which could be regarded as an automatic way to approximate $c(x)$.

To search for a concept that describes the neuron, most methods utilize a similarity function $\mathsf{sim}(f, c)$. It's a functional measuring the similarity between concept and neuron. With the similarity function, the neuron identification problem can be formulated as:

$$\hat{c}(x) = \underset{c(x) \in C}{\arg\max}\, \mathsf{sim}(f(x), c(x)) \tag{1}$$

where C is the concept set (a function space under our concept definition).

In our formal definition, $\mathsf{sim}(f, c)$ is a functional that takes two functions f and c as input, e.g. correlation. In practice, most works replace the function to its realization on a probing dataset D_{probe} as an approximation. For example, for the similarity function of accuracy, it is defined as the probability that two function has the same value:

$$\mathsf{sim}(f, c) = \mathbf{P}(f(x) = c(x)), \tag{2}$$

Utilizing probing dataset, we can get an unbiased empirical estimation:

$$\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) = \frac{1}{|D_{\text{probe}}|} \sum_{i=1}^{|D_{\text{probe}}|} \mathbf{1}(f(x_i) = c(x_i)). \tag{3}$$

[1] The input could also be texts or audio. In this work we focus on vision models.

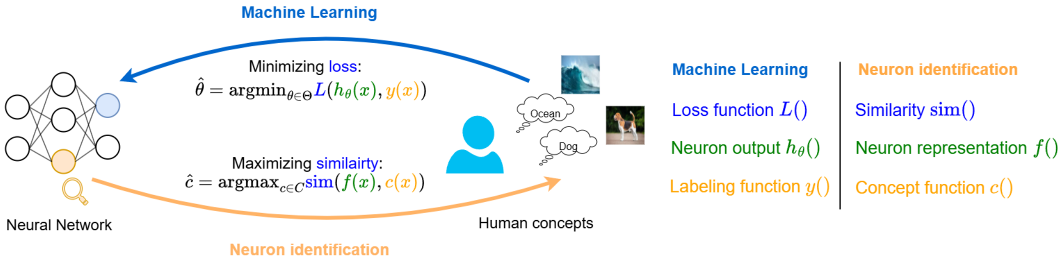


Figure 1: Illustration of neuron identification and machine learning. Neuron identification searches for concepts matching a neuron, while learning searches for model parameter matching human concepts. Thus, neuron identification can be viewed as the reversed learning process.

Under this approximation, the optimization goal can be written as:

$$\begin{gathered}\hat{c} = \arg\max_{c\in C} \quad \hat{\mathsf{sim}}(f, c; D_{\text{probe}}) \\ \text{where } \hat{\mathsf{sim}}(f, c; D_{\text{probe}}) = \hat{\mathsf{sim}}(f(x_i), c(x_i)), x_i \in D_{\text{probe}}.\end{gathered} \tag{4}$$

It could be seen that D_{probe} plays an important role in this approximation. However, the investigation on D_{probe} is still lacking.

Why do we choose similarity-based definition? Currently, there is no formal definition of a neuron's concept. A practical criterion is: a concept describes a neuron if the neuron's activation can be used to predict the presence of that concept. This criteria can be measured by corresponding classification metrics, for example, F1-score [8] or AUC [9]. The definition of similarity scores include these metrics and generalize to other useful scores, e.g. correlation or soft-wpmi[12]. Thus, we adopt it in our theoretical framework.

3 Explanation faithfulness

In this section, we start to discuss a key question in neuron identification: *How can we trust the neuron explanation provided by the algorithm?* We first discuss an important observation: **Neuron identification could be regarded as the reverse process of machine learning**. Inspired by that, we utilize the rich literature in statistical learning theory to study faithfulness in Sec. 3, and quantify uncertainty of neuron explanation in Sec. 4.

