Zero-Shot Model Search via Text-to-Logit Matching

Anonymous authors
Paper under double-blind review

Abstract

With the increasing number of publicly available models, there are pre-trained, online models for many tasks that users require. In practice, users cannot find the relevant models as current search methods are text-based using the documentation which most models lack of. This paper presents ProbeLog, a method for retrieving classification models that can recognize a target concept, such as "Dog", without access to model metadata or training data. Specifically, ProbeLog computes a descriptor for each output dimension (logit) of each model, by observing its responses to a fixed set of inputs (probes). Similarly, we compute how the target concept is related to each probe. By measuring the distance between the probe responses of logits and concepts, we can identify logits that recognize the target concept. This enables zero-shot, text-based model retrieval ("find all logits corresponding to dogs"). To prevent hubbing, we calibrate the distances of each logit, according to other closely related concepts. We demonstrate that ProbeLog achieves high retrieval accuracy, both in ImageNet and real-world fine-grained search tasks, while being scalable to full-size repositories. Importantly, further analysis reveals that the retrieval order is highly correlated with model and logit accuracies, thus allowing ProbeLog to find suitable and accurate models for users tasks in a zero-shot manner.

1 Introduction

Neural networks have revolutionized fields such as computer vision (He et al., 2016; Dosovitskiy, 2020; Redmon, 2016; Li et al., 2023; Rombach et al., 2022) and natural language processing (Touvron et al., 2023; Devlin, 2018; Vaswani, 2017), becoming indispensable tools for many real-world classification tasks. However, their high training cost leaves users with two suboptimal options: i) invest heavily in computational resources for training or fine-tuning a model, ii) settle for a general-purpose model which with substantial inference cost. Now, imagine that instead, one could simply search online for the most accurate model for their specific task and use it directly without additional training. With the rise of large public model repositories, this is becoming feasible. For instance, Hugging Face, the largest existing model repository, hosts over a million models, with more than $100,000$ models added each month. This significantly increases the likelihood of finding a suitable public model for most user tasks. However, the main challenge lies in retrieving the right model for each task. Current model search methods (Shen et al., 2024; Luo et al., 2024) rely on provided metadata or text descriptions, while in practice most models are either undocumented or have very limited descriptions (See Fig. 1), which severely limits these methods ability to retrieve suitable models.

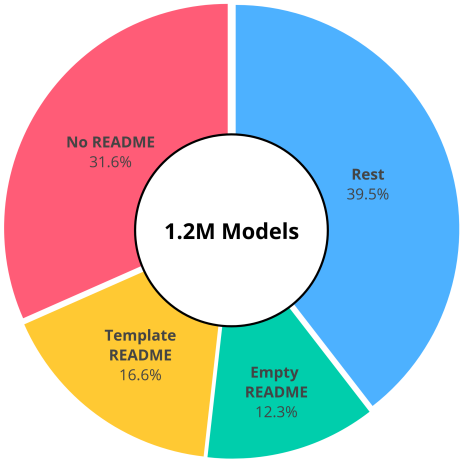

Figure 1: ***HuggingFace Documentation.*** We analyze over $1M$ model cards from Hugging Face, showing that most models are either undocumented or poorly documented.

We aim to search for new models based on their weights, without assuming access to their training data or metadata, as these are often unavailable. More precisely, the goal is to retrieve all classification models

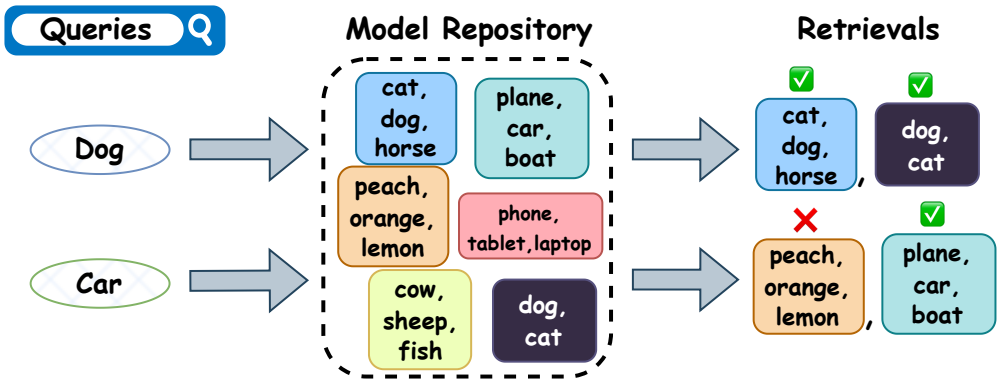

Figure 2: ***Classification Model Search.*** We present a new task of Classification Model Search, where the goal is to find classifiers that can recognize a target concept. Concretely, given an input prompt, such as "Dog", we wish to retrieve all classifiers that one of their classes is "Dog". The search space is a large model repository, which contains many models and concepts to search from. The retrieved models can replace model training, increase accuracy and reduce computational cost.

capable of recognizing a particular concept, such as "Dog". For a solution to be effective and practical, it must meet several requirements: i) identify models that recognize the target concept, regardless of the other concepts they can detect and their order, ii) scale to large model repositories, ii) support text-based search, and iv) retrieve the most accurate models. This task is challenging, as it requires understanding what neurons do. While previous approaches (Oikarinen & Weng, 2023; Bau et al., 2017) have made great progress in zero-shot neuron concept classification, they were not designed for text-based model search.

In this paper, we present *ProbeLog*, a method designed for the new task of Classification Model Search. We begin by analyzing the performance of zero-shot neuron classification methods on model retrieval (search). We show these methods suffer from *hubbing*, where many different queries retrieve only a small fraction of the concept gallery. We therefore propose a simpler approach. Given a set of query probes and a concept name we wish to look for, we compute the i) logit response for each probe and ii) CLIP's cosine similarity between the concept name and each probe. We find that a truncated euclidean distance between logit responses and concept-probe similarities (via CLIP) provides an effective matching metric, while being considerably simpler than the pointwise mutual information used by state-of-the-art zero-shot neuron classification methods. However, in the case of model retrieval, the hubness issue remains: some logits are much closer to many concepts. Therefore, our final method, ProbeLog, proposes a hubness correction term which overcomes this issue. Additionally, we demonstrate that ProbeLog is especially practical for model search, as its retrieval order is highly correlated with the accuracies of the models. Thus, by taking ProbeLog's first retrieval, users can find the most relevant and accurate model for their task. Lastly, we test ProbeLog in a multi-concept search setting and show it can be easily extended to model-level search.

We showcase ProbeLog's effectiveness on two real-world datasets that we curate: one based on models that we train on ImageNet subsets and the other containing models that we download from Hugging Face. Our method is scalable and can handle large models with high effectiveness and efficiency. It achieves high retrieval accuracy, reaching 42.6% top-1 retrieval accuracy when predicting whether an in-the-wild model can recognize a target concept from text.

Our main contributions are:

1. Reevaluating zero-shot logit classification and showing a simple truncated euclidean distance approach compares favorably to the state-of-the-art.

2. Introducing ProbeLog, a method for text-based model search.

3. Introducing 2 new model zoos for this task (including 1300 real model logits from HuggingFace).

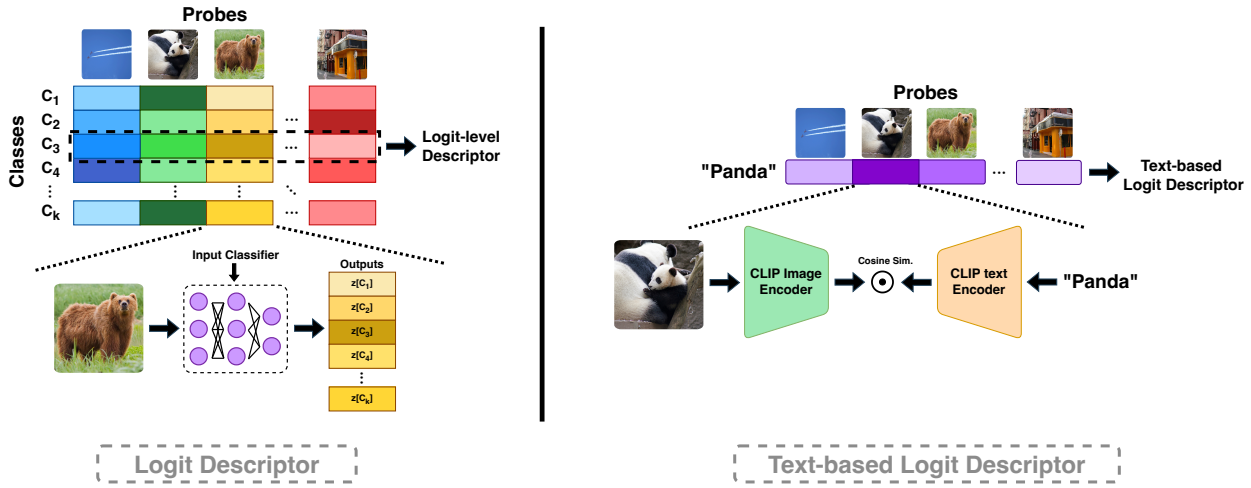

Figure 3: ***ProbeLog Descriptors.*** Our method generates descriptors for individual output dimensions (logits) of models. First, we sample a set of inputs (e.g., from the COCO dataset), and fix them as our set of probes. Then, to create a new logit descriptor, we feed the set of ordered probes into the model and observe the responses on the matching output dimension. Similarly to Oikarinen & Weng (2023), we extend this idea to zero-shot concept descriptors using CLIP, embedding text into the logit descriptors space as the vector of similarities between the text and our set of probes according to CLIP. Lastly, we find that by normalizing both descriptors we can compare them directly.

