Towards flexible perception with visual memory

Robert Geirhos [*1] **Priyank Jaini** [*1] **Austin Stone** [*1] **Sourabh Medapati** [*1]
Xi Yi [1] **George Toderici** [1] **Abhijit Ogale** [1] **Jonathon Shlens** [1]

Abstract

Training a neural network is a monolithic endeavor, akin to carving knowledge into stone: once the process is completed, editing the knowledge in a network is hard, since all information is distributed across the network's weights. We here explore a simple, compelling alternative by marrying the representational power of deep neural networks with the flexibility of a database. Decomposing the task of image classification into image similarity (from a pre-trained embedding) and search (via fast nearest neighbor retrieval from a knowledge database), we build on well-established components to construct a simple and flexible visual memory that has the following key capabilities: (1.) The ability to flexibly add data across scales: from individual samples all the way to entire classes and billion-scale data; (2.) The ability to remove data through unlearning and memory pruning; (3.) An interpretable decision-mechanism on which we can intervene to control its behavior. Taken together, these capabilities comprehensively demonstrate the benefits of an explicit visual memory. We hope that it might contribute to a conversation on how knowledge should be represented in deep vision models—beyond carving it in "stone" weights.

1. Introduction

In the pretty diagrams on ”Intro to Machine Learning” slides, an ideal ML workflow looks like this: Data collection, preprocessing, choosing a model, training, evaluation, deployment. Happy ending—the model is deployed, the users love it, and one can finally go on that well-deserved vacation and catch up on the latest AGI memes.

Until, of course, the enemy of any ideal world sets in: reality. The real world constantly keeps changing, and so do data requirements. New data and datasets become available, and existing ones become deprecated for a variety of reasons, including concerns around fairness, biases or unsafe content. Knowledge changes, and concepts drift (Tsymbal, 2004; Lu et al., 2018): Phones and cars look different today than they did a few years ago, and different from how they will look in the future. When it comes to data, the only constant is change (Cao & Yang, 2015; Bourtoule et al., 2021; Nguyen et al., 2022; Zhang et al., 2023). Consequently, from a modeling perspective, in order to keep up with this change one would ideally want to constantly re-train or fine-tune models, which is not feasible. In short, as anyone who has ever deployed a model has experienced firsthand, one is constantly battling the symptoms of a single underlying cause: the fact that deep learning models have a static knowledge representation entangled in millions or billions of model parameters. We, among many others working on memory e.g. Weston et al. (2014); Chen et al. (2018); Wu et al. (2021); Iscen et al. (2022); Nakata et al. (2022); Iscen et al. (2023); Prabhu et al. (2023); Gui et al. (2024); Shao et al. (2024); Silva et al. (2024), believe that this is not a great way to represent visual knowledge for deep learning. Instead, we argue that we should build models that cleanly separate representation (*how* things are represented, e.g. through feature embeddings) from visual memory (*what* is known). In short, deep learning models need a flexible visual memory: a way to explicitly utilize and edit knowledge.

In this work, we build a simple visual memory for classification and show that it has seven desirable capabilities, including the ability to flexibly add data across scales (from individual samples to classes and even billion-scale data), the ability to remove data from our model's classification process through machine unlearning and memory pruning, and a simple, interpretable decision-mechanism on which we can intervene to control its behavior. Our main goal is to provide a compelling idea of how beneficial a flexible visual memory for deep learning can be from a variety of perspectives and capabilities. From a technical standpoint, we aim for simplicity: retrieving k nearest neighbors (in an embedding feature space) along with their labels to classify a query image. This approach allows us to investigate where a simple visual memory mechanism helps, where its limi-

[*] Equal contribution

[1] Google DeepMind. Correspondence to: Robert Geirhos <lastname@google.com>, Priyank Jaini <pjaini@google.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

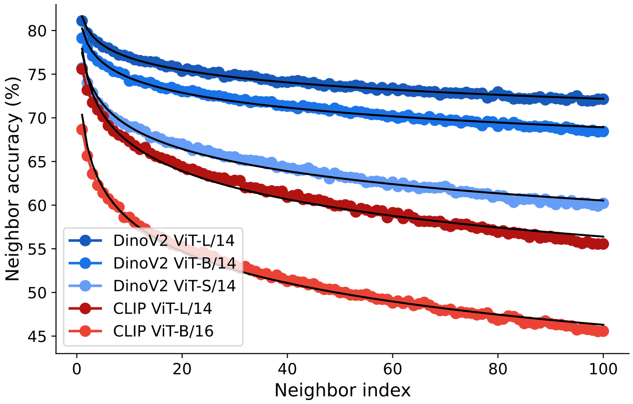


Figure 1: **Reliability of retrieved memory samples.** This plot visualizes the ImageNet top-1 validation accuracy of a single retrieved neighbor depending on the index of the neighbor (index 0: nearest neighbor). The decrease in accuracy with increasing neighbor index follows smooth trajectories and can be approximated by a two-parameter logarithmic fit (black lines). More detailed plot in Appendix H.

tations may be, and where there might be opportunities for improvement through a more complex system. We hope that by demonstrating clear benefits from a simple visual memory, this article might contribute to a conversation on how knowledge ought to be represented in deep vision models. Our contributions are as follows:

1. Our main contribution is to demonstrate that a visual memory based system enables **flexible capabilities, making progress towards some of deep learning's grand challenges:** In specific settings, *unlearning with perfect guarantees* which is currently considered a major open problem in deep learning, an improved understanding of how decisions relate to datapoints through an *interpretable decision-mechanism enabling data attribution*, and controlling sample influence through *memory pruning*. These capabilities highlight the promise of separating knowledge from representation through a visual memory.

2. **A simple visual memory performs well at scale**: On the methodological side, we aim for simplicity. Instead of re-inventing the wheel, we build on established building blocks from the literature like SSL features, kNN classification and fast, scalable similarity search. We contribute technical improvements like RankVoting and VLM re-ranking, achieving 88.5% top-1 ImageNet validation accuracy which improves over both DinoV2 ViT-L14 kNN and linear probing.

We argue that the way current deep learning models represent knowledge (static knowledge representation, hard to update, hard to unlearn, hard to understand how a decision is made) is problematic. As an alternative, we built a working proof-of-concept: By building on the long history of nearest neighbor methods, and "marrying" them with a powerful deep learning representation (such as SSL features from DinoV2) and a billion-scale visual memory.

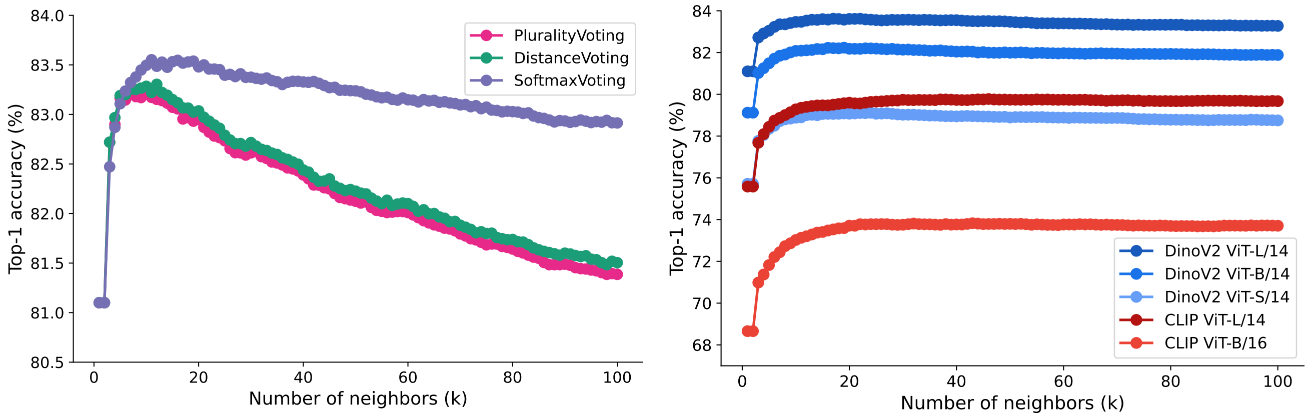
Related work. The concept of a visual memory has a long history in ML, neuroscience and psychology. In psychology, *exemplar theory* posits that humans recognize objects by comparing them to existing examples in visual memory (Medin & Schaffer, 1978; Nosofsky, 1986; Dopkins & Gleason, 1997; Jäkel et al., 2008; Nosofsky, 2011), like the ALCOVE model (Kruschke, 2020). In ML, prior to deep learning, *instance-based learning* (also known as memory-based learning) was a popular alternative to *model-based learning* (Aha et al., 1991; Quinlan, 1993). For instance, Turk & Pentland (1991) used nearest neighbor methods to classify faces, and Sivic & Zisserman (2003) build a visual memory inspired by text retrieval for object retrieval from videos. In recent years, hybrid approaches have started to combine the benefits of both approaches. Deep neural network variants (model-based since they learn generalized abstractions of data) of k-nearest neighbor algorithms (instance-based since they compare new data to existing exemplars in memory) have been proposed with various motivations, including few-shot learning (Wang et al., 2019b; Yang et al., 2020; Bari et al., 2021), improving adversarial robustness (Sitawarin & Wagner, 2019; Papernot & McDaniel, 2018; Rajani et al., 2020), medical image classification (Zhuang et al., 2020), confidence calibration (Papernot & McDaniel, 2018), interpretability (Papernot & McDaniel, 2018; Wallace et al., 2018; Lee et al., 2020; Rajani et al., 2020), image denoising (Plötz & Roth, 2018), retrieval-augmented learning (Khandelwal et al., 2019; Drozdov et al., 2022), anomaly and out-of-distribution detection (Bergman et al., 2020; Sun et al., 2022). Recently, Nakata et al. (2022) tested a kNN-based visual memory up to ImageNet-scale (1.28M images), and Khandelwal et al. (2019); Wu et al. (2021) applied kNN-based approaches to neural language models. In contrast, we scale visual memory to the billion scale, improve ranking, and show systematic benefits across different tasks.

2. Building a retrieval-based visual memory for classification

Given a dataset $\mathcal{D}_{\text{test}} := \{(\tilde{\boldsymbol{x}}_1, y_1), \cdots, \tilde{\boldsymbol{x}}_n, y_n\}$, we want to classify each image $\tilde{\boldsymbol{x}}_i \in \mathcal{D}_{\text{test}}$. Our classification approach consists of two steps: (i) building a visual memory, and (ii) using it for fast nearest neighbor based inference.