3.1 Duality of neuron explanation and learning

Based on the notation in Sec. 2, we observe that the neuron identification problem closely parallels the traditional learning problem. For example, given a model with parameter θ in parameter space Θ, a classification problem could be formalized as minimizing the loss L, which is approximated by the empirical loss $\hat{L}$ on the training dataset:

$$\begin{gathered}\hat{\theta} = \arg\min_{\theta\in\Theta} \hat{L}(\theta; D_{\text{train}}) \\ \text{where } \hat{L}(\theta; D_{\text{train}}) = \hat{L}(h_\theta(x_i), y(x_i)), x_i \in D_{\text{train}},\end{gathered} \tag{5}$$

where $y(x)$ denotes the label function and $h_\theta(x)$ is the neural network. Comparing Eq. 5 and Eq. 4, These two problems have a similar form: Both problems solve an optimization problem and the objectives are identical in form. We list the correspondence between these two domains in Fig. 1.

Furthermore, we observe that the neuron identification can be regarded as the reverse process of learning: during learning, we search for neural network (parameters) that mimics target human concept (e.g. ImageNet classes). Neuron identification, instead, searches for concept (or simple combination of concepts) that is most similar to a specific neuron.

This observation enables us to utilize the rich literature of machine learning in the neuron identification problem. Below, we first discuss how to measure the **faithfulness** of neuron explanation via the generalization theory. Next, we discuss how to perform uncertainty quantification and measure **stability** in Sec. 4.

3.2 Explanation faithfulness

A natural question for the neuron identification is: *Does the concept identified by the algorithm faithfully describe the neuron?* To answer this, we first need a quantitative measure of faithfulness. Under the framework of similarity score, the faithfulness could be split into two questions:

1. Which similarity function shall we choose? This question has been discussed by Oikarinen et al. [14] so we will not put our focus on it.
2. Does the selected concept have a high (true) similarity score? Since concept is selected based on probing dataset, we need to investigate how this influence the true similarity score of output concept.

Inspired by the traditional generalization theory in machine learning [17], we address the second question by first defining the generalization gap for neuron identification as:

$$g(D_{\text{probe}}, C, f) \triangleq \sup_{c \in C} [\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) - \mathsf{sim}(f, c)]. \tag{6}$$

We show that this gap can be bounded under mild assumptions:

1. The concept set C is finite.
2. The probing dataset D_{probe} is sampled i.i.d.
3. Similarity function sim is bounded. More specifically, $0 \leq \mathsf{sim}(x) \leq 1$.

For a finite concept set C, we have the following theorem:

Theorem 3.1. *With probability at least* $1 - \delta$*,*

$$\sup_{c \in C} |\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) - \mathsf{sim}(f, c)| \leq r(f, D_{\text{probe}}, \frac{\delta}{|C|}),$$

where $r(f, D_{\text{probe}}, \delta)$ *describes the convergence rate of similarity function* $\hat{\mathsf{sim}}(f, c; D_{\text{probe}})$ *and satisfies*

$$\mathbb{P}\left[\left|\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) - \mathsf{sim}(f, c)\right| \geq r(f, D_{\text{probe}}, \delta)\right] \leq \delta. \tag{7}$$

In the first equation, the confidence parameter is adjusted using a union bound, replacing δ *with* $\frac{\delta}{|C|}$.

Remark 3.2. This theorem follows classical result in generalization theory and could be directly derived via union bound. The convergence rate function $r(f, D_{\text{probe}}, \delta)$ describes how fast the estimator $\hat{\mathsf{sim}}$ converges. We will show that for most popular similarity estimators in practice, $r(f, D_{\text{probe}}, \delta) = \mathcal{O}(\sqrt{\frac{-\log \delta}{|D_{\text{probe}}|}})$

Corollary 3.3. *With probability at least* $1 - \delta$*,*

$$\mathsf{sim}(f, \hat{c}) \geq \max_{c \in C}[\mathsf{sim}(f, c)] - 2r(f, D_{\text{probe}}, \frac{\delta}{|C|}). \tag{8}$$

where $\hat{c}$ *is the optimal concept based on the probing dataset, as defined in Eq. 4.*

Remark 3.4. This corollary suggests that by maximizing similarity on the probing dataset, we can find an **approximately optimal** concept within a gap decided by convergence rate of the similarity function and size of concept set.

3.3 Illustrative bound on popular similarity metrics

In this section, we discuss the convergence rate of common similarity metrics. For simplicity, we directly list the empirical estimator $\hat{\mathsf{sim}}$; the true similarity function is its expectation unless otherwise mentioned.