2 Related Works

2.1 Weight-Space Learning

While neural networks can learn effective representations for many traditional data modalities, effective representations for neural networks are still a work in progress. Unterthiner et al. (2020) took the first step, proposing to observe simple statistics of weights, and use (Ke et al., 2017) on them. Others proposed encoding the weights by modeling the connections between neurons (Navon et al., 2023; De Luigi et al., 2023; Schürholt et al., 2024; 2021; Eilertsen et al., 2020; Lim et al., 2024; Zhou et al., 2024a; Tran et al., 2024; Dupont et al., 2022; Horwitz et al., 2024a). Recent methods (Kofinas et al., 2024; Zhou et al., 2024b; Lim et al., 2023; Kalogeropoulos et al., 2024) model a network as a graph where every neuron is a node, and train permutation-equivariant architectures (Gilmer et al., 2017; Kipf & Welling, 2016; Diao & Loynd, 2022) on these graphs. Probing is an alternative paradigm that encodes the network by observing its outputs on a fixed set of inputs (probes) (Kahana et al., 2024; Herrmann et al., 2024; Carlini et al., 2024; Tahan et al., 2024; Choshen et al., 2022; Kofinas et al., 2024; Huang et al., 2024). Learning on model weights has found many applications including advanced generation abilities (Dravid et al., 2024; Erkoç et al., 2023; Dravid et al., 2024; Shah et al., 2023), model compression (Ha et al., 2016; Ashkenazi et al., 2022; Peebles et al., 2022), model graph recovery (Horwitz et al., 2025; 2024c; Yax et al., 2024), model merging (Yadav et al., 2024; Gueta et al., 2023; Izmailov et al., 2018; Wortsman et al., 2022; Ramé et al., 2023), and even recovering black-box models (Horwitz et al., 2024b; Carlini et al., 2024). Some relevant works search for new adapters for generative models (Shen et al., 2024; Luo et al., 2024; Lu et al., 2023), however these approaches either rely on available metadata or tailored for generative models.

2.2 Interpreting Individual Neurons

Several works attempted to describe the function of individual neurons by visualizations of relevant inputs (Zeiler & Fergus, 2014; Girshick et al., 2014; Mahendran & Vedaldi, 2015; Karpathy et al., 2015). Others use automatic categorization to classify each neuron to a known set of classes (Bau et al., 2017; 2020; Oikarinen & Weng, 2023; Dalvi et al., 2019) or choose natural language to describe neurons (Schwettmann et al., 2021; Hernandez et al., 2021a; Gandelsman et al., 2023; Shaham et al., 2024). In this work, we tackle the problem of model search, where, given a task of choice, one searches for a suitable model for the task inside

a model gallery. We show that neuron classification is highly connected with model search and propose how to successfully adapt previous methods of neuron classification (Oikarinen & Weng, 2023) for retrieval of suitable models.

3 Background and Motivation

3.1 Problem Definition: Model Search

We assume a model repository composed of m classifiers, $f_1, f_2, ..., f_m$. Each classifier f_i can have multiple output dimensions (logits), each corresponding to an unknown concept $l_{i,j}$. The user then inputs a text prompt describing the query concept, c, they wish to search for (e.g., the concept's name). Finally, the goal is to return a model f_i such that one of its classes matches the query concept. Formally, the set of all valid retrieval models, $R(c)$, is defined as:

$$R(c) = \{f_i \ \mid \ \exists \ j \ \ s.t. \ \ l_{i,j} = c\} \tag{1}$$

As mentioned above, the retrieval algorithm does not know the class concepts of each model. We assume access to them solely for evaluation purposes.

3.2 The Challenge: Real Models are Poorly Documented

The existing solution for model search is text-based search in the user-uploaded documentation. To understand the effectiveness of this solution, we explore the level of documentation of models in Hugging Face, the largest model repository. For that, we analyzed $1.2M$ model cards. As shown in Fig. 1, over 30% of all models have no model card at all. Moreover, there are another 28.9% of model cards that are either empty or include an empty automatic template with no information. The remaining 40% of model cards may include some information, however we cannot determine exactly how many of them include relevant information about the training data. As most models are poorly documented, it is important to look for alternative search methods. As all models with API access can be probes, we develop a probing-based approach to model search.

3.3 Network Dissection Methods Struggle on Search

Network dissection methods (Oikarinen & Weng, 2023; Bau et al., 2017; Shaham et al., 2024; Hernandez et al., 2021b) aim to classify the concepts of monosemantic neurons. They first probe the model with a set of curated samples (probes) and identify the probes that highly activate each neuron. Each neuron is then labelled by the main repeating concept across the set of highly activated probes. CLIP-Dissection (Oikarinen & Weng, 2023) proposed zero-shot neuron classification, by choosing the target concept whose mostly activated images have the highest mutual information according to CLIP (Radford et al., 2021). As a first step, we test CLIP-Dissect for model search on a real-world Model Zoo (Schürholt et al., 2022) collected from HuggingFace (see App. E) with 1300 logits spanning more than 500 different concepts. Results are presented in Tab. 1. While CLIP-Dissect reaches decent performance of 35.3% top-1 accuracy, Fig. 4 shows it suffers from severe hubness. I.e., many of the top retrievals come from the same small set of logits. This behavior is highly detrimental to model search; since a logit can only represent one true concept, repeatedly returning the same ones for diverse queries (concept searches) inevitably results in a high rate of incorrect and irrelevant retrievals.

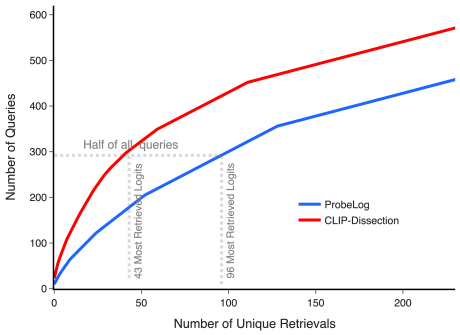

Figure 4: ***Hubness in CLIP-Dissect and ProbeLog.*** Hubness causes many queries to retrieve a handful of unique logits. Here, we show the number of queries covered by the top K logits for both methods. We observe that ProbeLog has significantly fewer hub logits.

Table 1: ***Neuron Classification Results.*** We evaluate the Top-1 and Top-5 neuron classification accuracies of our method and CLIP-Dissection on the INet-Hub (synthetic) and HF-Hub (real world). As both methods are comparable, we conclude our simplified approach is indeed at least as good.

	Top-1 Accuracy		Top-5 Accuracy	
Method	text → INet	text → HF	text → INet	text → HF
CLIP-Dissect (WPMI)	26.0%$_{\pm 0.4}$	44.7%$_{\pm 0.4}$	55.2%$_{\pm 0.4}$	66.2%$_{\pm 0.7}$
CLIP-Dissect (SoftWPMI)	14.7%$_{\pm 0.1}$	37.1%$_{\pm 1.1}$	31.7%$_{\pm 0.4}$	54.7%$_{\pm 1.0}$
All probes + Anti-Hub	13.6%$_{\pm 0.1}$	22.0%$_{\pm 0.4}$	31.6%$_{\pm 0.6}$	37.4%$_{\pm 0.7}$
ProbeLog (Ours)	26.2%$_{\pm 0.2}$	45.3%$_{\pm 0.7}$	52.1%$_{\pm 0.4}$	67.2%$_{\pm 0.2}$

4 Method

In this section, we first present a zero-shot classification method for logits by directly comparing texts and logits. It has similar accuracy as other popular approaches (Oikarinen & Weng, 2023) while being much simpler. We then show how to adapt this method for model search.