2.1. Building a visual memory

Our visual memory retrieves (image, label) pairs from an image dataset when a query is made by directly retrieving those images that are considered similar to a test image



(a) Existing aggregation methods, DinoV2 ViT-L14

(b) RankVoting across models

Figure 2: **Aggregating information across retrieved memory samples. (left)** Existing aggregation methods are overconfident in distant neighbors, resulting in the paradox of decaying ImageNet-1K accuracy with more information. The same pattern is also seen for other models and datasets in the Appendix (Figures 7 and 8). **(right)** This is not the case for RankVoting, a power-function based method which reaches higher and stable performance across models and choices of k.

according to a model. The model is a fixed pre-trained image encoder, meaning that no training takes place when adding information to visual memory. No copies of the dataset are stored in the visual memory. Instead, feature maps are extracted from the model based on a set of images related to the downstream classification task at hand, such as a standard training set. For our experiments, our visual memory comprises of features extracted from a dataset like the ImageNet-1K (Russakovsky et al., 2015) training set or JFT (Zhai et al., 2022) using different encoders like DinoV2 (Oquab et al., 2023) and CLIP (Radford et al., 2021). Thus, given a pretrained image encoder, Φ, and a dataset of (image, label) pairs $\mathcal{D}_{\mathsf{train}} := (\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_N, y_N)$, we obtain features $\boldsymbol{z}_i := \Phi(\boldsymbol{x}_i), \forall \boldsymbol{x}_i \in \mathcal{D}_{\mathsf{train}}$. Subsequently, the feature maps and corresponding label pairs are put in a database thereby creating $\mathsf{VisualMemory} := \{(\boldsymbol{z}_1, y_1), (\boldsymbol{z}_2, y_2), \cdots, (\boldsymbol{z}_N, y_N)\}$ for classification. For both DinoV2 and CLIP, we use the last image embedding layer as a feature space.

2.2. Retrieval-based classification using visual memory

Given a query image $\tilde{\boldsymbol{x}} \in \mathcal{D}_{\mathsf{test}}$, we extract its feature map, $\tilde{\boldsymbol{z}} = \Phi(\tilde{\boldsymbol{x}})$. We then query $\mathsf{VisualMemory}$ to extract k feature vectors, $\mathsf{Neighbors}(\tilde{\boldsymbol{x}}) := \{(\boldsymbol{z}_{[1]}, y_{[1]}), (\boldsymbol{z}_{[2]}, y_{[2]}((, \cdots, (\boldsymbol{z}_{[k]}, y_{[k]})\}$, that are closest to the query features $\tilde{\boldsymbol{z}}$ using the cosine distance, which is the default retrieval similarity measure for SSL models like DinoV2. $\mathsf{Neighbors}(\tilde{\boldsymbol{x}})$, are ordered by distance i.e.

$$\mathsf{dist}(\tilde{z}, \boldsymbol{z}_{[i]}) \leq \mathsf{dist}(\tilde{z}, \boldsymbol{z}_{[j]}), \ \forall i \leq j.$$

We then assign a weight, w_i, to each neighbour $(\boldsymbol{z}_{[i]}, y_{[i]})$ and aggregate the scores for each neighbour with the same label. Finally, we assign that label to the query image with the highest aggregate score. We implemented retrieval based classification using one of the following two approaches:

1. Fast inference using matrix multiplication on GPUs/TPUs: For smaller datasets like ImageNet, we saved $\mathsf{VisualMemory}$ as a matrix of size $\mathsf{num\_images} \times \mathsf{num\_dims}$. During inference, for an encoded query image of size $1 \times \mathsf{num\_dims}$, we computed the dot product of this encoded image with every entry in $\mathsf{VisualMemory}$ getting a matrix of size $\mathsf{num\_images} \times 1$. We then computed the k nearest neighbors using the $\arg\max$ operation.

2. Fast and scalable nearest neighbor search: We used ScaNN (Guo et al., 2020) for accelerating nearest neighbor search at scale. Specifically, we saved the $\mathsf{VisualMemory}$ as a database and used ScaNN for fast lookup of nearest neighbors during inference. Storing features requires only about 1–3% of the space of storing the dataset itself and search latency is 500–600 QPS at perfect recall for 1M features, thus approximately 2 milliseconds per query due to parallelization which can be done on CPUs. See Appendix K for details on latency and storage. This method scales easily to billion-scale memory (cf. Section 3.3).

We mentioned earlier that we retrieve a set of neighbors, $\mathsf{Neighbors}(\tilde{\boldsymbol{x}})$ and aggregate information across them to make a classification decision. In order to understand how reliable (i.e., accurate) retrieved memory samples are from the first to the 100th neighbor, we systematically analyze neighbor reliability in Figure 1. As expected, reliability decreases as the neighbor index k increases, but even at large k the neighbors contain above-chance information about the ground truth class. This suggests that aggregating information across different neighbors may be beneficial to decision-making, leading to the question: *What is the best* *aggregation strategy?* We empirically study this by testing different weighting strategies for aggregation:

Plurality voting: Each neighbour in $\mathsf{Neighbors}(\tilde{\boldsymbol{x}})$ is assigned an equal weight of 1.0. This is the classic, most simple voting method and used e.g. by Nakata et al. (2022).

Distance voting: Each neighbour in $\mathsf{Neighbors}(\tilde{\boldsymbol{x}})$ is assigned a weight based on its Cosine distance to the query image $\tilde{\boldsymbol{x}}$ i.e. $w_i = \exp\big(-\mathsf{dist}(\tilde{\boldsymbol{z}}, \boldsymbol{z}_{[i]})\big)$. This approach has been used by Khandelwal et al. (2019) for nearest neighbor language models.

Softmax voting: Each neighbour is assigned a weight based on the softmax function i.e. $w_i = \mathsf{softmax}\big(\mathsf{dist}(\tilde{\boldsymbol{z}}, \boldsymbol{z}_{[i]}), \tau\big)$ where τ is the temperature. This voting method is considered state-of-the-art; for example nearest neighbor accuracies of self-supervised models are reported using this method. A temperature of $\tau = 0.07$ frequently appears in literature (Wu et al., 2018; Caron et al., 2021; Oquab et al., 2023) and is reported as a parameter "which we do not tune" in the Dino paper (Caron et al., 2021, p. 18). We observe that performance is sensitive to this parameter; other temperatures perform worse. We therefore follow the literature in using $\tau = 0.07$.

Rank voting: We propose using a simple aggregation approach wherein each neighbour is assigned a power-function weight based on its rank in the ordered set $\mathsf{Neighbors}(\tilde{\boldsymbol{x}})$ i.e. $w_i = 1/(\alpha + \mathsf{rank}_i)$ where rank_i is i and α is an offset to avoid division by zero that is set to 2.0. This is similar, though not identical to, Gou et al. (2011) who used power-law weighting in a different context.

In Figure 2a, we compare the top-1 ImageNet validation accuracy of different ranking methods as a function of number of neighbours, with the ImageNet-1K training set as the visual memory using the DinoV2/ViT-L14 model as the featurizer. Paradoxically, existing aggregation methods like plurality voting, distance-based voting, and softmax voting show *decaying* performance as the provided information (number of nearest neighbors) *increases*. This suggests that the methods are overconfident in distant neighbors, assigning them too much weight. Our simple, parameter-free rank based voting method, however, leads to an increase in performance with more neighbors until a certain k after which the performance plateaus, which is the ideal scenario (Figure 2b). Furthermore, rank-based voting also outperforms baselines in absolute terms; quantitative comparisons can be found in the Appendix (Tables 4 to 8) where we also study the influence of hyperparameters (Figure 9). This indicates that a simple, power-function based method can reliably integrate information across retrieved memory samples.

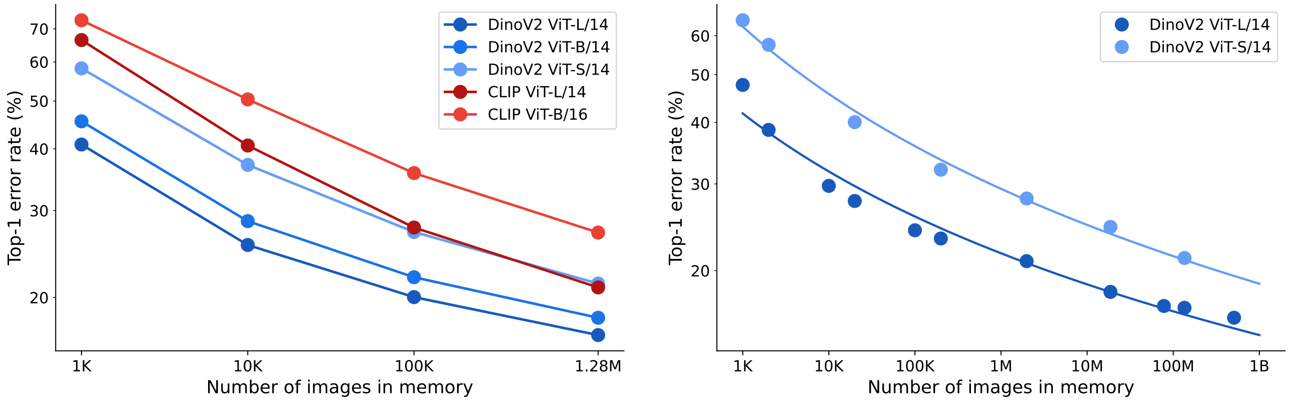
Gemini re-ranking. Our results above demonstrate that different aggregation strategies have a large impact on downstream performance. How far can we push the upper limit on aggregating information from different neighbors? We perform a controlled experiment using the Gemini 1.5 Flash model (Reid et al., 2024) to test this: We add the 50 nearest neighbors from DinoV2 ViT-L14 for a query image along with their labels into Gemini's context. We then query Gemini to predict the query image's label. This achieves 88.5% ImageNet validation accuracy, a substantial improvement over both DinoV2 ViT-L14 kNN (83.5%) and linear probing (86.3%) performance. Interestingly, Gemini's performance is mainly driven by the neighbor information through in-context learning since it only achieves 69.6% accuracy without neighbors (when just the query image is provided to the model). The performance improvement highlights the potential of using vision-language models as a visual memory re-ranker. Given that our main goal is to explore a simple visual memory system, we mostly focus on non-Gemini ranking methods throughout our analysis.

3. Capabilities of a visual memory

Our primary goal is to motivate the concept of a machine *visual memory* from a variety of different perspectives. To this end, we investigate how such a memory can benefit the following capabilities: 3.1 Flexible lifelong learning: adding novel OOD classes; 3.2 Flexibly trading off compute and memory; 3.3 Flexibly adding billion-scale data without training; 3.4 Flexible removal of data: machine unlearning; 3.5 Flexibly increasing dataset granularity; 3.6 Flexible data selection: memory pruning; 3.7 Interpretable & attributable decision-making.

3.1. Flexible lifelong learning: adding novel OOD classes (data and labels)

Standard classifiers, whether trained end-to-end (supervised models) or with a linear classifier (self-supervised models), are not able to handle new information without re-training. For instance, adding new classes or changing labels in an existing model usually involves either re-training or fine-tuning parts of the model. A retrieval-based visual memory, in contrast, is able to process such information in a natural and flexible way, aligning with the requirements of lifelong learning (Parisi et al., 2019). We tested this by adding data for 64 new classes, along with their new labels, to the visual memory of a pre-trained DinoV2 ViT-L14 model (in addition to the ImageNet train set, which is in-distribution for the model). We took the new classes from the NINCO dataset (Bitterwolf et al., 2023), a dedicated OOD dataset that is designed to have no overlap with existing ImageNet labels and samples. This requires the model to transfer what it has learned to new, unseen concepts. The new task is therefore harder, as the model has to retrieve images from both in-distribution and OOD classes. The resulting visual memory has 1064 classes (1K from ImageNet and 64 from NINCO).



