Faithful and Stable Neuron Explanations for Trustworthy Mechanistic Interpretability

Ge Yan
CSE, UCSD
geyan@ucsd.edu

Tuomas Oikarinen
CSE, UCSD
toikarinen@ucsd.edu

Tsui-Wei (Lily) Weng
HDSI, UCSD
lweng@ucsd.edu

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 absent, which is crucial to enable trustworthy and reliable explanations. In this work, we observe that neuron identification can be viewed as the *inverse process of machine learning*, which allows us to derive guarantees for neuron explanations. Based on this insight, we present the first theoretical analysis of two fundamental challenges: (1) **Faithfulness:** whether the identified concept faithfully represents the neuron's underlying function and (2) **Stability:** whether the identification 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 along with **BE** (Bootstrap Explanation) method to generate concept prediction sets with guaranteed coverage probability. Experiments on both synthetic and real data validate our theoretical results and demonstrate the practicality of our method, providing an important step toward trustworthy neuron identification. [1]

1 Introduction

Despite the rapid development and application of deep neural networks, their lack of interpretability raises growing concerns [Samek et al., 2017, Zhang et al., 2021]. A popular strategy to "open the black-box" is to analyze internal representations at the level of individual neurons and associate them with human-interpretable concepts. This process is known as **neuron identification** in the field of mechanistic interpretability, which yields *neuron explanations* [Bau et al., 2017, Oikarinen and Weng, 2023]. Over the past few years, many neuron identification methods have been proposed. For example, Bau et al. [2017] use curated concept datasets to identify the corresponding concept, while Oikarinen and Weng [2023] leverage multimodal models to automatically generate neuron explanations. A growing body of methods has been developed to identify concepts corresponding to neurons [Srinivas et al., 2025, Huang et al., 2023, Gurnee et al., 2023, Mu and Andreas, 2020, La Rosa et al., 2023, Zimmermann et al., 2023, Bykov et al., 2023, Kopf et al., 2024, Shaham et al., 2024].

Despite rapid empirical progress, systematic comparison and rigorous theoretical understanding of neuron identification remain limited. Recently, Oikarinen et al. [2025] unified the evaluation of neuron identification methods within a single mathematical framework to enable fair comparisons. However, deeper theoretical foundations are still lacking, which undermines the trustworthiness

[1] Our codes are available at https://github.com/Trustworthy-ML-Lab/Trustworthy_Explanations.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability Workshop at NeurIPS 2025.

and reliability of neuron explanations. Consider a chest-X-ray model that predicts pneumonia and attributes its decision to a neuron purportedly representing lung opacity, when in fact the neuron responds to hospital-specific markings. Such unfaithful explanations can mislead clinicians, lead to harmful treatment decisions, and ultimately erode trust.

These concerns motivate a closer examination of the core obstacles to trustworthy neuron explanations. In particular, we identify two central challenges in current neuron identification methods:

1. **Faithfulness.** *Does the identified concept truly capture the neuron's underlying function?*
2. **Stability.** *Is the identified concept consistent across different probing datasets?*

Both challenges are closely connected with probing datasets, which are an essential component of neuron identification methods that determines the stimuli used to measure neuron activity. However, their influence is often overlooked and not rigorously examined. To address these challenges, we provide a theoretical analysis grounded in a key observation: *neuron identification can be (roughly) viewed as an inverse process of learning*. This perspective highlights structural parallels between neuron identification process and traditional machine learning, enabling us to adapt tools from statistical learning theory to formally analyze the effect of probing datasets and bound the performance of neuron identification methods.

Our contributions are summarized as follows:

1. New insights for neuron identification. We are the first to show that neuron identification can be viewed as an inverse process of learning, revealing structural parallels with traditional machine learning. This insight is non-trivial: it enables us to import and adapt tools from statistical learning theory to rigorously analyze key questions in neuron identification that prior work could not address, including the impact of probing datasets.

2. Rigorous guarantees for explanation faithfulness. We establish the first theoretical guarantees for the faithfulness of neuron explanations, answering the critical question of when a concept identified by a neuron-identification algorithm can be trusted. Our analysis is derived under a general framework, making the results applicable to most existing neuron identification methods. Simulation studies demonstrate that our theory allows quantitative analysis of how factors such as probing dataset size, concept frequency, and similarity metrics affect performance.

3. Quantifying stability of explanations. We present the first formal analysis of probing datasets, an essential yet previously overlooked component that determines the stimuli used to measure neuron activity. Using a bootstrap ensemble over probing datasets, we quantify the stability of neuron explanations and design a procedure to construct a set of possible concepts for each neuron, with statistical guarantees on the probability of covering the true concept.

The remainder of this paper is organized as follows: Sec. 2 formalizes the notion of neuron identification. Sec. 3 provides a rigorous analysis of the faithfulness of neuron explanations with high probability guarantees. Sec. 4 quantifies the stability of neuron identification algorithms and establishes statistical guarantees.