4.1 Concept-Logit Truncated Distance

Our objective is to accurately and efficiently find relevant models in a large repository that can recognize a target concept, e.g., "Husky Dog". To do so, we probe the model with several input images and record the logit response for each one. Formally, we input each probe x into the model f, obtaining the response $f(x)[j]$ in the model's j^{th} logit. We denote the descriptor of a logit l from the jth output neural of model f_i (such that $l(x) = f_i(x)[j]\ \forall\ x$) as the responses of all probes at this logit:

$$\phi_{logit}(l) = [l(x_1), l(x_2), \cdots, l(x_n)] = [f_i(x_1)[j], f_i(x_2)[j], \cdots, f_i(x_n)[j]] \tag{2}$$

We also compute the similarity between the concept and each of the probes. We do so using CLIP, by using its text encoder to embed the concept text and its image encoder to embed the image. The cosine similarity of these embeddings provides the similarity between the concept and probe:

$$\phi_{text}(c) = [CLIP(x_1, c), CLIP(x_2, c), \cdots, CLIP(x_n, c)] \tag{3}$$

where $CLIP(x, c) = cosine(E_{img}(x), E_{text}(c))$ such that E_{img}, E_{text} are CLIP's image and text encoders. We illustrate the creation of our zero-shot text-based logit descriptors in Fig. 3. Then, to compare a logit l and text concept c we can simply measure the euclidean distance of the probe responses:

$$d(l, c) = \ell_2(\phi_{logit}(l), \phi_{text}(c)) \tag{4}$$

However, there are two issues with using the euclidean distance. First, probe similarities of logits and CLIP might have very different scales, and must be normalized to compare properly. Secondly, we find that probes which the classifier is not confident about, i.e., probes for which logit responses are low, provide much less information and harm retrieval performance. To combat this, we ignore all probes that have low response values, keeping only the top r probes (in our experiments we choose $r = 50$). Formally, let $a = [a_1, a_2, \ldots, a_n]$ be the sorted probe indices in descending order according to $\phi_{logit}(l)$, our distance measure becomes:

$$d_{trim}(l, c) = \sqrt{\sum_{i=1}^{r} \left(\frac{\phi_{logit}(l)[a_i] - \mu_l}{\sigma_l} - \frac{\phi_{text}(c)[a_i] - \mu_c}{\sigma_c}\right)^2} \tag{5}$$

where μ_l, mu_c and $\sigma_l, sigma_c$ are the mean and standard deviation on the entire logit and concept descriptor (not only the top probes). We call this measure: *truncated euclidean distance*. We compare our simple approach with the more complex pointwise mutual information approaches of CLIP-Dissect, for neuron classification accuracy. Results are presented in Tab. 1. Our truncated distance compares favorably with PMI while being linear and not requiring probabilistic estimation.

4.2 Hubness Calibration

In Sec. 3.2 we showed that a standard neuron classification approach suffers from hubness, where most queries return the same logits (hubs), although more suitable logits exists. To mitigate that, we propose to calibrate the hubness, essentially down weight hubs. This is inspired by techniques from retrieval (Lample et al., 2018). However, at inference time we receive one query at a time, meaning we cannot detect hubs as we see only a single distance from each logit. Therefore, we create a background set of concepts. Specifically, we randomly choose 500 classes from ImageNet-21K (Deng et al., 2009), and use these to calibrate the distances of each logit. Given a query descriptor ϕ_c and gallery logit descriptor, ϕ_l, we compute the truncated distances of ϕ_l to all 500 chosen class descriptors $\phi_{c_1}, ..., \phi_{c_{500}}$. We then subtract the mean of the k smallest distances from $d_{trim}(\phi'_c, \phi_c)$. Formally, let $b = [b_1, b_2, \ldots, b_n]$ be the indices of the sorted truncated distances in descending order. Our calibration then becomes:

$$d_{trim}(l, c) \leftarrow d_{trim}(l, c) - \frac{1}{k} \sum_{i=1}^{k} d_{trim}(l, c_{b_i}) \tag{6}$$

Intuitively this means that if the logit represents the class "Cat" we ask whether this logit is more similar to "Cat" than "Tiger", "Puma" or "Lynx", rather than just asking whether this is a "Cat" against all classes, which could aid in detecting fine-grained concepts.

4.3 ProbeLog Retrieval Pipeline

We summarize the end-to-end ProbeLog retrieval approach. Given a large repository of models and a fixed set of probes, we first pre-compute the logit descriptors for all available output dimensions by recording their responses to the probes. When a user queries a target concept c, we compute its text-based descriptor using CLIP's similarity to the same probes. We then measure the truncated Euclidean distance between the query descriptor and all pre-computed logit descriptors. Finally, we apply our hubness calibration to these distances using a background set of concepts, and retrieve the models containing the logit with the lowest calibrated distance.

4.4 Relation to Current Concept-Logit Similarity Measures

In this section, we show that our truncated distance strikes a good balance between previous linear and non-linear similarity measures. Previously proposed linear measures used the inner product of the concept and logit descriptor, however this was shown (Oikarinen & Weng, 2023) to fail. WPMI works much better but is much more complex:

$$\text{wpmi}(c, l) = \sum_{i=1}^{r} \log p(c|x_{a_i}) - \lambda \log \sum_{a' \in A} \left(\Pi_{i=1}^{r} p(c|x_{a'_i}) \right) + \lambda \log |C| \tag{7}$$

Computing this quantity requires estimating the probability of the concept given the probe, which is difficult to do precisely. It also assumes independence of between the probes which might not always be true. Additionally, its normalization term is quite complex and includes a combinatorial sum of subparts. The soft weighted PMI is even more complex and is detailed in Appendix A of (Oikarinen & Weng, 2023). In contrast, our truncated distance method is much simpler and does not suffer from the above limitations.

Table 2: ***Retrieval Results.*** We evaluate the Top-1 and Top-5 retrieval accuracies of our method and the baselines for text-based retrievals and logit classification. All methods use COCO images as probes. For a fair comparison, all experiments are performed with $4,000$ probes.

	Top-1 Accuracy		Top-5 Accuracy	
Method	text $\rightarrow$ INet	text $\rightarrow$ HF	text $\rightarrow$ INet	text $\rightarrow$ HF
CLIP-Dissect (WPMI)	44.9%$_{\pm 0.4}$	35.3%$_{\pm 1.2}$	67.1%$_{\pm 0.8}$	50.7%$_{\pm 1.6}$
CLIP-Dissect (SoftWPMI)	42.1%$_{\pm 0.6}$	35.1%$_{\pm 1.3}$	64.1%$_{\pm 0.5}$	49.2%$_{\pm 0.9}$
All probes + Anti-Hub	58.7%$_{\pm 1.3}$	31.8%$_{\pm 1.2}$	78.9%$_{\pm 0.7}$	49.4%$_{\pm 0.3}$
ProbeLog (Ours)	**64.4%**$_{\pm 0.4}$	**42.6%**$_{\pm 1.1}$	**82.5%**$_{\pm 0.7}$	**60.1%**$_{\pm 0.9}$

It also vastly improves over simple inner products by: i) normalizing the logit and concept features, so they operate in the same scale, and ii) computing the inner product on just the top r probes. Our method therefore strikes a better balance between simplicity and performance.

5 Experiments

5.1 Experimental Setting

Datasets. As there are no suitable existing datasets for model search that include ground-truth data, we created 2 new ones, INet-Hub and HF-Hub. For each model in the INet-Hub (see App. D), we sample a subset of ImageNet classes, a model architecture and foundation model initialization checkpoint. We then train the model on the selected data. The final dataset consists of $1,500$ models, making $85,000$ logits, derived from 1000 unique fine-grained concepts (see App. D). Our second hub, HF-Hub, is a set of 1300 real-world model logits (collected from over 250 different models) downloaded from HuggingFace (see App. E).

Baselines. We test our retrieval algorithm against two baselines: (i) CLIP-Dissect (Oikarinen & Weng, 2023), which is designed for neuron classification, and measures how confident is CLIP on the mostly activated probes of each neuron. WPMI and SoftWPMI are two variants of CLIP-Dissect which differ in their weighting of the probes in the mutual information calculation. (ii) normalized euclidean distance using all probes, instead of truncating the top-r probes as in our approach. Here we also include our anti-hubness calibration.

Metrics. We first note that our ground-truth labels are the concepts each classifier logit is associated with, and that these are only used for evaluation purposes (they are not available to ProbeLog during search). To evaluate retrieval performance we use the standard top-1 and top-5 accuracies. Specifically, we define top-k accuracy as a "hit-at-k" metric, which measures the percentage of query concepts with a relevant logit in any of their top-k retrievals. In cases where several models are trained with the same concept and multiple options are available, we prefer to retrieve the most accurate one (for further analysis on retrieved classifiers accuracy see Sec. 5.4).