Example 1: Accuracy The accuracy could be used as a similarity metric:

$$\mathsf{sim}_{\text{Acc}}(f,c) = \mathbb{P}(f(x) = c(x)). \tag{9}$$

The accuracy can be estimated from samples by:

$$\hat{\mathsf{sim}}_{\text{Acc}}(f,c) = \frac{\sum_{x\in D_{\text{probe}}} \mathbf{1}(f(x) = c(x))}{|D_{\text{probe}}|}. \tag{10}$$

The convergence rate can be estimated via the Hoeffding's inequality [7]:

$$r_{\text{Acc}}(f, D_{\text{probe}}, \delta) = \sqrt{\frac{\log(\frac{2}{\delta})}{2|D_{\text{probe}}|}} \tag{11}$$

Example 2: AUC Area under the ROC curve (AUC) is also a popular similarity function used in practice[5]. The AUC similarity is calculated as:

$$\mathsf{sim}(f,c) = \frac{\sum_{\{x|c(x)=0\}} \sum_{\{x|c(x)=1\}} \mathbf{1}[f(x) < f(y)]}{|\{x \mid c(x) = 0\}||\{x \mid c(x) = 1\}|}. \tag{12}$$

[1] proved that the AUC estimator converges to the expected ranking accuracy with rate

$$r_{AUC}(f, D_{\text{probe}}, \delta) = \sqrt{\frac{\log(\frac{2}{\delta})}{2\rho(\underline{c})(1-\rho(\underline{c}))|D_{\text{probe}}|}}, \tag{13}$$

where $\rho(\underline{c})$ is called positive skew which equals the the portion of positive examples in the probing dataset. We refer to Theorem 2 of [1] a formal statement and proof.

In Fig. 5a, we show the convergence rate r_{AUC} under different ρ. We can see that when ρ is small, the convergence rate r_{AUC} blows up when D_{probe} is small, indicating that imbalanced probing datasets may cause larger generalization error, reducing explanation faithfulness.

Example 3: Recall, precision and IoU Precision, recall, and intersection-over-union (IoU) are also widely used similarity metrics in neuron identification. Given a neuron representation $f(x)$ and a concept function $c(x)$, we define:

$$\hat{\mathsf{sim}}_{\text{prec}}(f,c) = \text{Precision}(f,c) = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \hat{\mathsf{sim}}_{\text{rec}}(f,c) = \text{Recall}(f,c) = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

$$\hat{\mathsf{sim}}_{\text{IoU}}(f,c) = \text{IoU}(f,c) = \frac{\text{TP}}{\text{TP}+\text{FP}+\text{FN}}$$

where TP (true positives), FP (false positives), and FN (false negatives) are computed over the probing dataset D_{probe} based on the binary predictions from $f(x)$ and ground-truth labels from $c(x)$. These metrics can be regarded as conditional versions of accuracy: for example, precision can be regarded as the accuracy in examples where $f(x) = 1$. Thus, the convergence rate is similar to r_{Acc}, with the only difference being the effective dataset size:

$$r_{\text{prec}}(f, D_{\text{probe}}, \delta) = \sqrt{\frac{\log(\frac{2}{\delta})}{2|f(x) = 1 \mid x \in D_{\text{probe}}|}}. \tag{14}$$

The convergence rate for recall and IoU can be calculated similarly, with effective sample size $|c(x) = 1 \mid x \in D_{\text{probe}}|$ and $|c(x) = 1 \text{ or } f(x) = 1 \mid x \in D_{\text{probe}}|$, respectively.

In practice, users can collect additional data until the effective sample size reaches a desired level. Here, for easy comparison of different metrics, we plot r v.s. expected number of total samples required.

4 Quantifying stability

Another important question in neuron identification is *how to quantify the stability of the algorithm across different probing datasets*. This also quantifies uncertainty in neuron identification results. Leveraging the connection to machine learning, we adopt the *bootstrap ensemble* approach for stability/confidence estimation, which is applicable to any neuron identification algorithm without the need to modify its internal mechanism.