2 Formalizing Neuron Identification

In this section, we introduce the formal definition of neuron identification and the notations used in Sec. 3 and 4. Although we use the term "neuron" identification for simplicity, the framework also accommodates larger functional units within the network. Examples include a linear combination of neurons (i.e., a direction in representation space), a feature in a Sparse Autoencoder [Cunningham et al., 2023], a direction derived by TCAV [Kim et al., 2018], or a linear probe [Alain and Bengio, 2016]. Below, we formally define neuron representation and concept:

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. Here, $\mathcal{X}$ denotes the input space (e.g. images [2]). For example, a neuron in an MLP maps the input to a scalar value. For general neural networks, the output may not be a single real number, e.g. for convolutional neural networks (CNN) $f(x)$ is a 2-D feature

[2]The input could also be audio [Wu et al., 2024] or text [Huang et al., 2023, Gurnee et al., 2023]. In this work we focus on vision models.

map. For simplicity in similarity calculation, existing works often conduct pooling (avg, max) to aggregate the feature into a single real value.

Concept label $c(x)$: In the literature of neuron identification [Bau et al., 2017, Oikarinen and Weng, 2023], a concept is usually defined as a human-understandable text description. For example, "cat" or "shiny blue feather". Although intuitive, this definition is not a formal mathematical definition. In this work, we define concepts as a function: a concept $c(x) : \mathcal{X} \to [0, 1]$ is a function that takes images as input, and outputs the probability of the concept. This definition is consistent with the previous works: for example, Bau et al. [2017], Bykov et al. [2024] use human annotations which output 1 if the concept is present, otherwise 0. Oikarinen and Weng [2024] use SigLIP [Zhai et al., 2023] to automatically estimate the probability that concept c appears.

To search for a concept that describes the neuron representation, different methods use different measures (e.g. IoU [Bau et al., 2017], WPMI [Oikarinen and Weng, 2023], AUC [Bykov et al., 2024] and F1-score [Gurnee et al., 2023]). Interestingly, these different methods can all be described by a general similarity function $\mathsf{sim}(f, c)$, which is a functional measuring the similarity between concept $c(x)$ and neuron representation $f(x)$. With the similarity function, the neuron identification problem can be formulated as:

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

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. accuracy, correlation, IoU, etc. In practice, most works replace the function $f(x)$ and $c(x)$ with their realization $f(x_i)$ and $c(x_i)$ on a probing dataset D_{probe} as an empirical approximation, where x_i is sampled i.i.d. from the underlying distribution. For example, the similarity function of accuracy is defined as the probability that two functions have the same value: $\mathsf{sim}(f, c) = \mathbb{P}(f(x) = c(x))$. When utilizing a probing dataset D_{probe}, we can get an unbiased empirical estimation $\hat{\mathsf{sim}}(f, c; D_{\text{probe}})$ for $\mathsf{sim}(f, c)$:

$$\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) = \frac{1}{|D_{\text{probe}}|} \sum_{i=1}^{|D_{\text{probe}}|} \mathbf{1}(f(x_i) = c(x_i)). \tag{2}$$

Under this approximation, the neuron identification can be formulated as the following optimization problem:

[Neuron identification]

$$\begin{aligned} &\hat{c} = \arg\max_{c\in C} \quad \hat{\mathsf{sim}}(f, c; D_{\text{probe}}) \\ &\text{where } \hat{\mathsf{sim}}(f, c; D_{\text{probe}}) = \hat{\mathsf{sim}}(f(x_i), c(x_i)),\ x_i \in D_{\text{probe}}. \end{aligned} \tag{3}$$

Eq. 3 shows that D_{probe} plays a critical role in this approximation, yet a rigorous analysis of its effect is still lacking. We address this gap in this work in Sec. 3.2 and 4.

Why do we choose similarity-based definition? Similarity provides a broad and unifying notion of a neuron's concept: many existing definitions can be expressed as special cases of similarity with appropriate functions. For example, a common practical criterion is that *a neuron represents concept c if its activation can successfully classify concept c.* This criterion can be formulated as a similarity function using standard classification metrics such as F1-score [Huang et al., 2023], AUC [Kopf et al., 2024], recall [Zhou et al., 2014] and accuracy [Koh et al., 2020].

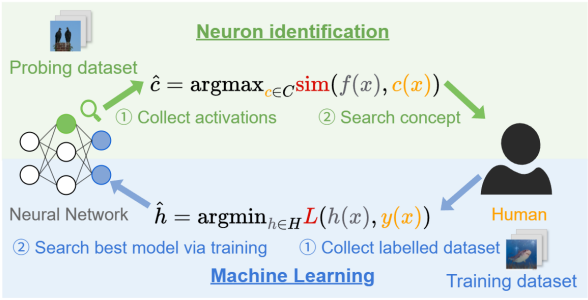
3 Theoretical Guarantees for Explanation Faithfulness

In this section, we address a key question in neuron identification: *When can we trust a neuron explanation produced by a neuron-identification algorithm?* We begin with an important observation: **neuron identification can be viewed as an inverse process of machine learning** in Sec. 3.1. This perspective enables us to derive formal guarantees for explanation faithfulness in Sec. 3.2 and, building on these results, to quantify the stability of neuron explanations in Sec. 4.