5.2 Model Search Results

We evaluate our method on the 2 Model Zoos and present the results in Tab. 2. We report both the top-1 and top-5 accuracies. Here, we evaluate model retrieval performance, i.e., search-by-text evaluation where we search for the closest retrievals to a zero-shot text descriptor in either the INet-Hub or the HF-Hub. We can see that in both cases, our approach greatly exceeds the baselines, reaching an impressive top-1 accuracy of 64.4% on the INet-Hub. Moreover, when tested on the HF-Hub we can see that our method generalizes to real-world models, as it finds suitable matches for more than a 40% of the queries in the first search result, and for more than 60% of queries within the first 5 retrievals. This shows that while simple, our approach can generalize to real-world scenarios. In Sec. 5.8 we demonstrate that our method can also match between images and model logits, allowing to search for concepts which are difficult to describe by text.

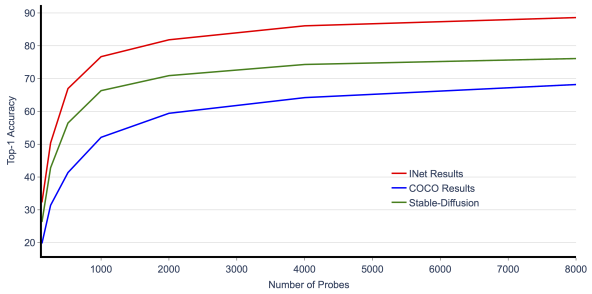

Figure 5: ***Number of Probes.*** We test ProbeLog on the INet-Hub with increasing numbers of probes. While more probes lead to higher accuracy, the gains are diminishing.

Figure 6: ***ProbeLog Semantic False Retrievals.*** We visualize 4 random retrieval errors and see the errors are generally semantic.

Table 3: ***Multi-Concept Search.*** We evaluate model-level retrieval, which searches for an entire list of query concepts ($3-10$ concepts) at once. ProbeLog significantly outperforms the baselines in this setting.

	Top-1 Accuracy		Top-5 Accuracy	
Method	text $\rightarrow$ INet	text $\rightarrow$ HF	text $\rightarrow$ INet	text $\rightarrow$ HF
All Probes	9.5%$\pm$0.6	30.6%$\pm$0.7	21.4%$\pm$0.5	55.6%$\pm$1.8
WPMI	7.0%$\pm$0.6	45.5%$\pm$1.1	15.6%$\pm$0.6	69.7%$\pm$0.4
SoftWPMI	6.3%$\pm$0.5	44.9%$\pm$1.5	14.0%$\pm$0.4	68.1%$\pm$0.8
ProbeLog (Ours)	**15.3**%$\pm$1.1	**56.6**%$\pm$2.3	**26.6**%$\pm$0.4	**81.0**%$\pm$0.7

5.3 Errors or Near-Misses?

Retrieval mistakes can vary in severeness, e.g. given a query concept of "German Shepherd" a retrieval of "Husky Dog" is more forgivable than "Pickup Truck". To qualitatively evaluate the severity of our mistakes we randomly sample a few random wrong retrievals and plot the images matching to their concepts. We visualize 4 such random retrievals in Fig. 6. For more uncurated retrieval examples see App. F. We can see that ProbeLog's mistakes are often near misses rather than random errors. E.g., its mistake for "Green Mamba" is simply another similar snake species.

5.4 Retrieved Models Accuracy

While ProbeLog shows impressive results for retrieving a logit with the right concept, it is interesting to evaluate how well the returned models recognize the query class. To test that, we evaluate the first correct retrieval for each query on the INet-Hub using precision-recall Area Under Curve (PR-AUC). We emphasize that this PR-AUC evaluates the found model's logit, not the retrieval process itself. We test the retrieved logits in a one-vs-all setting on the entire ImageNet test set, where samples are labeled 1 for the query class and 0 for all other ImageNet classes. We use PR-AUC in this setup to provide a uniform metric across models trained on varying numbers of classes. Note, this means many of the samples used are out-of-distribution to the model, as each model was trained on a subset of ImageNet classes.

We compare the PR-AUC of ProbeLog retrieved models against the zero-shot PR-AUC obtained by CLIP in the same setting. I.e., we score the ImageNet test set according to the text prompt description of the class in question, and compute the PR-AUC of CLIP itself. Although large language models might score better than CLIP in this setting, we do not include them in our comparison, as they are much more computationally demanding, having $1000\times$ as many parameters as the specialized classification models. The results are presented in Tab. 4. We can clearly see that ProbeLog finds models that are much more accurate than the average model available on the INet-Hub as well as CLIP itself, by a large margin. Moreover, to test whether ProbeLog ranks accurate models higher, we present the average PR-AUC over the retrieval ranks in Fig. 7. We can see that indeed better models are returned first, demonstrating that to use ProbeLog, one can simply

Table 4: ***PR-AUC of Retrieved Logits.*** ProbeLog's top retrievals are much better than the average model in the repository and CLIP zero-shot binary classification.

	PR-AUC
Mean INet-Hub	0.433
CLIP Zero-shot	0.421
ProbeLog	**0.647**

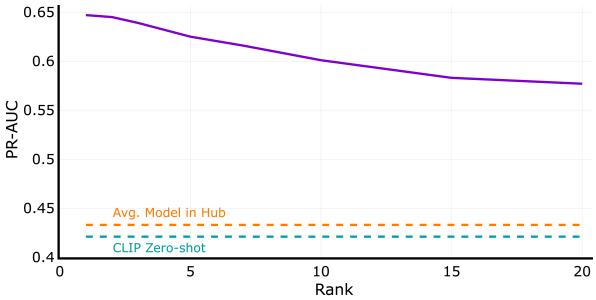

Figure 7: ***PR-AUC vs. Retrieval Rank.*** ProbeLog retrieves accurate logits first. For full models accuracies see App. A.

take the its top-1 retrieval. App. A provides a similar analysis on model accuracy, showing that ProbeLog's top retrievals arrive from highly accurate models (surpassing CLIP by up to 8.9% on average).

5.5 Multi-Concept Search

Single-concept search serves as a fundamental building block that allows users to build complex queries by intersecting single-concept retrievals. However, real-world scenarios often require a single model capable of recognizing multiple specific concepts simultaneously (e.g., cats, dogs, and cars). To demonstrate ProbeLog's ability to extend to this broader setting, we evaluate it on multi-concept search. We randomly selected 500 query lists, each containing a random subset of 3 to 10 concepts known to co-exist in at least one model within the repository. We then adapt ProbeLog for a model-level search: given a list of query concepts, we find the closest unique logit in a candidate model to each concept in the query list, and rank the candidate models based on their mean distance across all queried concepts. A retrieval is considered a correct "hit" only if the retrieved model contains the *entire* queried list of concepts in its ground-truth classes.

As presented in Tab. 3, ProbeLog can be effectively extended to the realistic multi-class retrieval task. It significantly outperforms the baselines, achieving a 56.6% top-1 accuracy on the real-world HF-Hub compared to WPMI's 45.5%. This highlights the applicability of our text-to-logit matching framework for real-world search scenarios.

5.6 Hubbing Analysis

We evaluate ProbeLog's hubness similarly to the evaluation in Sec. 3.2, checking whether a small set of logits are still responsible for the most returns. We provide the results in Fig. 4. We see that indeed hubness is substantially reduced, and more logits are included in the top-1 returns. Specifically, while in CLIP-Dissection the 43 biggest hubs were responsible for over half the queries (300), in ProbeLog it takes almost a hundred smaller hubs.

5.7 Ablation Studies

Key Method Ablations. We perform an ablation study evaluating the two core components of ProbeLog: the hubness calibration (Anti-Hub) and the truncated distance metric (using only the top-r activating probes). As shown in Tab. 5, removing the Anti-Hub calibration results in a drastic drop in performance across all datasets and metrics. For instance, the top-1 accuracy of full ProbeLog on the INet-Hub collapses from 64.4% to just 0.5% without the Anti-Hub component. This demonstrates the severe practical harm of the hubbing phenomenon on the model retrieval task. Furthermore, comparing the use of all probes against our truncated distance mechanism demonstrates that filtering out unconfident probes heavily improves performance, increasing the top-1 accuracy on the HF-Hub from 31.8% to 42.6%. These results confirm that both components are vital for achieving high-accuracy, zero-shot classification model search.

Table 5: ***Method Ablations***. We evaluate the impact of the hubness calibration and the truncated distance metric. Both components are crucial for achieving high retrieval accuracy.