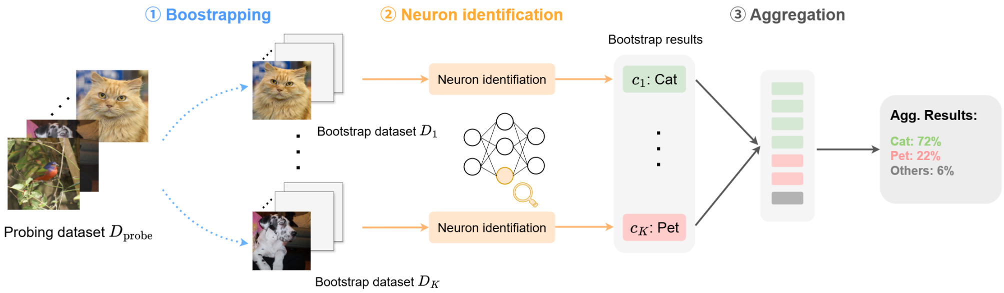


Figure 2: Illustration of bootstrap ensemble in neuron identification. Multiple probing datasets are generated via bootstrapping. Then, neuron identification algorithm is applied on each dataset and final concepts are aggregated into probability of each concept.

4.1 Bootstrap ensemble

Bootstrap ensemble [4] is a machine learning technique that can improve both prediction accuracy and uncertainty quantification. This method aggregates multiple models, each trained on a different resampled version of the original dataset, obtained via bootstrapping (sampling with replacement). The final prediction is determined by majority voting among these models, and confidence is calculated as the proportion of models voting for the final prediction [10].

For neuron identification, we adapt the bootstrap ensemble approach by bootstrapping the probing dataset to produce multiple identification outcomes for the same neuron. This process is analogous to training multiple models in standard machine learning. The procedure can be described as:

1. **Collect bootstrap datasets:** Sample K datasets $\{D_i\}_{i=1}^{K}$ independently by randomly selecting samples from the probing dataset D_{probe} with replacement.
2. **Run neuron identification:** Apply the neuron identification algorithm to each bootstrap datasets D_i and record the predicted concept c_i.
3. **Aggregate predictions:** After K runs, estimate the probability of each concept as:

$$\mathbb{P}(c) = \frac{1}{K} \sum_{i=1}^{K} \mathbf{1}(c_i = c), \tag{15}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

4.2 Construct prediction set

While bootstrap ensembles provide an empirical measure of stability, we also seek theoretical guarantees on the top concept. In particular, we want to bound the probability that the most frequent concepts in bootstrap ensemble capture the desired concept[2]. For this goal, we construct a **concept prediction set**, a set of concepts likely to describe the neuron, rather than a single best guess. This approach could be applied on any neuron identification algorithm, without any modification. The procedure is described in Alg. 1.

The following theorem gives a probabilistic guarantee that a desired concept c^* will be included in the prediction set constructed via the bootstrap ensemble, under mild assumptions on the candidate set and similarity function.

Lemma 4.1. *Let p be defined implicitly by the equation*

$$r(f, D_{\text{probe}}, \frac{p}{|C|}) = \frac{\Delta}{2}, \tag{16}$$

where $r(\cdot)$ is the uniform convergence rate in Theorem 3.1. Then,

$$\mathbb{P}(\hat{c} = c^*) \geq 1 - p \tag{17}$$

[2] It's similar to ground truth in conventional machine learning.

Remark 4.2. Lemma 4.1 can be easily derived from Theorem 3.1: with probability $1 - p$, $\sup_{c \in C} |\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) - \mathsf{sim}(f, c)| \le \frac{\Delta}{2}$, thus

$$\begin{aligned} \hat{\mathsf{sim}}(f, c^*; D_{\text{probe}}) &\ge \mathsf{sim}(f, c^*; D_{\text{probe}}) - \frac{\Delta}{2} \\ &\ge \mathsf{sim}(f, c; D_{\text{probe}}) + \frac{\Delta}{2} \quad \text{(Assumption 2)} \\ &\ge \hat{\mathsf{sim}}(f, c; D_{\text{probe}}). \end{aligned} \tag{18}$$

In many cases, p is bounded by $p < e^{-Q|D_{\text{probe}}|\Delta^2}$ for some constant $Q > 0$, depending on the similarity metric.