3.1 Analogy between neuron identification and machine learning

From the formulation in Eq. 3, we observe that the neuron identification problem closely parallels supervised learning problem. Given a standard classification task and a neural network model $h \in H$,

(1) Illustration of **Inverse relationship** between neuron identification v.s machine learning



(2) **Analogy** between neuron identification and machine learning

Neuron identification	Machine Learning
Similarity function $\text{sim}()$	Loss function $L()$
Neuron output $f()$	Network output $h()$
Concept label $c()$	Class label $y()$

Figure 1: Analogous relationship between neuron identification and machine learning. Neuron identification searches for a concept matching a neuron, while machine learning searches for a model matching human labels. Thus, neuron identification can be viewed as inverse of learning process.

where H denotes the hypothesis space containing all possible neural network models, the problem can be formalized as minimizing the loss L, which is typically approximated by the empirical loss $\hat{L}$ on a training dataset D_{train} as follows:

[Machine learning]

$$\hat{h} = \arg\min_{h \in H} \quad \hat{L}(h; D_{\text{train}})$$
$$\text{where } \hat{L}(h; D_{\text{train}}) = \hat{L}(h(x_i), y(x_i)),\ x_i \in D_{\text{train}}, \tag{4}$$

and $y(x)$ denotes the label function and $h(x)$ is the neural network. Comparing Eq. 4 and Eq. 3, we see that these two problems share a similar structure: Both are optimization problems with objectives of similar form. The left panel of Fig. 1 compares the procedures of these two domains, while the right panel lists their detailed correspondences. As illustrated in Fig. 1, neuron identification can be roughly viewed as the inverse process of machine learning: during learning, we search for neural network (parameters) $h(x)$ that approximates a target human concept $y(x)$ (e.g. ImageNet classes), whereas neuron identification instead searches for concept $c(x)$ (or a simple combination of concepts) that best matches a specific neuron representation $f(x)$.

Importantly, this observation enables us to leverage and adapt tools from machine learning theory while extending them to the unique setting of neuron identification. In the following, we first develop formal guarantees for the **faithfulness** of neuron explanations in Sec 3.2, and then extend this perspective to perform uncertainty quantification and assess **stability** in Sec. 4.

3.2 Theoretical Guarantees for Neuron Explanations

In this section, we address the **faithfulness** challenge: *Does the identified concept truly capture the neuron's underlying function?* Using the framework introduced in Sec. 2, this question reduces to asking whether the identified concept truly achieves high similarity $\mathsf{sim}(f, c)$ to neuron representation. Building on the analogy between neuron identification and machine learning established in Sec. 3.1, we develop a new generalization framework tailored to the neuron identification setting. Although inspired by classical learning theory [Shalev-Shwartz and Ben-David, 2014], our analysis provides the first formal guarantees on the concept-neuron similarity $\mathsf{sim}(f, c)$. We first define the generalization gap g for neuron identification as:

$$g(D_{\text{probe}}, C, f) \triangleq \sup_{c \in C} [\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) - \mathsf{sim}(f, c)]. \tag{5}$$

We show that this gap $g(D_{\text{probe}}, C, f)$ can be bounded in Thm. 3.1 under two mild assumptions: (i) the concept set C is finite, and (ii) the probing dataset D_{probe} is sampled i.i.d. These conditions are met by most existing neuron identification methods, e.g., Bau et al. [2017], Oikarinen and Weng [2023], Bykov et al. [2024].

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

$$\sup_{c \in C} |\hat{\mathsf{sim}}(f, c; D_{\text{probe}}) - \mathsf{sim}(f, c)| \leq r(f, D_{\text{probe}}, \frac{\delta}{|C|}), \tag{6}$$

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

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

In Eq. 6, the confidence parameter δ is adjusted using a union bound, replacing δ with $\frac{\delta}{|C|}$.

Corollary 3.2. *With probability at least $1 - \delta$,*

$$\textsf{sim}(f, \hat{c}) \geq \textsf{sim}(f, c^*) - 2r(f, D_{\text{probe}}, \frac{\delta}{|C|}), \tag{8}$$

where $\hat{c}$ is selected concept using Eq. 3 and $c^* = \arg\max_{c \in C}[\textsf{sim}(f, c)]$ is the optimal concept.

Discussion. Thm. 3.1 adapts classical generalization theory to the neuron identification setting, where the objective of interest is $\textsf{sim}$ and $\hat{\textsf{sim}}$. This provides the first theoretical result on the $\textsf{sim}(f, c)$, which is enabled by our key insight in Sec. 3.1. The convergence rate function $r(f, D_{\text{probe}}, \delta)$ characterizes how fast the estimator $\hat{\textsf{sim}}$ converges. In Sec. 3.2.1, we will derive convergence rates for several popular similarity functions, showing that for many commonly used similarity estimators $r(f, D_{\text{probe}}, \delta) = \mathcal{O}(\sqrt{\frac{-\log \delta}{|D_{\text{probe}}|}})$. On the other hand, Corollary 3.2 suggests that by maximizing similarity on the probing dataset, the identified concept $\hat{c}$ is *approximately optimal*, within a gap determined by the convergence rate of the similarity function and the size of the concept set C. This result guarantees that the concept identified with the probing dataset truly achieves high similarity to the target neuron representation.