Method	Top-1 Accuracy		Top-5 Accuracy	
	text → INet	text → HF	text → INet	text → HF
All Probes w.o. Anti-Hub	11.6% $\pm$0.2	20.9% $\pm$0.3	15.7% $\pm$0.2	38.9% $\pm$0.3
All Probes + Anti-Hub	58.7% $\pm$1.3	31.8% $\pm$1.2	78.9% $\pm$0.7	49.4% $\pm$0.3
ProbeLog w.o. Anti-Hub	0.5% $\pm$0.1	7.9% $\pm$1.1	1.5% $\pm$0.3	17.9% $\pm$0.6
ProbeLog (Ours)	**64.4%** $\pm$0.4	**42.6%** $\pm$1.1	**82.5%** $\pm$0.7	**60.1%** $\pm$0.9

Table 6: ***Dataset Ablations.*** We compare both real and synthetic probe distributions. While distributions closer to the model's training data lead to better results, even out-of-distribution probes sampled from the COCO dataset retrieve relevant logits with high accuracy.

Method	Top-1 Accuracy		Top-5 Accuracy	
	text → INet	text → HF	text → INet	text → HF
Dead-Leaves	4.1% $\pm$0.3	2.8% $\pm$0.3	11.4% $\pm$0.5	8.2% $\pm$0.6
Stable-Diffusion	73.9% $\pm$0.6	54.2% $\pm$0.9	89.9% $\pm$0.7	72.8% $\pm$0.7
ImageNet	86.1% $\pm$0.6	57.3% $\pm$0.5	95.3% $\pm$0.3	75.3% $\pm$1.4
COCO	64.4% $\pm$0.4	42.6% $\pm$1.1	82.5% $\pm$0.7	60.1% $\pm$0.9

How to select the probe distribution? We showed (Sec. 5.2) that ProbeLog can generalize to real-world scenarios. Here, we conduct an ablation study, to test the effect of sampling probes from different distributions: (i) Dead-Leaves (Baradad Jurjo et al., 2021; Lee et al., 2001): a very coarse, hand-crafted generative model. (ii) ImageNet images. (iii) StableDiffusion (Rombach et al., 2022) samples using prompts of ImageNet-21K objects. (iv) COCO Images. Results, shown in Tab. 6, demonstrate a consistent pattern: probes sampled from distributions that are closer to the target concept obtain more accurate retrievals. However, we note that even quite different probe distribution can yield high retrieval accuracies. E.g., even though COCO images are typically of scenes rather than objects, they are effective probes, reaching a top-1 accuracy of more than 60% when searching the INet-Hub by text. These results show that defining a general set of probes, which can retrieve a wide range of concepts is feasible. However, if there is access to probes from target concept's distribution, it is better to use them. In App. C, we further quantify this domain gap by evaluating ProbeLog on 10 diverse probe datasets, demonstrating a strong negative correlation between retrieval accuracy and the probe set's distance to the target domain.

How many probes are enough? Fig. 5 presents the results of text retrieval on INet-Hub using increasing numbers of probes. More probes lead to better results but with diminishing gains. For example, $4,000$ COCO probes achieve good performance of 64.4% top-1 accuracy, though it is possible to achieve a 68.2% using $8,000$ probes. We then ablate how many probes should be taken into account from each logit (r). Fig. 8 shows the top-1 accuracy on the INet-Hub against r, the number of top activating probes taken from each query. We can clearly see a peak when choosing $r = 50$, which aligns with our intuition: selecting too many probes also includes irrelevant ones, conversely, selecting too few probes adds too much variance to the estimate of the truncated distance.

5.8 Model Retrieval via Image Queries

In many cases, describing an abstract visual concept using an example is much easier than describing the same concept by text. We therefore extend our search approach to a search-by-image setting. In this setting the query is a small set of images, and the expected response is a model logit that recognizes the underlying concept of the query image set (evaluated as an exact match to the ground-truth ImageNet class these images were sampled from). This allows users to search for complex concepts using known samples, rather than a long text description.

Table 7: ***Search-by-Image Retrieval Results.*** We evaluate Top-1/5 retrieval accuracies of searching by image queries. s denotes the number of query images. All methods use COCO images as probes. For a fair comparison, all experiments are performed with $4,000$ probes.

	Top-1 Accuracy			Top-5 Accuracy		
Method	$s=1$	$s=5$	$s=10$	$s=1$	$s=5$	$s=10$
CLIP-Dissect (WPMI)	11.9%	18.6%	20.1%	24.1%	28.8%	31.3%
CLIP-Dissect (SoftWPMI)	9.4%	15.2%	16.0%	18.5%	28.1%	29.0%
ProbeLog (Ours)	**23.5%**	**49.2%**	**55.3%**	**41.1%**	**70.4%**	**76.4%**

We adapt ProbeLog to accept such a small set of images as its input query. We denote CLIPs (Radford et al., 2021) image encoder as E_{img} and the input set of query images as $q_1, ..., q_s$. We then probe CLIPs image encoder with each one of the input images $E_{img}(q_1), ..., E_{img}(q_s)$ and treat the average representation $c_q = \frac{1}{s}\Sigma_i E_{img}(q_i)$ as the representation of the main concept of the image set according to CLIP. Obtaining the similarities vector of the set with each probe is performed similarly to the text setting. Formally,

$$\phi_{image_set} = [< x_1, c_q >, ..., < x_n, c_q >] \tag{8}$$

Lastly, retrieval proceeds normally. We evaluate this approach on the INet-Hub where we have access to the distribution of images matching to each logit. I.e., we sample k ImageNet (Deng et al., 2009) images from the query concept class, and use these as the query set for the retrieval. We compare ProbeLog to CLIP-Dissect (Oikarinen & Weng, 2023). We experiment with a query image set sizes of $s = 1, s = 5$ and $s = 10$. Results are presented in Tab. 7.

6 Discussion

Non-random probe selection. We proposed an approach for searching models that can recognize a target concept. Our approach probes each model with $4,000$ COCO images to produce the representation of each logit. However, we believe this number can be reduced substantially. For instance, while we chose the set of probes at random, it is likely that a smaller and more curated of probes exists. Specifically, core-set methods, which aim to reduce the number of training data, could potentially reduce this number. Another direction is to use advanced collaborative filtering ideas which take into account the statistics of logit values. We believe this is a fruitful avenue for future research.

7 Limitations

Scaling-up to entire repositories. While our model zoos already have $1,500$ large models, including ViTs (Dosovitskiy, 2020) and RegNet-Ys Radosavovic et al. (2020), model repositories may contain millions of models. We tested our approach on smaller hubs mainly because we did not have the resources to probe and label a million models. However, this investment is feasible by industry standards: Using a single RTX-A5000 GPU we were able to probe each model with $4,000$ images in ~ 12 seconds on average. Meaning, probing a million classification models would require ~ 3300 GPU hours. After this one-time effort, probing new models uploaded to HuggingFace is relatively simple and requires just a single GPU dedicated for the task ($\sim 12M$ function evaluations a day). Having the representations for all models, the search is fast as our search algorithm operates in a space of a few tens of dimensions, where retrieval from even a billion entries is possible (Johnson et al., 2019; Jayaram Subramanya et al., 2019; Chen et al., 2021). Moreover, the descriptors are much lighter than actual model weights, and storing them is quite cheap. E.g., our INet-Hub model take $400GB$ of memory, but their logit descriptors for $8,000$ probes only consume $1.4GB$ of storage.

Extension beyond classification models. Our proposed method embeds each logit of each model on its own. This will require modifications for generative models where the output dimensions do not explicitly encode the learned concepts. While some works attempted to search for generative adapters (Lu et al., 2023), they typically required many more $(50,000)$ probes as their descriptors summarize the distribution of outputs. We believe that our approach, where the inputs are fixed and ordered, can reduce the number of probes substantially.

Out-of-distribution concepts. To enable search for diverse concepts we sampled probes from the COCO dataset (Lin et al., 2014) which does not contain just centered objects but also entire scenes. Still, these probes do not represent all concepts, e.g. medical concepts. Finding far OOD concepts will require a probe distribution that is better aligned to these concepts.

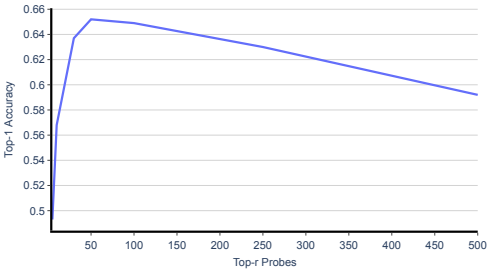

Figure 8: ***Number of Considered Probes in a Logit.*** We test different numbers of top-activating probes kept from each logit. Taking 50 probes performs best. Too many probes includes some irrelevant ones to the logit, while too few increases the variance of the truncated distance estimate.