(a) ImageNet (million-scale)

(b) JFT (billion-scale)

Figure 3: **Memory scaling: flexibly trading off compute and memory.** ImageNet top-1 validation error decreases systematically as the memory size is increased (i.e., recognition accuracy increases with scale). **(left)** Million-scale memory consisting of ImageNet-train labels. **(right)** Billion-scale memory bank consisting of machine-generated pseudo labels on the JFT dataset (Zhai et al., 2022). Accuracy continues to decrease even with billion-scale data in memory. The roughly constant offset between models of different sizes suggests the possibility of a flexible trade-off: The same error rate can be achieved with a small model and large memory, or a large model and a small memory.

Table 1 shows that with a visual memory it is possible to add new classes such that the in-distribution accuracy is maintained without catastrophic forgetting (the new classes only change ImageNet validation performance by 0.02–0.04% depending on the aggregation method), while at the same time reaching very high accuracy on the new OOD classes (approx. 87% top-1) without any training. Figure 12 in the appendix confirms that the samples are indeed OOD for the model, as demonstrated by larger distances to nearest neighbors. This highlights that a visual memory is capable of flexibly adding new information—an important capability since the world is not static. Furthermore, the memory is highly robust towards label corruption for up to 60% random labels, as shown in Appendix D.

Table 1: **Flexible lifelong learning: adding OOD classes.** A visual memory of DinoV2 ViT-L14 with ImageNet-train (IN-train) as memory database is able to handle a simple "insert into memory" operation for 64 out-of-distribution classes (data and labels) from the NINCO dataset (Bitterwolf et al., 2023), leading to high performance on new classes without harming existing ones.

memory →	IN-train	IN-train-and-NINCO	
query →	IN-val	IN-val	NINCO
no aggregation	81.1	81.1	86.4
PluralityVoting	83.2	83.2	86.9
DistanceVoting	83.3	83.3	87.1
SoftmaxVoting	83.6	83.5	87.5
RankVoting	83.6	83.6	87.4

3.2. Flexibly trading off compute and memory

Next, we turn our attention to studying the scaling behaviour of visual memory with increasing memory model size. We specifically test whether smaller models larger memory can match the performance of larger models with smaller memory. This is because, all else being equal, a bigger model might require fewer examples in memory to represent different concepts. We empirically study the scaling behaviour of visual memory based retrieval systems in Figure 3a using models of different sizes like DinoV2 ViT models of sizes S/14 (21M params), B/14 (86M params), and L/14 (300M params), as well as CLIP ViT models of sizes B/16 and L/14. We plot the top-1 error rate as a function of number of images in visual memory. The plot demonstrates that for each model, the error rate consistently decreases as we increase the visual memory size. Notably, already with a single exemplar per class in memory, ImageNet validation performance is well above chance (41% top-1 error for DinoV2 ViT-L14). It also suggests the possibility of a flexible trade-off between model size and memory size: e.g. for the different DinoV2 models, the S/14, B/14, and L/14 variant achieve similar performance at 1.28M, ∼150K, and ∼70K memory capacity respectively. This indicates that a smaller model with large memory can match the performance of a larger model with smaller memory.

3.3. Flexibly adding billion-scale data without training

Billion-scale dataset with pseudo labels. As demonstrated in Section 3.2, performance systematically improves with increased memory size across both small and large models.

We here test how far this trend holds beyond relatively small-scale, well-curated settings like ImageNet-1K by scaling visual memory to the billion-scale unlabeled data regime. Billion-scale search is known to be fast (Johnson et al., 2019; Guo et al., 2020; Chen et al., 2021; Khan et al., 2024), but it is an open question how well a billion-scale memory with a modern featurizer like DinoV2 performs for classification. In order to test predictive performance at scale, we obtain a billion-scale dataset from the union of the ImageNet-1K train set and a subset of the JFT-3B dataset (Zhai et al., 2022). To this end, we treat JFT as an unlabeled dataset by ignoring its original labels and instead obtaining *pseudo labels* via ViT-22B-224px (Dehghani et al., 2023), a highly performant classifier. We excluded images whose labels do not have a correspondence with ImageNet labels.

Scaling. In Figure 3b, we show the downstream ImageNet validation performance of two DinoV2-ViTs as a function of memory size. The plot demonstrates that even in the billion-scale data regime, validation error decreases when increasing memory size without any training. In log-log space, a logarithmic function fits the empirical scaling trend well. In the literature, simple scaling trends such as the one we observe are powerful predictors of scaling behaviour for different model and dataset sizes (Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022). This experiment confirms that a memory-based system is performant across seven orders of magnitude, improving from both better features and more data.

Out-of-distribution performance. In order to understand whether the benefits of increased memory size transfer to out-of-distribution (OOD) data, we compare DinoV2 ViT-L14 once with ImageNet-train in memory and once with JFT pseudo-labels in memory. The models are evaluated on the ImageNet-A (Hendrycks et al., 2021), ImageNet-R (Hendrycks et al., 2020), ImageNet-Sketch (Wang et al., 2019a), ImageNet-V2 (Shankar et al., 2020), and ImageNet-ReaL (Beyer et al., 2020) datasets. As an additional well-performing yet "inflexible" baseline, we report linear probing accuracies from the DinoV2 paper (Oquab et al., 2023). Table 2 shows that visual memory scaled with JFT data improves OOD performance across all datasets compared to an ImageNet-based visual memory. Gemini re-ranking again leads to performance gains. Overall, the finding that memory scale transfers to OOD improvements is important in the context of continual learning, where a flexible visual memory can easily incorporate newly available data that the model was not trained on, improving performance both in- and out-of-distribution.

3.4. Flexible removal of data: machine unlearning

The world is not static. Thus, in addition to the need to flexibly add novel data, it is often desirable to remove the influence of specific training data from a model's decision-making process after it has been trained (Cao & Yang, 2015; Bourtoule et al., 2021; Nguyen et al., 2022; Zhang et al., 2023). A range of intricate methods are being developed to remove or reduce the influence of certain training samples (Gupta et al., 2021; Sekhari et al., 2021; Ullah et al., 2021; Kurmanji et al., 2024; Sepahvand et al., 2024)—a challenging endeavour if knowledge is embedded in millions or billions of model weights. In contrast, for models with an explicit visual memory and in the context of classification, machine unlearning becomes as simple as removing the dataset sample from the visual memory. For instance, after adding the NINCO dataset (Bitterwolf et al., 2023) into visual memory, we can remove any NINCO sample with outstanding performance on all three key unlearning metrics reported by Liu (2024): *Efficiency*: How fast is the algorithm compared to re-training? (Very fast: less than 20 milliseconds for deleting a feature/sample from disk.) *Model utility*: Do we harm performance on the retain data or orthogonal tasks? (By design not at all.) *Forgetting quality*: How much and how well are the 'forget data' actually unlearned? (Completely and entirely by design.) In comparison, the winner of the NeurIPS 2023 machine unlearning challenge (Triantafillou et al., 2024) still suffers from an accuracy gap of 4%, low forgetting quality, and being relatively inefficient requiring 8 epochs of training.

Can machine unlearning therefore be solved with a visual memory? If the embedding model is trained on data that needs to be unlearned, machine unlearning remains challenging. If, however, the embedding model is trained on a safe, generalist dataset (e.g., a publicly available image dataset) and data that may need to be considered for unlearning later is put into the visual memory, then machine unlearning for classification indeed becomes as simple as deleting a datapoint from the visual memory. This can be particularly helpful for tasks that may require private or confidential data: a model can be trained on publicly available datasets to learn general and information features and the private data can be added to a visual memory on local devices for downstream tasks to preserve privacy.

3.5. Flexibly increasing dataset granularity on iNaturalist

In contrast to static classification, where a model is trained once without updates, a visual memory model should be able to flexibly refine its visual understanding as more information becomes available. We test this using DinoV2 ViT-L14 embeddings on the iNaturalist21 dataset (iNaturalistTeam, 2021), a large-scale imbalanced dataset of animal and plant images containing 10,000 species spanning seven taxonomic levels, from coarse (kingdom) to fine-grained (species). In a leave-one-out fashion, we simulate the discovery of a new species by putting 50 exemplars for each of

Table 2: **OOD evaluation.** Out-of-distribution performance improves with larger visual memory size. Across all datasets, a visual memory with JFT memory outperforms ImageNet memory demonstrating advantages of scaling visual memory for OOD performance. Probe details: Appendix J.

Model	Method	IN-A	IN-R	IN-Sketch	IN-V2	IN-ReaL
DinoV2 ViT-L14	linear probe	71.3	74.4	59.3	78.0	89.5
DinoV2 ViT-L14	ImageNet memory	58.8	62.8	61.5	75.6	87.1
	+ Gemini re-ranking	68.4	72.3	72.5	81.7	89.9
DinoV2 ViT-L14	JFT memory	61.1	73.7	68.0	77.6	88.2
	+ Gemini re-ranking	69.6	81.4	75.0	82.3	90.5

the 9,999 species into memory and then iteratively adding more data for the remaining "newly discovered" species—starting from zero exemplars all the way to 50 exemplars. Classification is performed using $k = 1$ nearest neighbors (no aggregation).

In Figure 4 we observe the following: (1.) Already before a single example of the new species is added, it can already be placed in the right part of the taxonomic tree well beyond chance (35.2% accuracy at the genus level compared to $\sim$0% chance). (2.) Accuracy at the species level improves substantially by adding just a handful of images of the target species (e.g., 5–10 images); a regime where training a classifier would typically fail due to data scarcity. (3.) Interestingly, adding more samples of the discovered species not only improves species-level accuracy, but also leads to a "rising tide lift" of improvements across all levels of the taxonomic hierarchy. This indicates that a visual memory is well-suited for hierarchical classification tasks and settings where data for new concepts is initially scarce but becomes more abundant over time—which is often the case in applications like fraud detection, personalized recommender systems, and scientific discovery.

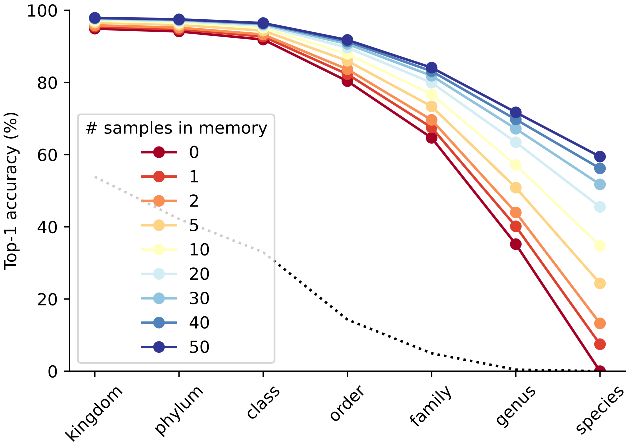


Figure 4: **Impact of memory bank size on top-1, k=1 accuracy across taxonomic levels on iNaturalist.** Top-1 accuracy for a target species across different taxonomic levels as the number of exemplars in the memory bank for that species increases from 0 to 50. Each line represents the average accuracy over all 10,000 species in the iNaturalist 2021 dataset, while the number of examples in visual memory is fixed at 50 exemplars for all other species. The black dotted line indicates baseline accuracy from predicting the majority class.