Theorem 4.3. *Let c^* be the desired concept for a given neuron. Assume that*

1. *$c^* \in C$ (the target concept is included in candidate concept set).*
2. *$\mathsf{sim}(f, c^*) \ge \mathsf{sim}(f, c) + \Delta, \forall c \in C, c \neq c^*$, where $\Delta > 0$ is a positive constant. This assumes the similarity function can distinguish the target concept with other concepts.*

Let $S \subseteq C$ be the prediction set constructed in Alg. 1, and let $k(S) = \sum_{i=1}^{K} [\hat{c}_i \in S]$ be the number of bootstrap trials that predict a concept in S. Then, under these assumptions,

$$\mathbb{P}(c^* \in S) \ge \sum_{i=0}^{K-k(S)-1} \binom{K}{i} p^i (1-p)^{K-i}, \tag{19}$$

where p is the single-trial error probability defined in Lemma 4.1.

Theorem 3.1 provides a statistical guarantee on the probability that our desired concept is included in the prediction set. We postpone its proof to appendix A.1.

5 Experiments

In this section, we conduct experiments to evaluate our proposed methods.

5.1 Simulation on synthetic data

To verify our theory in Sec. 3, we conduct simulations based on a synthetic dataset.

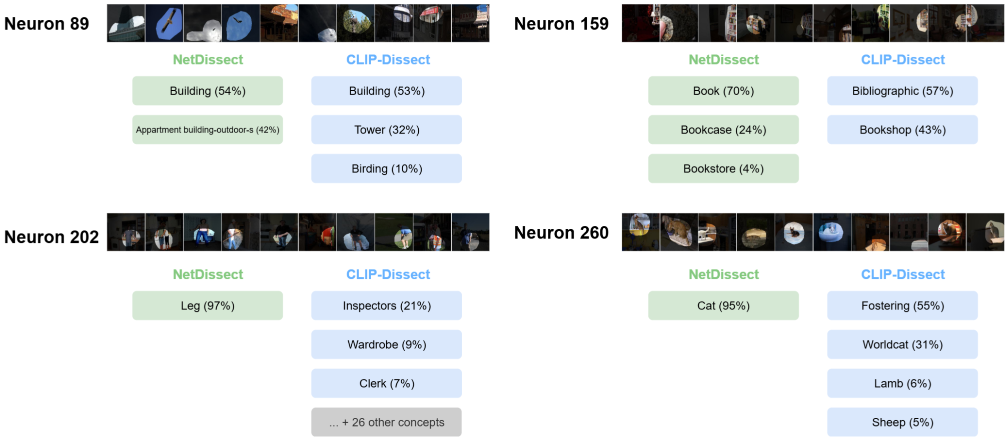
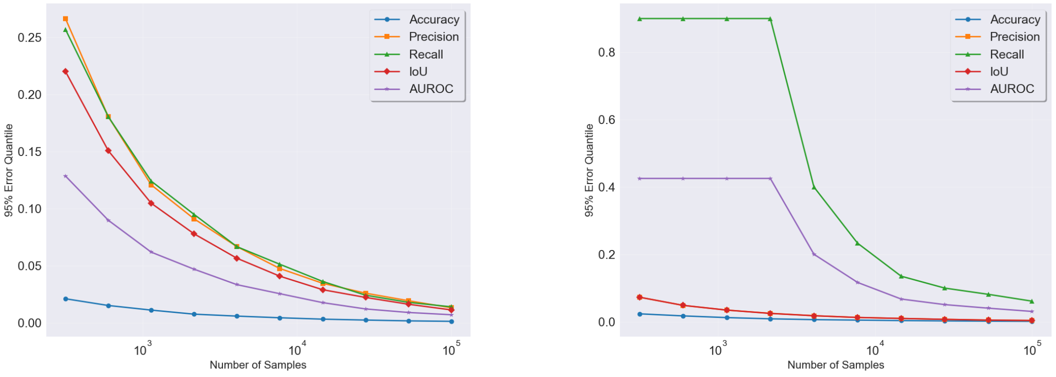


Figure 3: Results of applying bootstrap ensemble to NetDissect and CLIP-Dissect on ResNet 50 neurons. NetDissect shows more stable, concrete concepts. CLIP-Dissect outputs are more diverse and abstract.