8 Conclusion

In this paper we propose an approach for searching for models in large repositories that can recognize a target concept. We first probe all models with a fixed, ordered set of probes, and define the values from each output dimension (logit) across all probes as a descriptor. We proposed a truncated euclidean distance score to compare between logits and text. Then, by calibrating the distances of each logit according to a background concept set we mitigate the hubness issue, allowing to search models by text. We evaluate our approach on real-world models, and show it generalizes well to in-the-wild models collected from HuggingFace. Our method retrieves models that are significantly more accurate than zero-shot CLIP, and ranks models according to accuracy on the query task.

References

Maor Ashkenazi, Zohar Rimon, Ron Vainshtein, Shir Levi, Elad Richardson, Pinchas Mintz, and Eran Treister. Nern–learning neural representations for neural networks. *arXiv preprint arXiv:2212.13554*, 2022.

Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems*, 34:2556–2569, 2021.

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*, pp. 6541–6549, 2017.

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. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, et al. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*, 2024.

Qi Chen, Bing Zhao, Haidong Wang, Mingqin Li, Chuanjie Liu, Zengzhong Li, Mao Yang, and Jingdong Wang. Spann: Highly-efficient billion-scale approximate nearest neighborhood search. *Advances in Neural Information Processing Systems*, 34:5199–5212, 2021.

Leshem Choshen, Elad Venezian, Shachar Don-Yehia, Noam Slonim, and Yoav Katz. Where to start? analyzing the potential value of intermediate models. *arXiv preprint arXiv:2211.00107*, 2022.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6309–6317, 2019.

Luca De Luigi, Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Deep learning on implicit neural representations of shapes. *arXiv preprint arXiv:2302.05438*, 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*, pp. 248–255. Ieee, 2009.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Cameron Diao and Ricky Loynd. Relational attention: Generalizing transformers for graph-structured tasks. *arXiv preprint arXiv:2210.05062*, 2022.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Amil Dravid, Yossi Gandelsman, Kuan-Chieh Wang, Rameen Abdal, Gordon Wetzstein, Alexei A Efros, and Kfir Aberman. Interpreting the weight space of customized diffusion models. *arXiv preprint arXiv:2406.09413*, 2024.

Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*, 2022.

Gabriel Eilertsen, Daniel Jönsson, Timo Ropinski, Jonas Unger, and Anders Ynnerman. Classifying the classifier: dissecting the weight space of neural networks. In *ECAI 2020*, pp. 1119–1126. IOS Press, 2020.

Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14300–14310, 2023.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting clip's image representation via text-based decomposition. *arXiv preprint arXiv:2310.05916*, 2023.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Almog Gueta, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. Knowledge is a region in weight space for fine-tuned language models. *arXiv preprint arXiv:2302.04863*, 2023.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021a.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021b.

Vincent Herrmann, Francesco Faccio, and Jürgen Schmidhuber. Learning useful representations of recurrent neural network weight matrices. *arXiv preprint arXiv:2403.11998*, 2024.

Eliahu Horwitz, Bar Cavia, Jonathan Kahana, and Yedid Hoshen. Representing model weights with language using tree experts. *arXiv preprint arXiv:2410.13569*, 2024a.

Eliahu Horwitz, Jonathan Kahana, and Yedid Hoshen. Recovering the pre-fine-tuning weights of generative models. In *ICML*, 2024b. URL `https://openreview.net/forum?id=761UxjOTHB`.

Eliahu Horwitz, Asaf Shul, and Yedid Hoshen. On the origin of llamas: Model tree heritage recovery. *arXiv preprint arXiv:2405.18432*, 2024c.

Eliahu Horwitz, Nitzan Kurer, Jonathan Kahana, Liel Amar, and Yedid Hoshen. Charting and navigating hugging face's model atlas. *arXiv preprint arXiv:2503.10633*, 2025.

Qihan Huang, Jie Song, Mengqi Xue, Haofei Zhang, Bingde Hu, Huiqiong Wang, Hao Jiang, Xingen Wang, and Mingli Song. Lg-cav: Train any concept activation vector with language guidance. *arXiv preprint arXiv:2410.10308*, 2024.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. Diskann: Fast accurate billion-point nearest neighbor search on a single node. *Advances in Neural Information Processing Systems*, 32, 2019.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Jonathan Kahana, Eliahu Horwitz, Imri Shuval, and Yedid Hoshen. Deep linear probe generators for weight space learning. *arXiv preprint arXiv:2410.10811*, 2024.

Ioannis Kalogeropoulos, Giorgos Bouritsas, and Yannis Panagakis. Scale equivariant graph metanetworks. *arXiv preprint arXiv:2406.10685*, 2024.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Miltiadis Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J Burghouts, Efstratios Gavves, Cees GM Snoek, and David W Zhang. Graph neural networks for learning equivariant representations of neural networks. *arXiv preprint arXiv:2403.12143*, 2024.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, Dec 2013. doi: 10.1109/ICCVW.2013.77.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International conference on learning representations*, 2018.

Ann B Lee, David Mumford, and Jinggang Huang. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision*, 41:35–59, 2001.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Derek Lim, Haggai Maron, Marc T Law, Jonathan Lorraine, and James Lucas. Graph metanetworks for processing diverse neural architectures. *arXiv preprint arXiv:2312.04501*, 2023.

Derek Lim, Yoav Gelberg, Stefanie Jegelka, Haggai Maron, et al. Learning on loras: Gl-equivariant processing of low-rank weight spaces for large finetuned models. *arXiv preprint arXiv:2410.04207*, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Daohan Lu, Sheng-Yu Wang, Nupur Kumari, Rohan Agarwal, Mia Tang, David Bau, and Jun-Yan Zhu. Content-based search for deep generative models. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–12, 2023.

Michael Luo, Justin Wong, Brandon Trabucco, Yanping Huang, Joseph E Gonzalez, Zhifeng Chen, Ruslan Salakhutdinov, and Ion Stoica. Stylus: Automatic adapter selection for diffusion models. *arXiv preprint arXiv:2404.18928*, 2024.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning*, pp. 25790–25816. PMLR, 2023.

M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Tuomas Oikarinen and Tsui-Wei Weng. CLIP-dissect: Automatic description of neuron representations in deep vision networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=iPWiwWHc1V`.

William Peebles, Ilija Radosavovic, Tim Brooks, Alexei A Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints. *arXiv preprint arXiv:2209.12892*, 2022.

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*, pp. 8748–8763. PMLR, 2021.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 28656–28679. PMLR, 2023.

J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Konstantin Schürholt, Dimche Kostadinov, and Damian Borth. Self-supervised representation learning on neural network weights for model characteristic prediction. *Advances in Neural Information Processing Systems*, 34:16481–16493, 2021.

Konstantin Schürholt, Diyar Taskiran, Boris Knyazev, Xavier Giró-i Nieto, and Damian Borth. Model zoos: A dataset of diverse populations of neural network models. *Advances in Neural Information Processing Systems*, 35:38134–38148, 2022.

Konstantin Schürholt, Michael W Mahoney, and Damian Borth. Towards scalable and versatile weight space learning. *arXiv preprint arXiv:2406.09997*, 2024.

Sarah Schwettmann, Evan Hernandez, David Bau, Samuel Klein, Jacob Andreas, and Antonio Torralba. Toward a visual concept vocabulary for gan latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6804–6812, 2021.

Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. *arXiv preprint arXiv:2311.13600*, 2023.

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.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.

Shir Ashury Tahan, Ariel Gera, Benjamin Sznajder, Leshem Choshen, Liat Ein Dor, and Eyal Shnarch. Label-efficient model selection for text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8384–8402, 2024.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Viet-Hoang Tran, Thieu N Vo, An Nguyen The, Tho Tran Huu, Minh-Khoi Nguyen-Nhat, Thanh Tran, Duy-Tung Pham, and Tan Minh Nguyen. Equivariant neural functional networks for transformers. *arXiv preprint arXiv:2410.04209*, 2024.

Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya Tolstikhin. Predicting neural network accuracy from weights. *arXiv preprint arXiv:2002.11448*, 2020.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36, 2024.

Nicolas Yax, Pierre-Yves Oudeyer, and Stefano Palminteri. Phylolm: Inferring the phylogeny of large language models and predicting their performances in benchmarks. *arXiv preprint arXiv:2404.04671*, 2024.

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

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. *Advances in neural information processing systems*, 36, 2024a.