3.2.1 Convergence Results for popular similarity metrics

From Thm. 3.1 and Corollary 3.2, we see that the convergence rate is a key factor controlling the generalization gap. Therefore, in this section, we derive and examine the convergence rate of common similarity metrics. Table 1 summarizes several common similarity scores and their convergence rate r:

1. **Accuracy:** This similarity function is used in [Koh et al., 2020], and the convergence rate of accuracy can be estimated via the Hoeffding's inequality.
2. **AUROC:** This similarity function is used in [Bykov et al., 2023], and the convergence rate is related to concept frequency $\rho(c)$ and can be derived using Thm. 2 in Agarwal et al. [2004]. Fig. 3a plots the convergence rate r_{AUROC} under different ρ and shows that when both ρ and $|D_{\text{probe}}|$ are small, the convergence rate r_{AUROC} blows up, indicating imbalanced probing datasets may cause larger generalization error and reduce explanation faithfulness.
3. **Recall, precision, IoU:** These similarity functions are used in [Zhou et al., 2014], [Srinivas et al., 2025], [Bau et al., 2017] respectively. To derive their convergence rates, we view these metrics as conditional versions of accuracy: for example, precision can be regarded as computed only on examples where $f(x) = 1$. Thus, the convergence rate is similar to r_{Acc}, differing only in that the effective sample size changes from $|D_{\text{probe}}|$ to $(F_{11} + F_{10})$. The same reasoning applies to Recall and IoU. In practice, users can collect additional data until the effective sample size reaches desired level. Further details are provided in Sec. B.

Summary. So far, we have derived the generalization gap g for several popular similarity metrics. These results enable practitioners to select an appropriate metric based on available probing data and the properties of the concepts. For example, our experiments in Sec. 3.3 show that AUROC converges quickly when concept frequency is high, but much slower when the frequency is low; in such cases, switching to other similarity metric can reduce the generalization gap and improve performance.

3.3 Simulation studies

To verify the theory developed in Sec. 3.2 and to compare different similarity metrics, we conduct simulations on a synthetic dataset that contains ground-truth similarity values and allows us to simulate a variety of settings. Specifically, we use binary concept $c(x) \in \{0, 1\}$ for simplicity. Neuron activations $f(x)$ are binarized by setting top-5% activations to 1 and remaining to 0. The joint distribution of f, c is controlled by the probability matrix M: $M_{ij} = \mathbb{P}(f(x) = i, c(x) = j),\ i, j \in \{0, 1\}$.

We conduct two experiments: (1) a **single-concept study** to compare convergence speeds and (2) a **multi-concept simulation** to verify Thm. 3.1.

Experiment 1: Convergence speed. In Thm. 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. For the concept, we study the following two settings:

- **Setting 1:** $M =$

$$\begin{array}{c} \\ f=0 \\ f=1 \end{array} \begin{array}{c} \begin{array}{cc} c=0 & c=1 \end{array} \\ \begin{pmatrix} 0.93 & 0.02 \\ 0.02 & 0.03 \end{pmatrix} \end{array}$$

  This case simulates a regular concept.

- **Setting 2:** $M =$

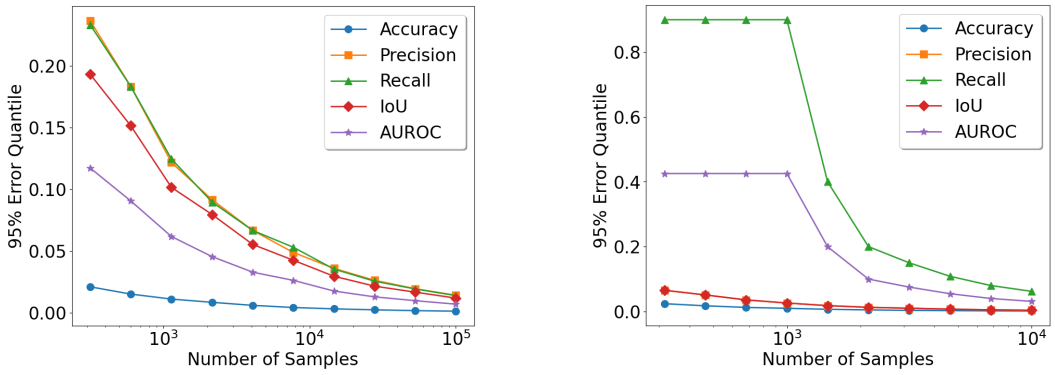
$$\begin{array}{c} \\ f=0 \\ f=1 \end{array} \begin{array}{c} \begin{array}{cc} c=0 & c=1 \end{array} \\ \begin{pmatrix} 0.9499 & 0.0001 \\ 0.0491 & 0.0009 \end{pmatrix} \end{array}$$

  This simulates a rare concept (frequency is 0.001), which often occurs when the concept is fine-grained.