(a) Setting 1

(b) Setting 2

Figure 4: 95% quantile of error of 5 similarity metrics under two synthetic simulation settings: (a) balanced concept frequency; (b) rare concept frequency (0.001). Accuracy converges fastest in both settings.

Experiment 1: Convergence speed. In Theorem 3.1, the key factor that controls the gap is the convergence rate r. To investigate this, we generate synthetic data and compare different similarity functions. We binarize the neuron representation by thresholding the top 5% activations and study the following two cases:

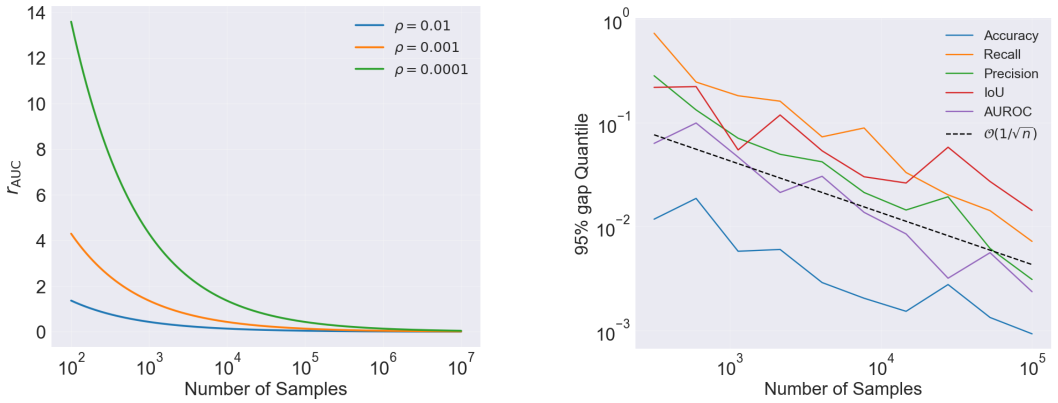
1. $\mathbb{P}(f(x) = 1, c(x) = 1) = 0.03$, $\mathbb{P}(f(x) = 1, c(x) = 0) = 0.02$, $\mathbb{P}(f(x) = 0, c(x) = 1) = 0.02$, $\mathbb{P}(f(x) = 0, c(x) = 0) = 0.93$. This case simulates a regular concept.
2. $\mathbb{P}(f(x) = 1, c(x) = 1) = 0.0009$, $\mathbb{P}(f(x) = 1, c(x) = 0) = 0.0491$, $\mathbb{P}(f(x) = 0, c(x) = 1) = 0.9499$, $\mathbb{P}(f(x) = 0, c(x) = 0) = 0.0001$. This case simulates the case that concept is rare (frequency is 0.001). This case often occurs when the concept is fine-grained.

We simulate with $N_{\text{exp}} = 1000$ randomly sampled dataset and plot how the 95% quantile of error change with the number of samples, as shown in Fig. 4. From the simulation results, we can see that

1. Accuracy has fastest convergence in both cases. On regular concept, IoU, recall and precision are similar. AUROC converges faster than them.
2. For rare concept, the case is different: AUROC and recall are much worse than precision and IoU. This matches our analysis in Sec. 3, where we showed that AUROC converges significantly slower when the concept frequency is low.