Allan Zhou, Kaien Yang, Yiding Jiang, Kaylee Burns, Winnie Xu, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Neural functional transformers. *Advances in neural information processing systems*, 36, 2024b.

A Retrieval Rank vs. Model Accuracy

We provide additional analysis, comparing the retrieved models accuracies vs CLIP's zero shot accuracy, at the model-level. For each query text, we retrieve the rank-k logit, and select the model it is a part of. We compare the accuracy of this model to CLIP zero-shot accuracy on the model's task and test set. The final number is averaged over all query texts. Results are presented in Tab. 8. It is clear that the retrieved models are substantially more accurate than zero-shot CLIP.

Table 8: ***Model Accuracy by Retrieval Rank.*** We compate ProbeLog retrieved models accuracy to CLIP zero-shot accuracy. It is clear that the top retrieved models significantly outperform CLIP.

Rank	1	2	3	5	7	10	15	20
Retrieved Models	92.7%	92.5%	92.3%	91.7%	91.6%	90.8%	89.8%	89.4%
CLIP	83.8%	83.7%	83.8%	83.8%	83.7%	83.8%	83.8%	83.9%

B Background Set Size

As additional analysis, we wish to test the required background set size needed for ProbeLog. We therefore evaluated text-based search with background sets of different sizes, taken from ImageNet21k classes. For each size, we sampled 5 non-overlapping sets. We then report the mean and standard deviation for each size below. Our analysis (presented in Tab. 9) reveals that larger sizes do achieve better results but this saturates around 500 concepts. We note, that the one-time cost of computing 500 CLIP text encodings requires fewer resources than a single model probing. Therefore, using a background set of this size is highly practical.

Table 9: ***Top-1 Accuracy vs. Background Set Size.*** ProbeLog's retrieval accuracy improves with larger background sets across both HuggingFace Hub and ImageNet Hub repositories.

Background Set Size	HF-Hub Top-1 Acc.	INet-Hub Top-1 Acc.
5	31.1±1.7%	44.8±3.6%
25	33.2±0.8%	48.0±1.5%
50	33.4±0.8%	49.3±1.8%
100	38.3±0.5%	57.2±1.2%
250	41.7±1.0%	62.9±1.6%
500	43.6±0.9%	65.2±0.3%
1000	44.3±1.0%	66.9±0.5%
2000	44.8±0.9%	67.2±0.9%

C Probe Set and Models Training Data

To further understand how to select the probing set, we conducted an additional experiment, where we tested if the similarity between the probe set and training set of the models affects the results. We tried 10 datasets as probe candidates, and computed the Frechet Distance (via DINO embeddings) from each one to ImageNet. We take ImageNet as the super-set of training data for our INet-Hub models. We then sampled probes from each of the 10 datasets, and computed text-based retrieval accuracy on the INet-Hub using them (averaged over 5 seeds). We present the results in Tab. 10. We highlight that the overall correlation of the DINO-FD and Top-1 accuracy is -0.920, showing that indeed the closer the probe dataset to the training data of the correct model (and thus to the target concept itself), the better the retrieval.

D INet-Hub Dataset Details

To simulate a model hub with many classifiers, we train $1,500$ classifier models on different subsets of ImageNet classes. Each classifier is trained on a subset of between 15 and 200 classes, where the classes

Table 10: ***Performance vs. Dataset Domain Distance.*** ProbeLog's retrieval accuracy increases as the Fréchet Distance from ImageNet decreases. This shows that smaller domain shift between the models training data and the probe set leads to better retrieval results.

Dataset	ImageNet FD	Top-1 Acc.
ImageNet (Deng et al., 2009)	0.0	86.1%
COCO (Lin et al., 2014)	1347.7	64.4%
SD (Rombach et al., 2022)	1500.9	73.9%
Food101 (Bossard et al., 2014)	4960.7	22.3%
OxfordFlowers (Nilsback & Zisserman, 2008)	6712.1	15.7%
CIFAR100 (Krizhevsky et al., 2009)	7903.8	42.3%
StanfordCars (Krause et al., 2013)	7742.9	12.3%
GTRSB (Stallkamp et al., 2011)	8437.1	2.9%
EuroSAT (Helber et al., 2019)	8813.3	5.7%
DeadLeaves (Baradad Jurjo et al., 2021)	10363.9	4.1%

are chosen at random separately for each model. 90% of the classifiers are initialized from a foundation model, and the rest 10% are trained from scratch. The pre-training weights are selected from a set of 49 different models spanning various architectures including ViTs (Dosovitskiy, 2020), ResNets (He et al., 2016), RegNet-Ys (Radosavovic et al., 2020), MLP Mixers (Tolstikhin et al., 2021), EfficientNets (Tan & Le, 2019), ConvNexts (Liu et al., 2022) and more. Each model is then trained for $2-5$ epochs. This process results in a model hub with over $85,000$ different logits to search for and $1,000$ different fine-grained concepts. Below we list the possible pre-training weights of each model. All pre-training weights are taken from the timm library (Wightman, 2019).

- vit_base_patch32_clip_quickgelu_224.laion400m_e32
- vit_base_patch32_clip_224.laion400m_e32
- vit_base_patch32_clip_224.laion2b
- vit_base_patch32_clip_224.datacompxl
- convnext_base.clip_laiona
- convnext_base.clip_laion2b
- vit_base_patch32_clip_quickgelu_224.metaclip_400m
- vit_base_patch32_clip_quickgelu_224.metaclip_2pt5b
- vit_base_patch32_clip_224.metaclip_400m
- vit_base_patch32_clip_224.metaclip_2pt5b
- vit_base_patch32_clip_224.openai
- seresnextaa101d_32x8d.sw_in12k
- resmlp_24_224.fb_dino
- resmlp_12_224.fb_dino
- mixer_l16_224.goog_in21k
- mixer_b16_224.miil_in21k
- mixer_b16_224.goog_in21k

- resnetv2_152x2_bit.goog_in21k
- resnetv2_101x1_bit.goog_in21k
- resnetv2_50x1_bit.goog_in21k
- regnety_320.seer
- regnety_160.sw_in12k
- regnety_120.sw_in12k
- swin_tiny_patch4_window7_224.ms_in22k
- swin_base_patch4_window7_224.ms_in22k
- convnext_small.in12k
- convnext_tiny.in12k
- convnext_tiny.fb_in22k
- convnext_small.fb_in22k
- convnext_nano.in12k
- convnext_base.fb_in22k
- eca_nfnet_l0
- vit_base_patch16_224.dino
- vit_small_patch16_224.dino
- vit_base_patch16_224.mae
- vit_base_patch16_224.orig_in21k
- vit_base_patch32_224.orig_in21k
- vit_tiny_r_s16_p8_224.augreg_in21k
- vit_small_r26_s32_224.augreg_in21k
- vit_tiny_patch16_224.augreg_in21k
- vit_small_patch32_224.augreg_in21k
- vit_small_patch16_224.augreg_in21k
- vit_base_patch32_224.augreg_in21k
- vit_base_patch16_224_miil.in21k
- vit_base_patch16_224.augreg_in21k
- tf_efficientnetv2_s.in21k
- tf_efficientnetv2_m.in21k
- tf_efficientnetv2_l.in21k
- tf_efficientnetv2_b3.in21k

E HF-Hub Dataset Details

In order to test our method on real-world data, we collected more than 250 classifiers uploaded by users to hugging face where overall there 1300 possible logits in the dataset. The models are trained on a diverse set of models, and class names are given by free text. Hence, class names may not align perfectly as each user spells concept a bit differently (e.g., “Apple” vs. “Apples’). Moreover, some classifiers have different levels of granularity, such as “Car” vs. a specific car model “Toyota”. For evaluation purposes only, we created a label mapping where we manually annotated to which classes each logit can be mapped. We follow these rules to allow mappings between labels: (i) Different spelling map to each other. (ii) An object can be mapped to a specific type of it, e.g. "cat" -> "siamese cat". (iii) A specific type of object can be mapped to its super-class e.g. "siamese cat" -> "cat". (iv) object of the same level of granularity that share a super class cannot be mapped to each other. For example, a "Golden Retriever" is not a good match for a "Husky". Additionally, we created an additional mapping which matches each class to its corresponding ImageNet concept when available.

Saying that we highlight that this label mapping is only needed for numerical evaluations purposes and that our method is completely unsupervised. When trying to search for models in-the-wild models the label mapping is not necessary.