3.6. Flexible data selection: memory pruning

The ability to flexibly remove the influence of certain datapoints is not just desirable in the unlearning sense, but also advantageous in the context of dataset pruning, an emerging field that analyzes the quality of individual data points. The goal of dataset pruning is to retain only *useful* samples, while removing those that have a neutral or harmful effect on model quality. The key challenge is that in standard black-box models, it is entirely unclear whether any given sample is helpful or harmful. The gold standard is leave-one-out-training (for ImageNet, this would consist of training 1.28 million models); current methods seek to approximate this extremely costly approach with various heuristics (Feldman & Zhang, 2020; Chitta et al., 2021; Paul et al., 2021; Sorscher et al., 2022; Abbas et al., 2023a). By contrast, the contribution of a data sample to decision-making in a visual memory based system is straightforward. For any given query image $\tilde{\boldsymbol{x}}$, the neighbor set $\mathsf{Neighbors}(\tilde{\boldsymbol{x}})$ fully describes which samples contributed to the decision. Furthermore, this information also highlights whether the samples were helpful (correct label) or harmful (wrong label) for the decision. We, therefore, transfer the concept of dataset pruning to memory, and propose visual *memory pruning*. To this end, we estimate sample quality by querying the ImageNet training set against a visual memory consisting of the exact same dataset (IN-train, discarding the first neighbor which is identical to the query). This approach requires no more compute than a single forward pass over the training set. We then record the number of times any given neighbor contributed to a wrong decision, resulting

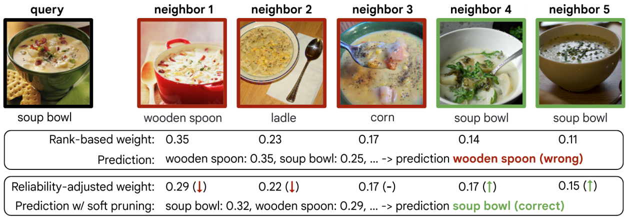

query
neighbor 1
neighbor 2
neighbor 3
neighbor 4
neighbor 5
soup bowl
wooden spoon
ladle
corn
soup bowl
soup bowl
Rank-based weight: 0.35 0.23 0.17 0.14 0.11
Prediction: wooden spoon: 0.35, soup bowl: 0.25, ... -> prediction wooden spoon (wrong)
Reliability-adjusted weight: 0.29 (↓) 0.22 (↓) 0.17 (-) 0.17 (↑) 0.15 (↑)
Prediction w/ soft pruning: soup bowl: 0.32, wooden spoon: 0.29, ... -> prediction soup bowl (correct)

Figure 5: **Visualization of memory-based decision-making with and without memory pruning.** Given a query image, nearest neighbors are retrieved from memory via Cosine similarity in the embedding space of a model (here: five closest neighbors from the ImageNet train set, embedded via DinoV2 ViT-L14). The model's prediction is based on the weighted aggregation of the neighbor class labels. The rank-based weight decreases with the rank of the neighbor. For soft memory pruning, those weights are adjusted by the reliability of their neighbors. In the specific example here, all five neighbors appear sensible, but they have four different labels. Since the first two neighbors contributed to wrong decisions on the training set, they are downweighted via soft memory pruning, and the prediction changes to the correct class.

in a sample quality estimate. This enables us to exclude low-quality neighbors from the decision-making process by either removing them from the visual memory entirely ("hard memory pruning") or by reducing their weight compared to higher-quality neighbors ("soft memory pruning"). Method details can be found in Appendix I. In Table 3, we show that both memory pruning variants improve ImageNet validation accuracy, with soft pruning leading to larger gains than hard pruning. Figure 5 visualizes the decision-making process for a randomly selected sample where estimating sample reliability improves decision quality. Given that observing the outcome of an intervention is many orders of magnitude faster than traditional leave-out-training, we are optimistic that the visual memory pruning gains we observed with two simple strategies can be improved further in the future.

3.7. Interpretable & attributable decision-making

Unlike a black-box deep learning model, a visual memory offers a natural way to understand a model's specific predictions by attributing them to training data samples (e.g. Papernot & McDaniel, 2018). In Figure 6, we visualize misclassified validation set examples from the ImageNet-A dataset (Hendrycks et al., 2021) using a memory of the ImageNet-1K training set. These randomly selected samples illustrate that many seemingly strange errors (e.g., predicting a type of fence instead of a teddy bear, or a unicycle instead of a bow tie) are in fact sensible given the data, raising questions about label quality of ImageNet-A—in a similar vein as label issues identified for ImageNet (Beyer et al., 2020; Shankar et al., 2020; Yun et al., 2021)—rather than about model quality. This issue is quantified through a human experiment in Appendix N, showing that 2 out of 5 model "errors" on this dataset are instead label errors.

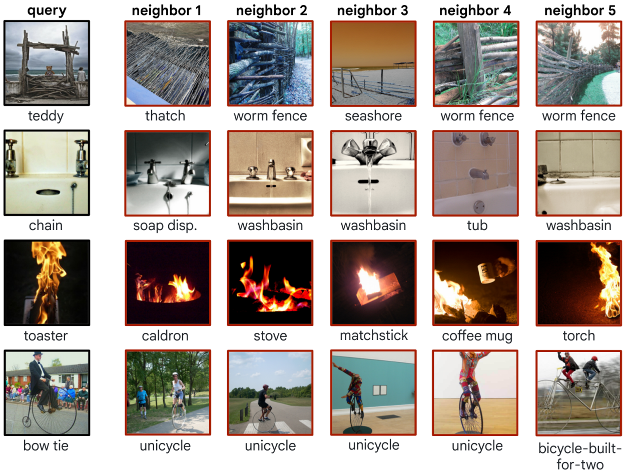


Figure 6: **Interpretable decision-making.** A visual memory enables a clear visual understanding of why a model makes a certain prediction. Here, we show four randomly selected misclassified query images from ImageNet-A (Hendrycks et al., 2021) with five nearest neighbors from DinoV2 ViT-L14 using the ImageNet-1K training set as memory. Labels are from the respective datasets (query: ImageNet-A; neighbors: ImageNet-train). Visually, all neighbors appear reasonable, but not all labels do.

In a separate human experiment (described in Appendix L), we tested whether access to five nearest neighbors helps humans predict model decisions. In this experiment, for each image we provided four label choices including the ground truth label and the model predicted label (if different). The remaining labels were plausible alternatives based on top CLIP predictions for the ImageNet-A test image. For a black-box classifier (no access to neighbors), human accuracy was 56%; this accuracy improved to 83% when given access to four nearest neighbours from our visual memory with a DinoV2 ViT-L14 featurizer, providing strong and falsifiable evidence for the improved interpretability of a visual memory system.

4. Discussion

Summary. Typical neural networks are trained end-to-end: perfect for static settings, yet cumbersome to update whenever knowledge changes. This is limiting their real-world potential since our world is constantly evolving. Incorporating a visual memory, in contrast, enables a range of flexible capabilities that embrace change: lifelong learning through incorporating novel knowledge, being able to forget,

Table 3: **Flexible data selection: memory pruning.** ImageNet validation accuracy improves when removing low-quality samples (hard pruning) or downweighting them (soft pruning). In contrast to standard black-box models, memory models (here: using DinoV2 ViT-L14) offer a strikingly simple way to estimate sample quality since their decisions are based on a few retrieved memory samples.

Pruning	PluralityVoting	DistanceVoting	SoftmaxVoting	RankVoting
no pruning (standard)	83.2	83.3	83.6	83.6
hard pruning (ours)	83.3	83.4	83.6	83.7
soft pruning (ours)	**83.6** (+0.4%)	**83.6** (+0.3%)	**83.9** (+0.3%)	**84.1** (+0.5%)

remove and unlearn obsolete knowledge, flexible data selection through memory pruning, and an interpretable decision-making paradigm on which one can intervene to control its behavior. We systematically explored a simple visual memory that decomposes the task of image classification into two primitives, image *similarity* (from a pre-trained embedding representation) and *search* (via fast, scalable nearest neighbor search from a vector database). Our results demonstrate that technical improvements like RankVoting improve kNN accuracies for both DinoV2 and CLIP over the widely used SoftmaxVoting method that is sensitive to two hyper-parameters (temperature τ and neighbors k). Our approach also narrows the accuracy gap between a nearest neighbor memory (best flexibility, perfect unlearning, improved interpretability) and a fixed linear probe (highest accuracy on static image classification). Importantly, we show that visual memory enables *flexible* perceptual capabilities.

Limitations and future work. First, we only considered the task of image classification across a broad range of datasets. It will be interesting to extend the approach to other visual tasks, such as object detection, image segmentation, instance recognition and to image generation where a visual memory would be desirable, too (since it is prohibitively expensive to re-train large generative models every time data needs to be removed or added). Secondly, our approach relies on a fixed, pre-trained model; strong distribution shifts may require updating the embedding. Self-supervised models are a particularly flexible choice, but it is an open question whether one could train smaller models that excel at their task with the help of a larger memory database. Conceptually, if a model needs to save less information in its weights, it might be possible to reduce the computational footprint of such a model. Additionally, for scalable vector search, adding/removing samples can require adapting the search index, though the amortized cost is low. Furthermore, we sometimes observe a trade-off between flexibility and accuracy. The use of memory pruning weights as a data selection criterion in the context of dataset pruning (Sorscher et al., 2022) might be an interesting avenue for future work.

Outlook. Deep learning is increasingly becoming a victim of its own success: the more widely it is deployed, the stronger its limitations are felt. While the static nature of end-to-end trained networks can easily be forgotten when focusing on fixed academic benchmarks, the real world is anything but static. Data is constantly evolving, leading to the dreaded "model drift" where once-optimal models gradually become less effective (Bayram et al., 2022). Incorporating an explicit visual memory appears to be a promising way forward for real-world tasks where flexibility is key. While the specific approach we employ here might well be improved through more complex systems, we hope that the flexible capabilities we demonstrated might inspire and contribute to a conversation on how knowledge ought to be represented in vision models.

Impact Statement

The use of a flexible visual memory as proposed in this article has several positive societal implications: (1.) through improved machine unlearning with strong guarantees (by design), it becomes possible for certain classifiers to easily remove data that is no longer considered safe (e.g., complying with user requests or in order to fix data safety issues that are discovered after a model is deployed); (2.) increased transparency and trust through a decision-making mechanism that can be inspected visually and intervened on as opposed to an opaque black-box machine learning system; (3.) potentially increased privacy by storing sensitive data in memory (e.g. on-device) as opposed to training models on the data; (4.) resource efficiency: since fast similarity search can be done on CPUs with a low energy footprint, this suggests possibilities for reducing a system's carbon footprint (e.g., combining a small model with a larger database).