Experiment 2: Gap simulation In this test, we verify Theorem 3.1 via synthetic data. We generate the synthetic data with the following steps:

1. **Generate neuron representation.** Binarized neuron representation $f(x)$ is generated by setting the top-5% of activations to 1 and the rest to 0.
2. **Generate concepts.** We generate $|C| = 1000$ concepts as the candidate set. For each concept, we first generate its frequency from a log-uniform distribution in the interval $(10^{-4}, 10^{-1})$. Then, we randomly generate $TP = \mathbb{P}(f(x) = c(x))$ from $(0, \min[\mathbb{P}(f(x) = 1), \mathbb{P}(c(x) = 1)])$ to ensure the probability is valid. The rest probability (FP, TN, FN) can then be inferred. Given the probabilities, we compute corresponding conditional probability ($\mathbb{P}(c(x) \mid f(x))$ and sample $c(x)$ accordingly.
3. **Experiment and simulation.** We repeat the above steps $N_{\text{exp}} = 1000$ times. We use the sampled neuron representation $f(x)$ and concept activation $c(x)$ to calculate similarity and select top-ranked concept $\hat{c}$. Then, we calculate the ground-truth similarity with the real probability (TP, FP, TN, FN) and calculate the error as the difference between similarity of selected concept and max similarity in the candidate set ($\max_{c \in C}[\mathsf{sim}(f, c)] - \mathsf{sim}(f, \hat{c})$. We take the 95% quantile of error among all experiments to approximate the bound under success probability $1 - \delta = 95\%$.



(a) Convergence rate r_{AUC} with respect to probing dataset size $|D_{\text{probe}}|$ under different positive rates (concept frequency) $\rho(\underline{c})$.

(b) Simulation of the generalization gap predicted by Theorem 3.1 versus probing dataset size, showing an empirical convergence rate of $\mathcal{O}(1/\sqrt{n}$.

Figure 5: Theoretical and simulation results on generalization gap.

In Fig. 5b, we plot the simulated gap against the size of the probing dataset $|D_{\text{probe}}|$. We observe that:

1. All curves have similar slope to the reference $\mathcal{O}(\sqrt{1/n})$ curve, suggesting an asymptotic convergence rate of $\mathcal{O}(\sqrt{1/n})$, which is consistent with our theoretical analysis.
2. For the constant term, accuracy has fastest convergence and AUROC is the second. This matches our simulation of r in **Experiment 1, Setting 1**, supporting our conclusion.

5.2 Bootstrap ensemble

We apply our bootstrap ensemble method to two base methods: CLIP-dissect [12] and NetDissect [3]. The base model we choose is a ResNet-50 model trained on the ImageNet dataset [6]. We run $K = 100$ bootstrap samples and choose the bootstrap count threshold $t = 0.95K = 95$ in Alg. 1. The results are shown in Fig. 3.

From the results, we can observe interesting difference of these two methods:

1. CLIP-Dissect prefers more abstract concept. For example, it gives concepts like fostering and bibliographic. NetDissect, in contrast, always uses concrete concepts.
2. In general, CLIP-Dissect provides more diverse concepts and sometimes captures ones missed by NetDissect (e.g. Birding for Neuron 89). NetDissect is more stable across different bootstrap samples. A potential reason is that NetDissect utilizes localization information, which improves stability.

6 Conclusion and limitation

In this work, we conducted a theoretical analysis of neuron identification problem, with the goal of clarifying the **faithfulness** and **stability** of current algorithms. With the key observation that **neuron identification is the reverse process of learning**, we introduced generalization gap to quantify faithfulness and provided corresponding bounds. We further introduced a bootstrap-based procedure to quantify stability and construct prediction sets of concepts. Together, we offer a principled framework for the trustworthiness of neuron identification, complementing existing empirical studies.

Our work also has some limitations: the bound on generalization gap is a general bound for any concept set. It does not utilize the relation between concepts thus may be improved for specific concept set. The bootstrap ensemble method provides an algorithm-agnostic way to quantify stability and generate prediction sets, but also introduces additional computational overhead.

References

[1] Shivani Agarwal, Thore Graepel, Ralf Herbrich, and Dan Roth. A large deviation bound for the area under the roc curve. *Advances in Neural Information Processing Systems*, 17, 2004.

[2] Nicholas Bai, Rahul Ajay Iyer, Tuomas Oikarinen, and Tsui-Wei Weng. Describe-and-dissect: Interpreting neurons in vision networks with language models. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024.

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[4] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[5] 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.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[8] Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. *arXiv preprint arXiv:2309.10312*, 2023.