We simulate with $N_{\text{exp}} = 1000$ randomly sampled datasets and plot how the 95% quantile of error changes with the number of samples, as shown in Fig. 2. From the simulation results, we can see that

sim Metric	$\mathsf{sim}(f, c)$	$\hat{\mathsf{sim}}(f, c)$	$r(f, D_{\text{probe}}, \delta)$
Accuracy	$\mathbb{P}(f(x) = c(x))$	$\frac{\sum_{x \in D_{\text{probe}}} \mathbf{1}(f(x)=c(x))}{\lvert D_{\text{probe}} \rvert}$	$\sqrt{\frac{\log(\frac{2}{\delta})}{2\lvert D_{\text{probe}} \rvert}}$
AUROC	$\mathbb{P}(f(x) < f(y) \mid c(x) = 0, c(y) = 1)$	$\frac{\sum_{\{x \mid c(x)=0\}} \sum_{\{y \mid c(y)=1\}} \mathbf{1}[f(x)<f(y)]}{\lvert\{x \mid c(x)=0\}\rvert \lvert\{x \mid c(x)=1\}\rvert}$	$\sqrt{\frac{\log(\frac{2}{\delta})}{2\rho(\underline{c})(1-\rho(\underline{c}))\lvert D_{\text{probe}} \rvert}}$
IoU	$\frac{W_{11}}{W_{01}+W_{11}+W_{10}}$	$\frac{F_{11}}{F_{01}+F_{11}+F_{10}}$	$\sqrt{\frac{\log(\frac{2}{\delta})}{2(F_{11}+F_{10}+F_{01})}}$
Recall	$\frac{W_{11}}{W_{01}+W_{11}}$	$\frac{F_{11}}{F_{01}+F_{11}}$	$\sqrt{\frac{\log(\frac{2}{\delta})}{2(F_{11}+F_{01})}}$
Precision	$\frac{W_{11}}{W_{10}+W_{11}}$	$\frac{F_{11}}{F_{10}+F_{11}}$	$\sqrt{\frac{\log(\frac{2}{\delta})}{2(F_{11}+F_{10})}}$

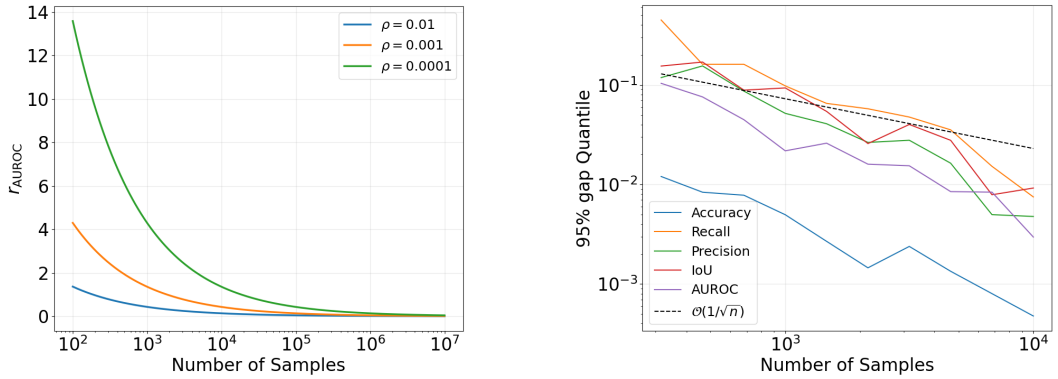
Table 1: Similarity metrics $\mathsf{sim}(f, c)$, estimation $\hat{\mathsf{sim}}(f, c)$ and their corresponding convergence speed $r(f, D_{\text{probe}}, \delta)$. For simplicity, denote $W_{ij} = \mathbb{P}(f(x) = i, c(x) = j),\ i, j \in \{0, 1\}$, $F_{ij} = \frac{\{\lvert f(x)=i, c(x)=j \rvert x \in D_{\text{probe}}\}\rvert}{\lvert D_{\text{probe}} \rvert}$. For AUROC, $\rho(\underline{c})$ is the portion of positive examples in the probing dataset D_{probe} (i.e. the frequency of concept).



(a) Setting 1

(b) Setting 2

Figure 2: 95% quantile of error of 5 similarity metrics under two settings: (a) balanced concept frequency; (b) low concept frequency (0.001). Accuracy converges fastest in both settings.



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

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

Figure 3: Theoretical and simulation results on generalization gap.

1. Accuracy has the fastest convergence in both cases. On regular concept, IoU, recall and precision are similar. AUROC converges faster than them.

2. For rare concept, the convergence pattern differs: AUROC and recall are much worse than precision and IoU. This matches our analysis in Sec. 3.2, where we showed that AUROC converges much more slowly when the concept frequency is low.

Experiment 2: Gap simulation In this experiment, we further verify Thm. 3.1 via synthetic data. Different from **Experiment 1** which simulates single concept, this test requires a concept set C. 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, i.e. $M_{10} + M_{11} = 0.05$.