Given that the system we explore is used for classification, standard classifier-related risks apply, such as for instance the potential for a system to discriminate, amplify biases, or displace human jobs. In addition, the general dual-use problematic of ML models applies: the same system can often be used for beneficial as well as harmful purposes.

We expect that the societal benefits outweigh the harms since the potential harms are shared with all ML classifiers, while the benefits are specific to our approach.

Code availability. Code to replicate experiments from this paper is available at https://github.com/google-deepmind/visual-memory.

Acknowledgements

We would like to thank Kevin Swersky, David Fleet, Been Kim, Hagen Soltau, Majid Hadian, Kaifeng Chen, Mojtaba Seyedhosseini, Ameya Prabhu and Sung Jin Hwang for feedback and discussions, and Mike Mozer for helpful comments on our article draft. We also thank our anonymous reviewers who provided valuable feedback that improved the paper.

References

Amro Kamal Mohamed Abbas, Evgenia Rusak, Kushal Tirumala, Wieland Brendel, Kamalika Chaudhuri, and Ari S Morcos. Effective pruning of web-scale datasets based on complexity of concept clusters. In *The Twelfth International Conference on Learning Representations*, 2023a.

Amro Kamal Mohamed Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023b.

David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6:37–66, 1991.

M Saiful Bari, Batool Haider, and Saab Mansour. Nearest neighbour few-shot learning for cross-lingual classification. *arXiv preprint arXiv:2109.02221*, 2021.

Firas Bayram, Bestoun S Ahmed, and Andreas Kassler. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245:108632, 2022.

Liron Bergman, Niv Cohen, and Yedid Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint arXiv:2002.10445*, 2020.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? Fixing ImageNet out-of-distribution detection evaluation. In *International Conference on Machine Learning*, pp. 2471–2506. PMLR, 2023.

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

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021.

Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. *arXiv preprint arXiv:1904.00760*, 2019.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

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.

Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Semi-supervised deep learning with memory. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 268–283, 2018.

Kashyap Chitta, José M Álvarez, Elmar Haussmann, and Clément Farabet. Training data subset search with ensemble active learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14741–14752, 2021.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, 2023.

Stephen Dopkins and Theresa Gleason. Comparing exemplar and prototype models of categorization. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 51(3):212, 1997.

Andrew Drozdov, Shufan Wang, Razieh Rahimi, Andrew McCallum, Hamed Zamani, and Mohit Iyyer. You can't pick your neighbors, or can you? when and how to rely on retrieval in the k nn-lm. *arXiv preprint arXiv:2210.15859*, 2022.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

Jianping Gou, Taisong Xiong, Yin Kuang, et al. A novel weighted voting for k-nearest neighbor rule. *Journal of Computers*, 6(5):833–840, 2011.

Zhongrui Gui, Shuyang Sun, Runjia Li, Jianhao Yuan, Zhaochong An, Karsten Roth, Ameya Prabhu, and Philip Torr. kNN-CLIP: Retrieval enables training-free segmentation on continually expanding large vocabularies. *arXiv preprint arXiv:2404.09447*, 2024.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pp. 3887–3896. PMLR, 2020.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34:16319–16330, 2021.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pp. 15262–15271, 2021.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

iNaturalistTeam. iNaturalist 2021 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2021, 2021.

Ahmet Iscen, Thomas Bird, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. A memory transformer network for incremental learning. *arXiv preprint arXiv:2210.04485*, 2022.

Ahmet Iscen, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. Retrieval-enhanced contrastive vision-text models. *arXiv preprint arXiv:2306.07196*, 2023.

Frank Jäkel, Bernhard Schölkopf, and Felix A Wichmann. Generalization and similarity in exemplar models of categorization: Insights from machine learning. *Psychonomic Bulletin & Review*, 15:256–271, 2008.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Saim Khan, Somesh Singh, Harsha Vardhan Simhadri, Jyothi Vedurada, et al. BANG: billion-scale approximate nearest neighbor search using a single gpu. *arXiv preprint arXiv:2401.11324*, 2024.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019.

John K Kruschke. ALCOVE: an exemplar-based connectionist model of category learning. In *Connectionist psychology*, pp. 107–138. Psychology Press, 2020.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024.

Ritchie Lee, Justin Clarke, Adrian Agogino, and Dimitra Giannakopoulou. Improving trust in deep neural networks with nearest neighbors. In *AIAA Scitech 2020 Forum*, pp. 2098, 2020.

Ken Ziyu Liu. Machine unlearning in 2024, Apr 2024. URL https://ai.stanford.edu/~kzliu/blog/unlearning.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

Douglas L Medin and Marguerite M Schaffer. Context theory of classification learning. *Psychological review*, 85(3):207, 1978.

Kengo Nakata, Youyang Ng, Daisuke Miyashita, Asuka Maki, Yu-Chieh Lin, and Jun Deguchi. Revisiting a knn-based image classification system with high-capacity storage. In *European Conference on Computer Vision*, pp. 457–474. Springer, 2022.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

Robert M Nosofsky. Attention, similarity, and the identification–categorization relationship. *Journal of experimental psychology: General*, 115(1):39, 1986.

Robert M Nosofsky. The generalized context model: An exemplar model of classification. *Formal approaches in categorization*, pp. 18–39, 2011.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113: 54–71, 2019.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018.

Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, and Ozan Sener. Online continual learning without the storage constraint. *arXiv preprint arXiv:2305.09253*, 2023.

J Ross Quinlan. Combining instance-based and model-based learning. In *Proceedings of the tenth international conference on machine learning*, pp. 236–243, 1993.

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 *ICML*, pp. 8748–8763, 2021.

Nazneen Fatema Rajani, Ben Krause, Wengpeng Yin, Tong Niu, Richard Socher, and Caiming Xiong. Explaining and improving model behavior with k nearest neighbor representations. *arXiv preprint arXiv:2010.09030*, 2020.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

Nazanin Mohammadi Sepahvand, Vincent Dumoulin, Eleni Triantafillou, and Gintare Karolina Dziugaite. Data selection for transfer unlearning. *arXiv preprint arXiv:2405.10425*, 2024.

Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on ImageNet. In *International Conference on Machine Learning*, pp. 8634–8644. PMLR, 2020.

Rulin Shao, Jacqueline He, Akari Asai, Weijia Shi, Tim Dettmers, Sewon Min, Luke Zettlemoyer, and Pang Wei Koh. Scaling retrieval-based language models with a trillion-token datastore. *arXiv preprint arXiv:2407.12854*, 2024.

Thalles Silva, Helio Pedrini, and Adín Ramírez Rivera. Learning from memory: Non-parametric memory augmented self-supervised learning of visual features. In *Forty-first International Conference on Machine Learning*, 2024.

Chawin Sitawarin and David Wagner. On the robustness of deep k-nearest neighbors. In *2019 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7. IEEE, 2019.

Sivic and Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1470–1477. IEEE, 2003.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pp. 20827–20840. PMLR, 2022.

Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, et al. Are we making progress in unlearning? findings from the first neurips unlearning competition. *arXiv preprint arXiv:2406.09073*, 2024.

Alexey Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2):58, 2004.

Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–587. IEEE Computer Society, 1991.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, pp. 4126–4142. PMLR, 2021.

Eric Wallace, Shi Feng, and Jordan Boyd-Graber. Interpreting neural networks with nearest neighbors. *EMNLP 2018*, pp. 136, 2018.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019a.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens Van Der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019b.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2021.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Xi Yang, Xiaoting Nan, and Bin Song. D2n4: A discriminative deep nearest neighbor neural network for few-shot space target recognition. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3667–3676, 2020.

Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling ImageNet: from single to multi-labels, from global to localized labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2340–2350, 2021.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12104–12113, 2022.

Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A review on machine unlearning. *SN Computer Science*, 4(4):337, 2023.

Jiaxin Zhuang, Jiabin Cai, Ruixuan Wang, Jianguo Zhang, and Wei-Shi Zheng. Deep kNN for medical image classification. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pp. 127–136. Springer, 2020.

Appendix

We here provide the following supplemental information:

Appendix A Aggregation method comparison on ImageNet-1K

Appendix B Aggregation method comparison on iNaturalist

Appendix C Hyperparameter sensitivity analysis

Appendix D Robustness towards label corruption

Appendix E Hit rate analysis as an upper bound on aggregation accuracy

Appendix F Scaling law details

Appendix G OOD analysis for NINCO dataset

Appendix H Reliability of retreived neighbors

Appendix I Memory pruning details

Appendix J Linear probe details

Appendix K Latency and storage

Appendix L Human experiment to predict model behavior with memory system

Appendix M Calibration analysis

Appendix N ImageNet-A error analysis

Appendix O Compositionality analysis

Appendix P Connection to bias removal and shortcut learning

A. Aggregation method comparison (ImageNet-1K)

Table 4: Benchmarking different aggregation variants at different k thresholds, DinoV2 ViT-L14.

Aggregation	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
PluralityVoting	83.2	82.9	82.6	82.4	82.1	82.0	81.8	81.6	81.5	81.4
DistanceVoting	83.3	83.0	82.7	82.4	82.2	82.1	81.9	81.7	81.6	81.5
SoftmaxVoting	**83.5**	83.5	83.4	83.3	83.2	83.1	83.1	83.0	82.9	82.9
RankVoting	**83.5**	**83.6**	**83.6**	**83.5**	**83.5**	**83.4**	**83.3**	**83.3**	**83.3**	**83.3**

Table 5: Benchmarking different aggregation variants at different k thresholds, DinoV2 ViT-B14.

Aggregation	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
PluralityVoting	81.8	81.4	81.1	80.9	80.7	80.4	80.2	80.0	79.8	79.6
DistanceVoting	81.9	81.5	81.2	81.0	80.8	80.5	80.3	80.0	79.9	79.7
SoftmaxVoting	82.0	82.0	81.9	81.8	81.7	81.7	81.6	81.5	81.3	81.3
RankVoting	**82.1**	**82.2**	**82.1**	**82.0**	**82.0**	**82.0**	**81.9**	**81.9**	**81.9**	**81.9**

Table 6: Benchmarking different aggregation variants at different k thresholds, DinoV2 ViT-S14.