[9] Laura Kopf, Philine L Bommer, Anna Hedström, Sebastian Lapuschkin, Marina Höhne, and Kirill Bykov. Cosy: Evaluating textual explanations of neurons. *Advances in Neural Information Processing Systems*, 37:34656–34685, 2024.

[10] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

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

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

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

[14] Tuomas Oikarinen, Ge Yan, and Tsui-Wei Weng. A principled evaluation framework for neuron explanations, 2025. URL `https://openreview.net/forum?id=todLTYB1I7`.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[16] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[17] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[18] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE transactions on emerging topics in computational intelligence*, 5(5):726–742, 2021.

A Details on bootstrap ensemble

Algorithm 1: Generating a concept prediction set for target neuron

```
Input: Concept set $C$, probing dataset $D_{\text{probe}}$, target neuron $f$, neuron identification procedure
       $\texttt{Identify}(C, f, D_{\text{probe}})$, bootstrap sample count $K$, bootstrap count threshold $t$
Output: Prediction set $S$ of candidate concepts
for $i \leftarrow 1$ to $K$ do
    Sample dataset $D_i$ from $D_{\text{probe}}$ with replacement (same size as $D_{\text{probe}}$);
    Calculate $\hat{c}_i = \texttt{Identify}(C, f, D_i)$;
end
For each concept $c_j \in C$, count number its appearances:
    $$k_j = \sum_{i=1}^{K} [\hat{c}_i = c_j]$$
Sort concepts by frequency, $k_{r_1} \geq k_{r_2} \cdots \geq k_{r_s}$, $s$ is the number of different concepts
  generated during bootstrapping;
Initialize $S \leftarrow \emptyset$, $j \leftarrow 0$, cur_count $\leftarrow 0$;
while cur_count $< t$ do
    Add $c_{r_j}$ to $S$: $S \leftarrow S \cup \{c_{r_j}\}$;
    Update $j \leftarrow j + 1$, cur_count $\leftarrow$ cur_count $+ k_{r_j}$
end
```

A.1 Proof for Theorem 3.1

Proof. Suppose we repeat our experiment K times and get $\{\hat{c}_i\}_{i=1}^{K}$. Then, we have the following theorem.

Theorem A.1. *Let $k^* = \sum_{i=1}^{K} \mathbf{1}[\hat{c}_i = c^*]$ denotes the number of times target neuron is given during K experiments. Then,*

$$\mathbb{P}(k^* \geq t) \geq \sum_{i=0}^{t} \binom{K}{i} (1-p)^i p^{K-i} \tag{20}$$

Remark A.2. This could be derived by Lemma 4.1 and binomial distribution CDF.

Using Theorem A.1, we can derive:

$$\begin{aligned} \mathbb{P}(c^* \notin S) &\leq \mathbb{P}(k^* \leq K - k(S)) \\ &= 1 - \mathbb{P}(k^* \geq K - k(S) - 1) \\ &\leq 1 - \sum_{i=0}^{K-k(S)-1} \binom{K}{i} (1-p)^i p^{K-i} \end{aligned} \tag{21}$$

Thus,

$$\mathbb{P}(c^* \in S) \geq \sum_{i=0}^{K-k(S)-1} \binom{K}{i} (1-p)^i p^{K-i}, \tag{22}$$

finishes the proof. □

B Related works

B.1 Neuron identification

The goal of neuron identification is to find a human-interpretable concept that describes the behavior and functionality of a specific neuron. A variety of methods have been proposed for neuron identification. Network Dissection [3] is a pioneering work with the idea of comparing neuron activations with ground-truth concept masks. Subsequent work explored extensions such as compositional explanations [11], automated labeling with CLIP [12], and multimodal summarization [2]. More recent approaches expand the concept space to linear combinations [13]. While these advances provide useful empirical tools, in this work we aim to fill the gap in a principled theoretical foundation for neuron identification.

B.2 Principled framework for neuron identification

To unify the rapid growing neuron identification methods, Oikarinen et al. [14] design a framework, summarizing most neuron identification algorithm into three major components: neuron representation, concept activations and similarity metrics. Additionally, two meta-tests are proposed to compare similarity metrics. While this work provides a good start point, rigorous theoretical analysis is still lacking, which we want to provide in this work.