2. **Generate concepts.** We generate $|C| = 1000$ concepts as the candidate set. For each concept c_i, we first generate its frequency $\mathbb{P}(c_i(x) = 1) = M_{01} + M_{11}$ from a log-uniform distribution in the interval $(10^{-4}, 10^{-1})$. Then, we sample $M_{11} = \mathbb{P}(f(x) = 1, c_i(x) = 1)$ uniformly from $(0, \min[\mathbb{P}(f(x) = 1), \mathbb{P}(c_i(x) = 1)])$ to ensure validity. Given M_{11}, the remaining part of M can then be inferred from concept frequency and activation binarization. Given the probabilities, we compute corresponding conditional probability $(\mathbb{P}(c_i(x) \mid f(x))$ and sample $c_i(x)$ 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 $c_i(x)$ to calculate similarity and select top-ranked concept $\hat{c}$. Then, we compute the ground-truth similarity with the real probability matrix M 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 $1 - \delta = 95\%$.

In Fig. 3b, 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 the fastest convergence and AUROC is the second. This matches our simulation of r in **Experiment 1, Setting 1**, supporting our conclusion.

In summary, the simulation experiments empirically validate the correctness of our theory and show its potential to help users choose appropriate similarity metric under different settings.

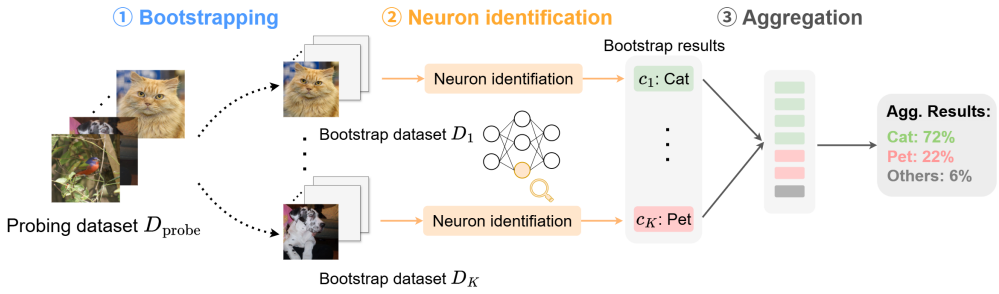


Figure 4: Illustration of bootstrap ensemble in neuron identification. Multiple probing datasets are generated via bootstrapping. Then, neuron identification algorithm is applied to each dataset and final concepts are aggregated to estimate the probability of each concept.

4 Quantifying stability in Neuron Explanations

In this section, we address the second key challenge in neuron identification methods – **stability**: *Is the identified concept consistent across different probing datasets?* Leveraging the connection established in Sec. 3.1, we adopt a *bootstrap ensemble* approach for stability estimation. This method is applicable to any neuron identification algorithm without modifying its internal mechanism. Building on this bootstrapping framework, we further design a method to construct a prediction set of candidate concepts that contains the desired concept with guaranteed probability.

4.1 Empirical measurement via Bootstrap ensemble

Bootstrap ensemble [Breiman, 1996] is a machine learning technique used to improve prediction accuracy and quantify uncertainty. The 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 typically determined by majority voting, and the confidence is estimated as the proportion of models voting for the final prediction [Lakshminarayanan et al., 2017].

For neuron identification, we introduce a bootstrap-based stability framework that resamples the probing dataset to produce multiple identification outcomes for a single neuron. This adaptation allows us to quantify the stability of the neuron explanations obtained. The procedure is:

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 dataset D_i and record the predicted concept c_i.
3. **Aggregate predictions:** After K runs, estimate the probability of each concept as: $\mathbb{P}(c) = \frac{1}{K}\sum_{i=1}^K \mathbf{1}(c_i = c)$, where $\mathbf{1}(\cdot)$ denotes the indicator function.

Fig. 4 summarizes the pipeline. With bootstrap ensemble, the algorithm now outputs probability of each candidate concept.

4.2 Theoretical guarantees via Concept Prediction-Set Construction

While bootstrap ensembles provide an empirical measure of stability, we also seek theoretical guarantees on the identified concept. In particular, we want to bound the probability that the most frequent concepts in bootstrap ensemble capture the desired concept[3]. To achieve this, we construct a **concept prediction set**, a set of concepts that are likely to describe the neuron, rather than a single best guess. This prediction-set approach can be applied to any neuron identification algorithm without any modifications. We call this method **Bootstrap Explanation (BE)** and list the full procedure in Alg. 1 in Sec. A.

[3] Analogous to the ground truth in conventional machine learning.