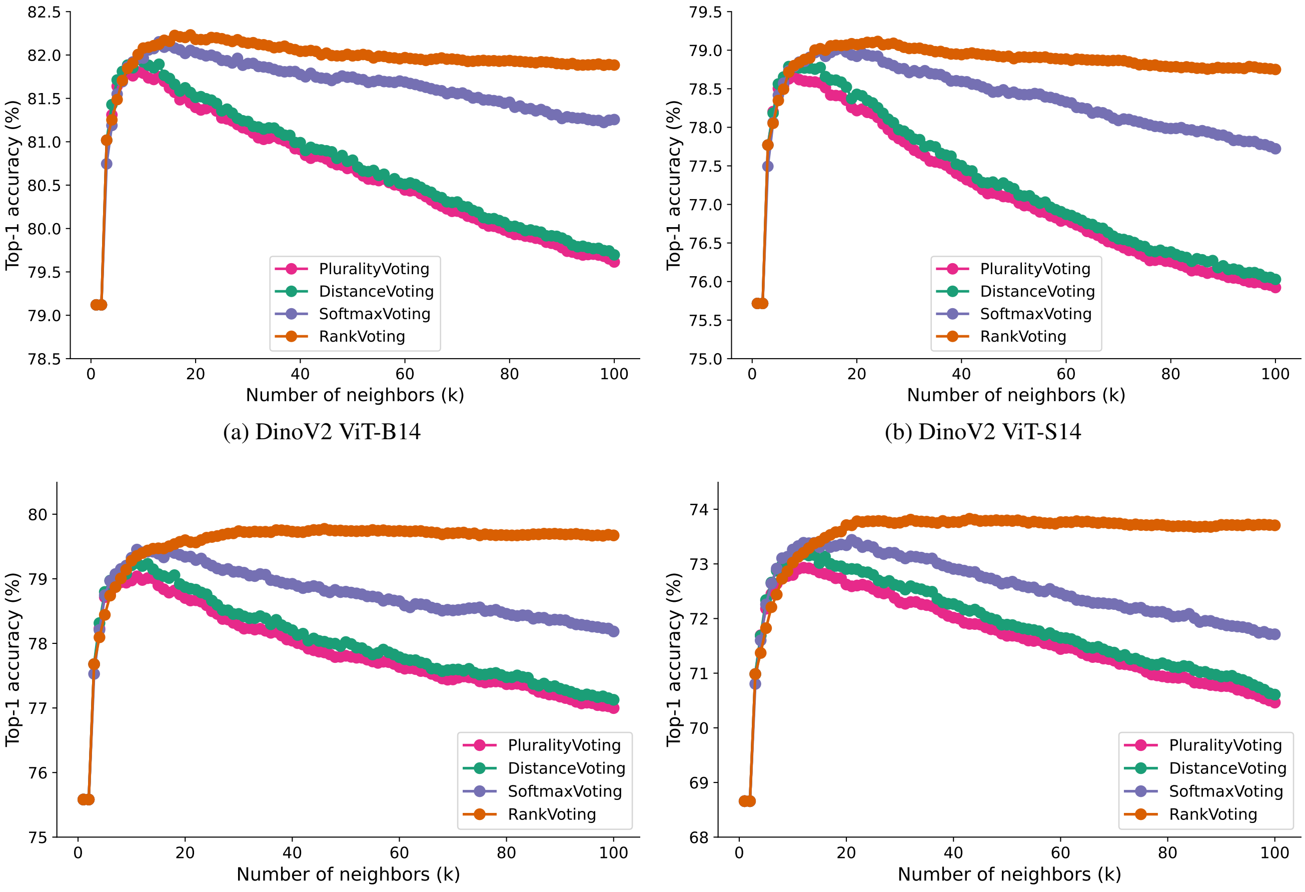
Aggregation	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
PluralityVoting	78.6	78.2	77.8	77.4	77.1	76.8	76.5	76.3	76.1	75.9
DistanceVoting	78.8	78.4	77.9	77.5	77.2	76.9	76.6	76.4	76.2	76.0
SoftmaxVoting	**78.9**	78.9	78.7	78.6	78.5	78.3	78.1	78.0	77.9	77.7
RankVoting	**78.9**	**79.1**	**79.0**	**78.9**	**78.9**	**78.9**	**78.9**	**78.8**	**78.8**	**78.8**

Table 7: Benchmarking different aggregation variants at different k thresholds, CLIP ViT-L14.

Aggregation	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
PluralityVoting	79.0	78.7	78.3	78.0	77.8	77.6	77.4	77.4	77.2	77.0
DistanceVoting	79.2	78.9	78.5	78.2	78.0	77.8	77.6	77.5	77.3	77.1
SoftmaxVoting	**79.3**	79.3	79.1	78.9	78.8	78.7	78.5	78.5	78.4	78.2
RankVoting	**79.3**	**79.6**	**79.7**	**79.7**	**79.7**	**79.7**	**79.7**	**79.7**	**79.7**	**79.7**

Table 8: Benchmarking different aggregation variants at different k thresholds, CLIP ViT-B16.

Aggregation	@10	@20	@30	@40	@50	@60	@70	@80	@90	@100
PluralityVoting	72.8	72.6	72.3	72.0	71.7	71.4	71.2	70.9	70.8	70.5
DistanceVoting	73.1	72.9	72.6	72.3	71.9	71.6	71.4	71.1	70.9	70.6
SoftmaxVoting	**73.3**	73.3	73.1	72.9	72.7	72.5	72.3	72.1	71.9	71.7
RankVoting	73.0	**73.7**	**73.8**	**73.8**	**73.8**	**73.8**	**73.7**	**73.7**	**73.7**	**73.7**



(a) DinoV2 ViT-B14

(b) DinoV2 ViT-S14

(c) CLIP ViT-L14

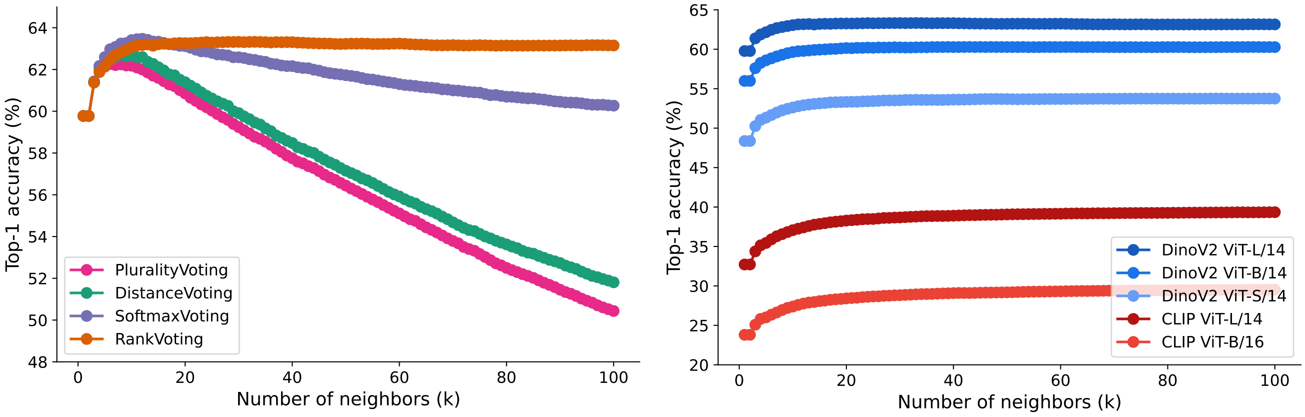
(d) CLIP ViT-B16

Figure 7: Aggregation method comparison on the ImageNet-1K validation set (same as Figure 2a but for other models).

Table 9: Benchmarking different aggregation variants on ImageNet-1K.

Model	Aggegation	IN-val acc (%)
CLIP ViT-L14	CLIP paper (zero-shot)	75.3
CLIP ViT-L14	no aggregation	76.0
CLIP ViT-L14	PluralityVoting	79.2
CLIP ViT-L14	DistanceVoting	79.4
CLIP ViT-L14	SoftmaxVoting	79.6
CLIP ViT-L14	RankVoting	**79.9**
DinoV2 ViT-L14	DinoV2 paper (kNN Softmax)	83.5
DinoV2 ViT-L14	no aggregation	81.1
DinoV2 ViT-L14	PluralityVoting	83.2
DinoV2 ViT-L14	DistanceVoting	83.3
DinoV2 ViT-L14	SoftmaxVoting	**83.6**
DinoV2 ViT-L14	RankVoting	**83.6**

B. Aggregation method comparison (iNaturalist)

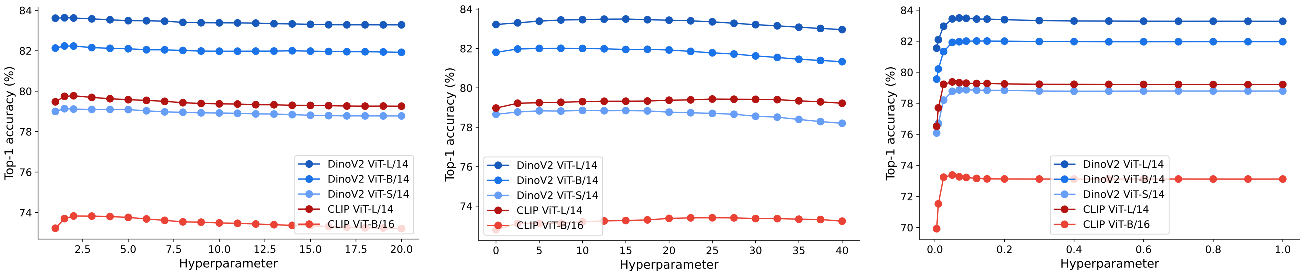


(a) Comparing aggregation methods, DinoV2 ViT-L14

(b) RankVoting (across models)

Figure 8: **Aggregating information across retrieved memory samples on iNaturalist.** Same as Figure 2 but for iNaturalist instead of ImageNet. **(left)** Existing aggregation methods (PluralityVoting, DistanceVoting and SoftmaxVoting) are overconfident in distant neighbors, resulting in the paradox of decaying iNaturalist accuracy with more information. **(right)** This is not the case for RankVoting which shows strong and stable performance across models and choices of k.

C. Hyperparameter sensitivity analysis



(a) RankVoting (α)

(b) DistanceVoting (ξ)

(c) SoftmaxVoting (τ)

Figure 9: **Sensitivity to hyperparameters for different aggregation methods.** Apart from PluralityVoting, all aggregation methods described in Section 2.2 have a hyperparameter (α for RankVoting, τ for SoftmaxVoting). For each model and method, we here plot the maximum performance when aggregating using a certain method, sweeping over the number of neighbors from 1 to 100, as a function of the hyperparameter. This analysis is performed to understand how sensitive the respective method is to the choice of the hyperparameter. Note that the x range is different since for instance the temperature parameter in SoftmaxVoting ranges from $[0, 1]$ while RankVoting for $\alpha = 0$ is undefined (division by zero). We therefore evaluate a broad range for each method and find that all methods have a regime in which they are relatively stable irrespective of the hyperparameter choice. Since DistanceVoting as implemented by Khandelwal et al. (2019) does not have a hyperparameter, we added a temperature-style parameter ξ for the purpose of this comparison by setting $w_i = \exp\big(-\mathsf{dist}(\tilde{\boldsymbol{z}}, \boldsymbol{z}_{[i]})\big)^{\xi}$.