F More Retrieval Samples

We present here 20 more randomly sampled retrievals of ProbeLog. We also provide a zip file with 80 more random retrievals as well. We can see that most of ProbeLog’s retrievals are near misses. E.g., the class “crt screen” was matched to “television”, and the class “space bar” was matched to “computer keyboard”. Moreover, we can see that many mistakes which may seem far away visually are actually semantically related. For instance, when querying for “proboscis monkey” the logit returned was of the class “banana”. Additionally, the query “hourglass” returned “barometer” as “barometer” is visually similar to an analog clock. However, there is still room for improvement as some mistakes are completely wrong such as “tarantula” and “baseball player”.

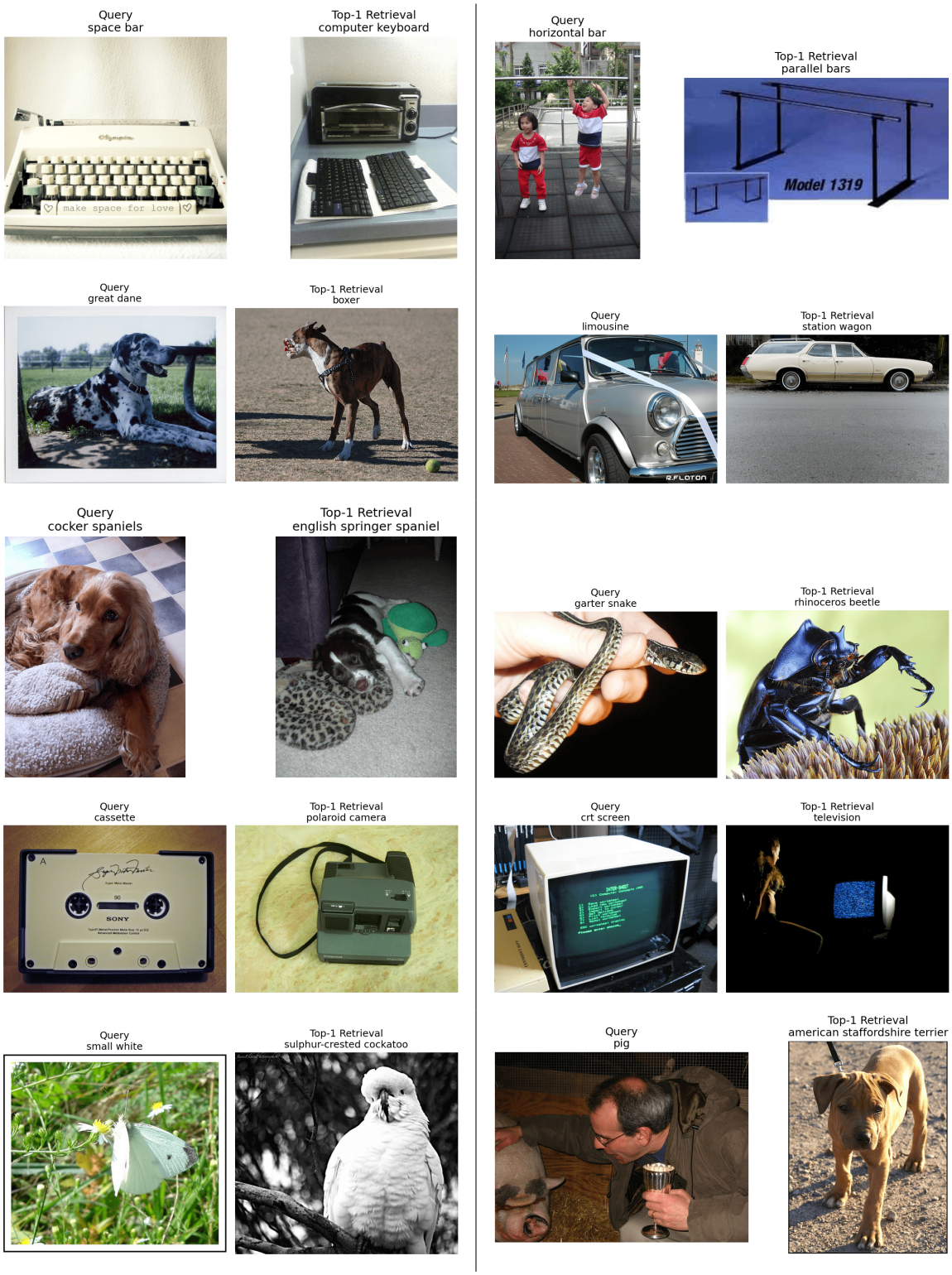

Figure 9: ***Retrieval Failure Cases.*** Visualization of random failure cases of ProbeLogs. Each pair shows a query image (left) and the top-1 retrieved result (right). These examples highlights that most of ProbeLogs mistakes are either semantically or visually similar classes.

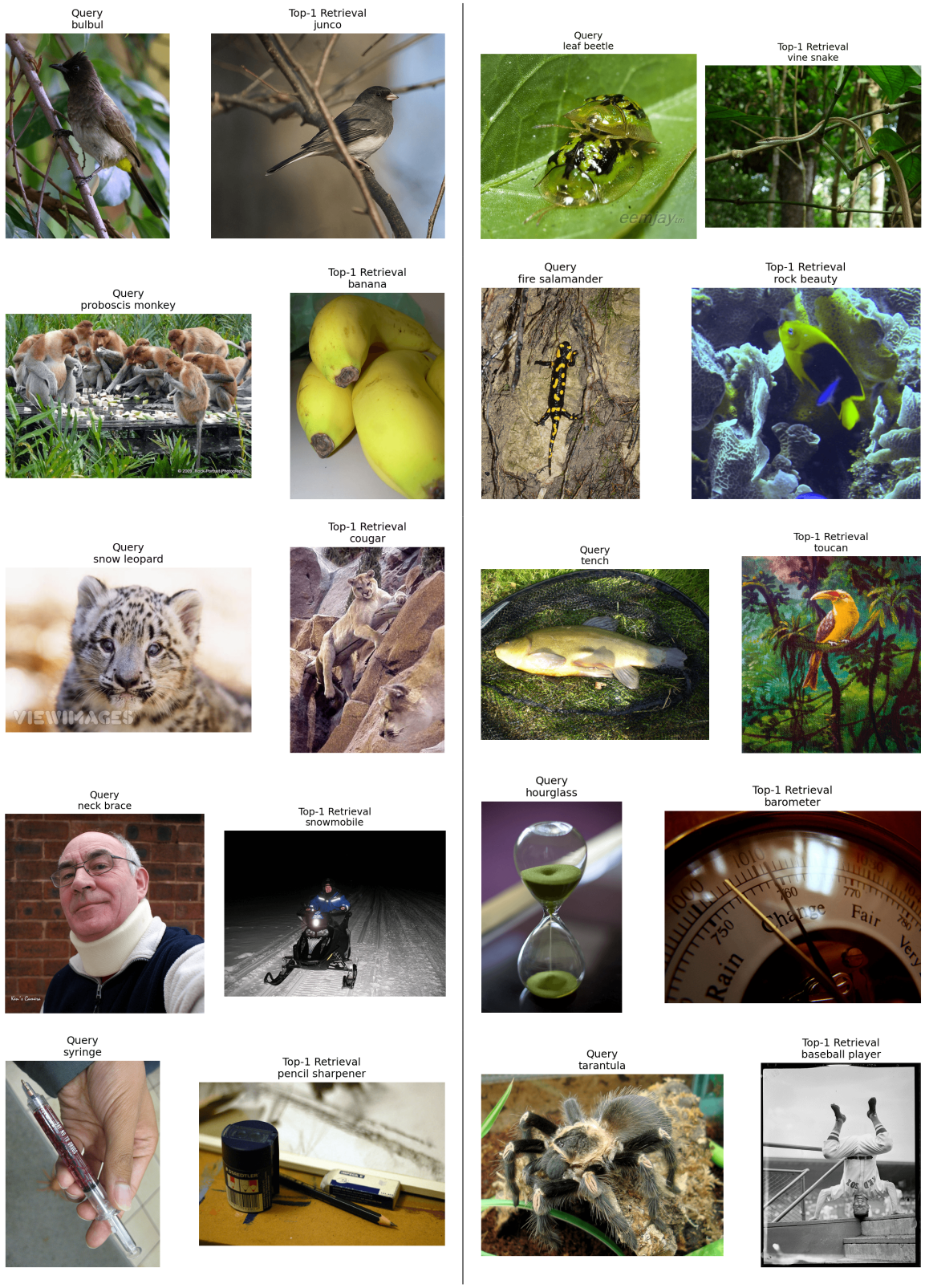

Figure 10: ***Retrieval Failure Cases.*** Visualization of random failure cases of ProbeLogs. Each pair shows a query image (left) and the top-1 retrieved result (right). These examples highlights that most of ProbeLogs mistakes are either semantically or visually similar classes.