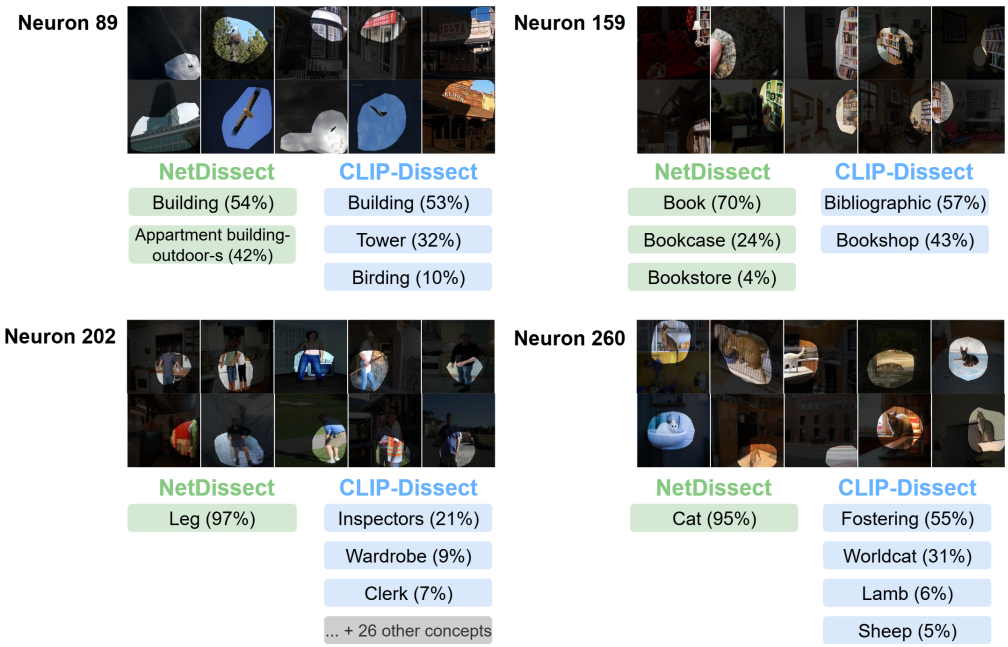


Figure 5: Results of applying bootstrap ensemble to NetDissect and CLIP-Dissect on ResNet-50 neurons. NetDissect shows more stable, concrete concepts. CLIP-Dissect outputs are more diverse and abstract.

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:

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

With these assumptions, we have the following theorem:

Theorem 4.1. *Let c^* be the desired concept for a given neuron and the assumptions above hold for c^*. 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,*

$$\mathbb{P}(c^* \in S) \geq \sum_{i=0}^{K-k(S)-1} \binom{K}{i} p^i (1-p)^{K-i}, \tag{9}$$

where p is the single-trial error probability defined implicitly by the equation $r(f, D_{\text{probe}}, \frac{p}{|C|}) = \frac{\Delta}{2}$.

Thm. 4.1 provides a statistical guarantee on the probability that our desired concept is included in the prediction set. We postpone its proof to Sec. A.1.

4.3 Experiments

We apply our BE method to two base methods: CLIP-Dissect [Oikarinen and Weng, 2023] and NetDissect [Bau et al., 2017]. We use a ResNet-50 model trained on the ImageNet dataset [Deng et al., 2009], run $K = 100$ bootstrap samples and choose the bootstrap count threshold $t = 0.95K = 0.95 \times 100 = 95$ in Alg. 1. The results are shown in Fig. 5.

From the results, we can observe interesting differences between these two methods: (1) CLIP-Dissect prefers more abstract concepts. For example, it gives concepts like fostering and bibliographic. NetDissect, in contrast, tends to identify 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.

5 Related works

5.1 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 [Bau et al., 2017] is a pioneering work with the idea of comparing neuron activations with ground-truth concept masks. Subsequent work explored extensions such as compositional explanations [Mu and Andreas, 2020], automated labeling with CLIP [Oikarinen and Weng, 2023], and multimodal summarization [Bai et al., 2024]. More recent approaches expand the concept space to linear combinations [Oikarinen and Weng, 2024]. While these advances provide useful empirical tools, in this work we aim to fill the gap in a principled theoretical foundation for neuron identification.

5.2 Principled framework for neuron identification

To unify the rapid growing neuron identification methods, Oikarinen et al. [2025] 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.

6 Conclusion and limitations

In this work, we presented a theoretical framework for neuron identification, with the goal of clarifying the **faithfulness** and **stability** of existing algorithms. Building on our key observation that **neuron identification can be viewed as the inverse process of learning**, we introduced the notion of generalization gap to quantify and derive formal guarantees for explanation faithfulness. To quantify stability, we proposed **BE** procedure to construct concept prediction sets with statistical coverage guarantees. Together, these results provide the first 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 sets. The bootstrap ensemble method provides an algorithm-agnostic way to quantify stability and generate prediction sets, but also introduces additional computational overhead.

References

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.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

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.

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.

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

Kirill Bykov, Laura Kopf, Shinichi Nakajima, Marius Kloft, and Marina Höhne. Labeling neural representations with inverse recognition. *Advances in Neural Information Processing Systems*, 36:24804–24828, 2023.

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.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

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.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

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.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

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.

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

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

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

Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *International Conference on Learning Representations*, 2023.

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

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

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

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.

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

Anvita A Srinivas, Tuomas Oikarinen, Divyansh Srivastava, Wei-Hung Weng, and Tsui-Wei Weng. Sand: Enhancing open-set neuron descriptions through spatial awareness. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2993–3002. IEEE, 2025.

Tung-Yu Wu, Yu-Xiang Lin, and Tsui-Wei Weng. And: audio network dissection for interpreting deep acoustic models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 53656–53680, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

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.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

Roland S Zimmermann, Thomas Klein, and Wieland Brendel. Scale alone does not improve mechanistic interpretability in vision models. *Advances in Neural Information Processing Systems*, 36: 57876–57907, 2023.