D. Robustness towards label corruption

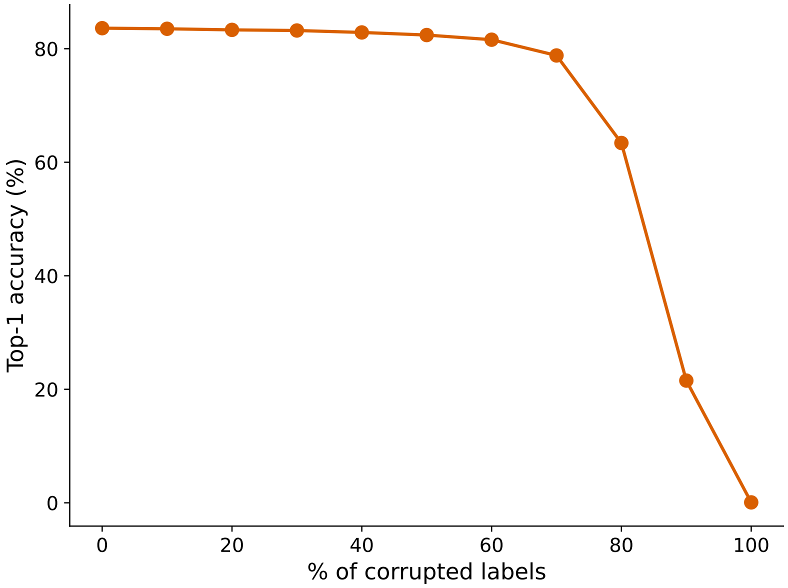


Figure 10: **Robustness towards label corruption.** How robust is a visual memory towards corrupted labels in the memory bank? This plot shows top-1 RankVoting accuracy on the ImageNet validation set as a function of how many labels in the memory (containing ImageNet-1K training set features via DinoV2 ViT-L/14) are corrupted, i.e., assigned to a random class. Intriguingly, performance stays almost unchanged all the way to about 60% (!) corrupted (random) labels in the database.

E. Hit rate analysis

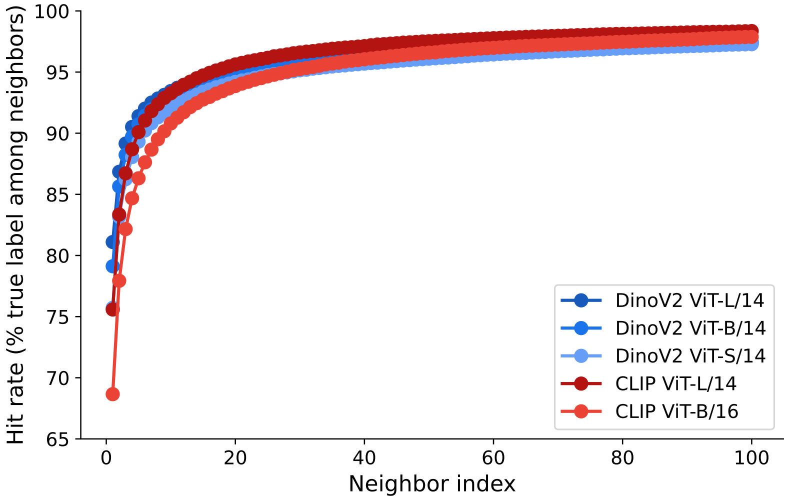


Figure 11: **Hit rate.** This plot shows the probability of the true label being contained in list of labels of the first k retrieved neighbors on ImageNet-1K, for five different models and $k \in [1, 100]$. With 100 neighbors, the hit rate approaches 98% for the best model. Conceptually, this is a very high upper bound on the performance that can be achieved by a given featurizer via nearest neighbor retrieval.

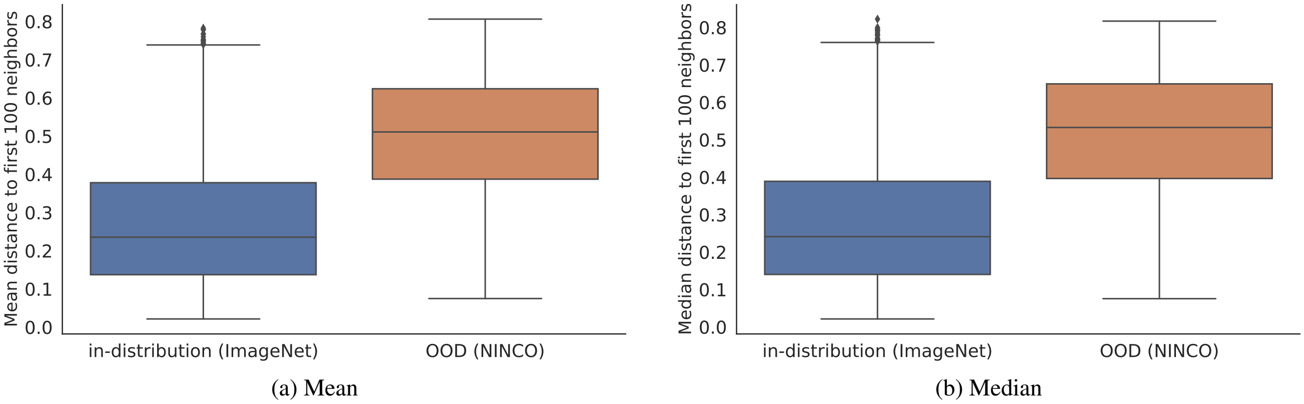
F. Scaling law

As we mentioned in Section 3.3, we found that a logarithmic form fits the data well between $\log_{10}$(memory size) and $\log_{10}$(error rate). Specifically, we found the following functional forms for DinoV2 ViT S14 and DinoV2 ViT L14 respectively via `np.polyfit(x, y, dim=1)`:

$$\begin{aligned} \textbf{DinoV2 ViT L14:} \quad & y = -0.9434 \cdot \log_{10}(x) + 2.0704 \\ \textbf{DinoV2 ViT S14:} \quad & y = -1.0942 \cdot \log_{10}(x) + 2.3187 \end{aligned}$$

where $x = \log_{10}$(memory-size) and $y = \log_{10}$(error-rate), where memory-size $\in [10^3, 10^9]$ and error-rate in $[0, 100]$.

G. OOD analysis for NINCO dataset



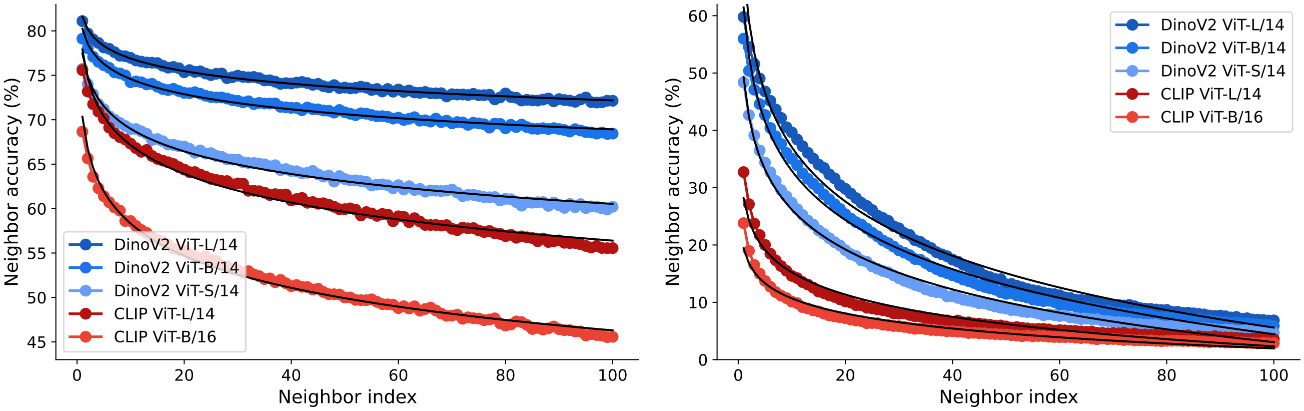
(a) Mean

(b) Median

Figure 12: **Distance comparison: the NINCO OOD samples are indeed out-of-distribution for the model.** In Section 3.1, we described that we can simply plug new out-of-distribution classes into memory and still perform well on both existing data as well as the new classes. This boxplot confirms that the added samples from the NINCO dataset (Bitterwolf et al., 2023) are indeed out-of-distribution for DinoV2 ViT-L14: The mean **(left)** and median **(right)** distances from query to the first 100 neighbors are substantially lower for ImageNet validation images than for OOD samples from NINCO.

Figure 12 confirms that there is a distribution difference between in-distribution data (ImageNet-1K) and OOD data (NINCO). That said, while a distribution shift exists, it is possible that individual NINCO samples were part of the training set for DinoV2. Test-set contamination is generally a concern when working with models trained on large-scale datasets, since test samples may occur as exact, semantic or near-duplicates in large training datasets (e.g. Abbas et al., 2023b). For instance, NINCO contains samples from Food-101 (Bossard et al., 2014) which are also part of LVD-142M dataset used to train DinoV2. That said, the NINCO samples belong to classes which are definitely not part of the ImageNet-train set which serves as a memory bank for our experiments, as ensured by the NINCO dataset collection process (Bitterwolf et al., 2023).

H. Reliability of retrieved memory samples



(a) ImageNet

(b) iNaturalist

Figure 13: **Reliability of retrieved memory samples.** This plot visualizes the ImageNet **(left)** and iNaturalist **(right)** top-1 validation accuracy of a single retrieved neighbor depending on the index of the neighbor (index 0: nearest neighbor). In both datasets and across models, the decrease in accuracy with increasing neighbor index follows smooth trajectories and can be approximated by a two-parameter logarithmic fit (black lines). Figure 13a is the same as Figure 1.

I. Memory pruning

For memory pruning from Section 3.6, we implemented two pruning methods: removing unreliable neighbors from memory entirely ("hard memory pruning"), and reducing their weight ("soft memory pruning"). We report results on the ImageNet validation set with a (potentially pruned) ImageNet-train set in memory. For hard pruning, we excluded images from memory that contributed to a wrong decision at least 128 times (this meant excluding 26,257 images for DinoV2 ViT-L14), based on querying the ImageNet-train set against a memory consisting of the ImageNet-train set and querying 100 neighbors for each sample. In order to obtain a fair comparison, instead of reporting accuracies for an arbitrary choice of k (the number of neighbors) we instead evaluate accuracy for each k in $[1, 100]$ and report the maximum accuracy obtained in Table 3. This ensures that differences in observed accuracy can indeed be attributed to memory pruning, as opposed to a choice of k. For soft pruning, instead of excluding unreliable neighbors entirely as in hard pruning, the neighbor weights (1.0 for PluralityVoting, or a rank-based weight in case of RankVoting) are instead multiplied by a reliability factor γ with $\gamma = \frac{d}{c+v}$ where v is the number of times the image contributed to a wrong decision on the ImageNet-train set, $c = 1$ to avoid division by zero, and $d = 1.75$. This results, for instance, in $\gamma = 0.88$ for images that only contribute to a single wrong decision; in $\gamma = 0.16$ for images that contribute to ten wrong decisions, and in $\gamma = 0.02$ for images that contribute to 100 wrong decisions on the training set. Images that never contributed to any wrong decision are assigned $\gamma = 1.0$, i.e. their default weight remains unchanged.