Appendix

Contents

A Details on bootstrap ensemble

Algorithm 1: BE: Generating a concept prediction set for target neuron

Input: Concept set C, probing dataset D_{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_{probe} with replacement (same size as D_{probe});
  Calculate $\hat{c}_i = \texttt{Identify}(C, f, D_i)$;
end

For each concept $c_j \in C$, count the number of 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 Thm. 3.1

In this section, we prove Thm. 3.1:

Theorem 3.1. *Let c^* be the desired concept for a given neuron and the assumptions above hold for c^*. 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,*

$$\mathbb{P}(c^* \in S) \geq \sum_{i=0}^{K-k(S)-1} \binom{K}{i} p^i (1-p)^{K-i}, \tag{10}$$

where p is the single-trial error probability defined implicitly by the equation $r(f, D_{\text{probe}}, \frac{p}{|C|}) = \frac{\Delta}{2}$.

Proof. We start the proof by estimating single-trial error rate.

Lemma A.1. *Let p be defined implicitly by the equation*

$$r(f, D_{\text{probe}}, \frac{p}{|C|}) = \frac{\Delta}{2}, \tag{11}$$

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

$$\mathbb{P}(\hat{c} = c^*) \geq 1 - p \tag{12}$$

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

$$\begin{aligned} \hat{\mathsf{sim}}(f, c^*; D_{\text{probe}}) &\geq \mathsf{sim}(f, c^*; D_{\text{probe}}) - \frac{\Delta}{2} \\ &\geq \mathsf{sim}(f, c; D_{\text{probe}}) + \frac{\Delta}{2} \quad \text{(Assumption 2)} \\ &\geq \hat{\mathsf{sim}}(f, c; D_{\text{probe}}). \end{aligned} \tag{13}$$

Previously, we show that for many similarity metrics (AUROC, accuracy, IoU, etc.), $r(f, D_{\text{probe}}, \delta) = \mathcal{O}(\sqrt{\frac{-\log \delta}{|D_{\text{probe}}|}})$, i.e. $r(f, D_{\text{probe}}, \delta) \leq Q(\sqrt{\frac{-\log \delta}{|D_{\text{probe}}|}})$ for some constant $Q > 0$. In this case, we can plug in $\delta = \frac{p}{|C|}$ and get

$$\frac{\Delta}{2} = r(f, D_{\text{probe}}, \frac{p}{|C|}) \leq Q\sqrt{\frac{-\log \frac{p}{|C|}}{|D_{\text{probe}}|}}, \tag{14}$$

which gives $p \leq |C| e^{-\frac{\Delta^2}{4Q^2}|D_{\text{probe}}|}$. This shows when probing dataset size $|D_{\text{probe}}|$ and gap between desired concept and other concept Δ becomes larger, the error probability p can be reduced.

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

Theorem A.3. *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,*

$$\mathbb{P}(k^* \geq t) \geq \sum_{i=0}^{t} \binom{K}{i} (1-p)^i p^{K-i} \tag{15}$$

Remark A.4. This could be derived by Lemma A.1 and binomial distribution CDF.

Using Thm. A.3, we can derive:

$$\begin{aligned} \mathbb{P}(c^* \notin S) &\leq \mathbb{P}(k^* \leq K - k(S)) \\ &= 1 - \mathbb{P}(k^* \geq K - k(S) - 1) \\ &\leq 1 - \sum_{i=0}^{K-k(S)-1} \binom{K}{i} (1-p)^i p^{K-i} \end{aligned} \tag{16}$$

Thus,

$$\mathbb{P}(c^* \in S) \geq \sum_{i=0}^{K-k(S)-1} \binom{K}{i} (1-p)^i p^{K-i}, \tag{17}$$

finishes the proof. □

B Details in recall, precision and IoU's convergence speed derivation

In the main text, we mention the key idea of deriving convergence speed r for recall, precision and IoU: that is regard them as special case of accuracy where data are limited in a subgroup. For recall:

$$\begin{aligned} \mathsf{sim}_{\text{recall}}(f, c) &= \frac{\mathbb{P}(f(x) = 1, c(x) = 1)}{\mathbb{P}(c(x) = 1)} \\ &= \mathbb{P}(f(x) = c(x) \mid c(x) = 1). \end{aligned} \tag{18}$$

Therefore, we can regard calculation of recall as a rejection sampling process: The samples satisfying $c(x) = 1$ are kept and others are rejected. Then, accuracy is calculated on remaining samples. Thus, the convergence speed can be calculated by inserting the effective sample size $|\{c(x) = 1 \mid x \in D_{\text{probe}}\}|$ into the accuracy's convergence rate:

$$r_{\text{recall}} = \sqrt{\frac{\log(\frac{2}{\delta})}{2|\{c(x) = 1 \mid x \in D_{\text{probe}}\}|}}. \tag{19}$$

For precision and IoU, the derivation is similar.

C LLM usage

In this article, LLM is used to check grammar and typos as well as improve the writing.