J. Linear probe details

For the linear probe results reported in the paper, we directly used the results that were reported in the DinoV2 and CLIP papers. For DinoV2, the authors froze the model backbone and trained the linear layers for 12500 iterations using SGD. Instead of training a single time, they performed a full grid search sweep over three settings (output layers in 1, 4; pooling token concatenation in yes, no, and 13 different learning rates), resulting in 52 linear probes. Then, the authors evaluated the ImageNet validation accuracy for all of those 52 probes and only reported the highest one, as described in Appendix B.3 of the DinoV2 paper. Some may call this test set tuning or double dipping; the DinoV2 paper describes it as "common practice" (Oquab et al., 2023, p. 31). CLIP linear probe results are based on a logistic regression classifier learned using scikit-learn's L-BFGS implementation, and hyperparameter sweeps are performed on a held-out set not used for evaluation, according to Radford et al. (2021).

K. Latency and storage

Latency. Nearest neighbor retrieval, fortunately, does not need to reinvent the wheel but can, instead, build on top of highly optimized workloads and libraries such as the ScaNN library (Guo et al., 2020). The ScaNN github README shows a latency comparison; with the requirement of perfect recall a million-size memory can handle roughly 500-600 queries per second. It may be worth mentioning that searching a large database can be done on CPUs and can be heavily parallelized.

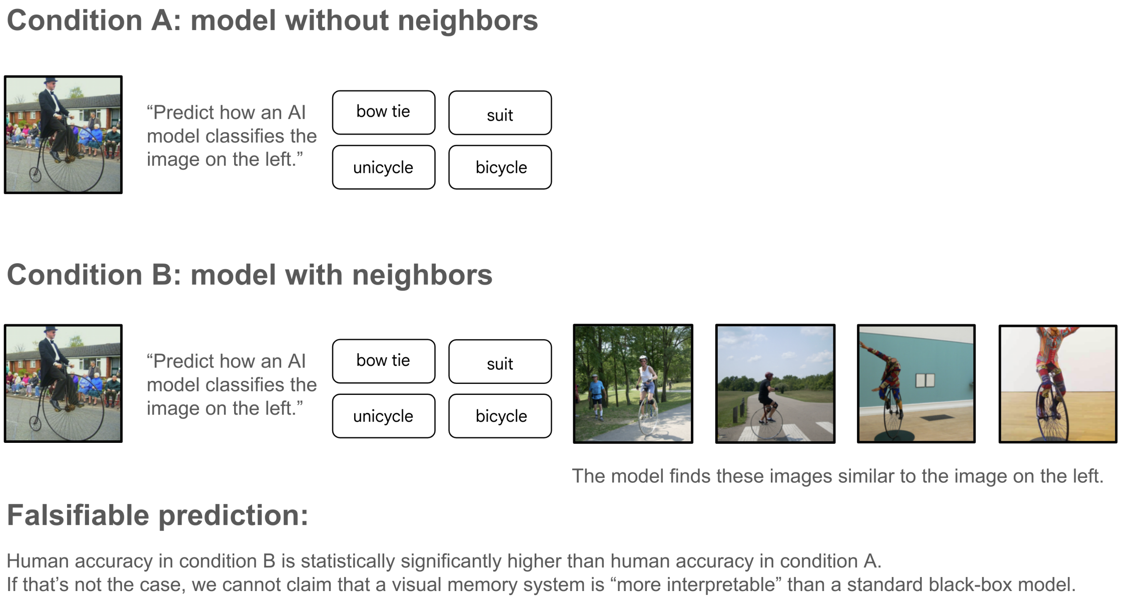
Model	IN-train features (GB)	IN-val features (MB)
DinoV2 ViT-L/14	4.9	197
DinoV2 ViT-B/14	3.7	148
DinoV2 ViT-S/14	1.9	75
CLIP ViT-L/14	3.7	148
CLIP ViT-B/16	2.5	100

Table 10: **Storage requirements for ImageNet features.** Storing features in a memory database requires only about 1–3% of the space that is needed to store the dataset (154.6 GB for ImageNet-train, 6.0 GB for ImageNet-validation).

Storage. In addition to latency, storage is another very practical consideration: How much does it take to store features for a large database? To put things into perspective, the ImageNet training dataset requires 154.6 GB of storage, and the ImageNet validation dataset requires 6.0 GB of storage. In comparison, as shown in Appendix K, storing DinoV2 or CLIP features for the entire ImageNet training dataset only requires between 1.9 and 4.9 GB of storage space. Thus compared to storing the training dataset, the model features account for only 1–3% of this size. This means that after constructing the memory, one may decide to keep the dataset which adds 1–3% of storage, or one may decide to delete the dataset only keeping the features which saves 97–99% of storage (compared to the dataset storage requirement). The ratio of features requiring 1–3% of the dataset size doesn't change with dataset scale since it only depends on the embedding model, thus this ratio would hold for very small datasets just as it would for a billion-scale dataset.

L. Human experiment to predict model behavior with memory system

We conducted a small human experiment to quantify how much, if at all, a memory-based system improves interpretability as operationalized by helping humans predict model behavior (as opposed to a standard black-box model). Figure 14 outlines the experimental setup.


Condition A: model without neighbors

"Predict how an AI model classifies the image on the left."

Condition B: model with neighbors

"Predict how an AI model classifies the image on the left."

The model finds these images similar to the image on the left.

Falsifiable prediction:

Human accuracy in condition B is statistically significantly higher than human accuracy in condition A.
If that's not the case, we cannot claim that a visual memory system is "more interpretable" than a standard black-box model.

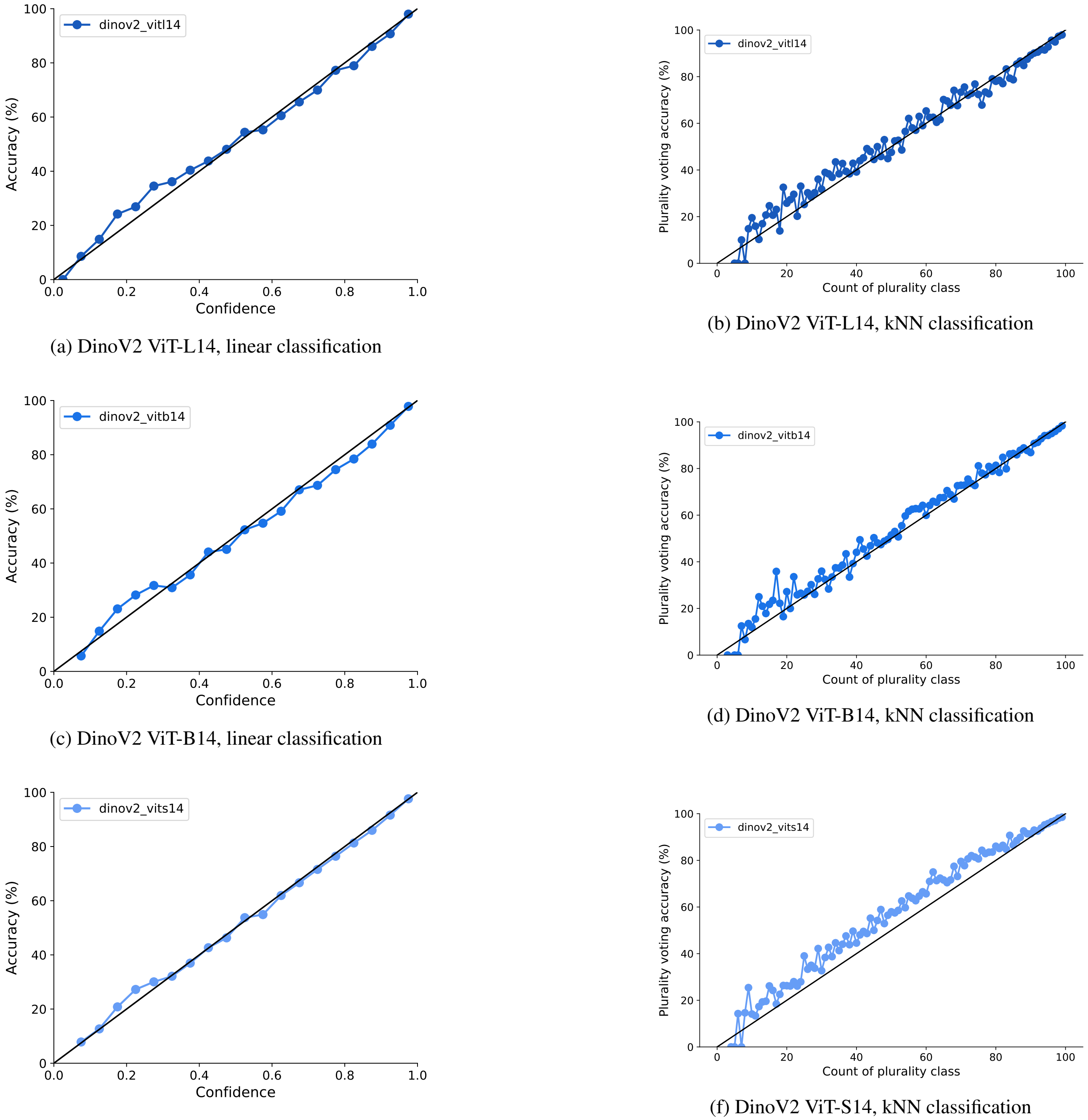
Figure 14: **Human experiment setup.** Conditions A and B were presented on a computer screen in separate trials.

Given 4 label choices (guessing accuracy 25%), human accuracy is 56% in the case of black-box predictions (no neighbor information). With access to four nearest neighbor images from our memory-based system (just the neighbor images but not their labels), human accuracy is at 83%. This represents an absolute improvement of +27% and a relative improvement of +67% in human prediction accuracy, providing strong, falsifiable evidence in favor of the statement that a memory-based model is more interpretable.

Experimental details:

- The accuracy difference is statistically significant (p ¡ 0.001).
- Featurizer = DinoV2 ViT-L14 (i.e. the best performing model).
- Dataset: randomly selected ImageNet-A test images
- Nearest neighbors for condition B: from ImageNet-train.
- 4 label choices per trial including ground truth label, model-predicted label (if different), and the remaining 2-3 labels were plausible alternatives based on top CLIP predictions for the test image. Label order randomized.

M. Calibration analysis



(a) DinoV2 ViT-L14, linear classification

(b) DinoV2 ViT-L14, kNN classification

(c) DinoV2 ViT-B14, linear classification

(d) DinoV2 ViT-B14, kNN classification

(e) DinoV2 ViT-S14, linear classification

(f) DinoV2 ViT-S14, kNN classification

Figure 15: **How well are predictions calibrated?** Left column: Accuracy vs. confidence from Softmax of linear classifier for three DinoV2 variants. Right column: Accuracy vs. count of plurality class among first 100 neighbors for the same three DinoV2 variants. A DinoV2-based kNN classifier is well calibrated, as is the DinoV2 softmax.

N. ImageNet-A error analysis

As shown in Figure 6, many “errors” on ImageNet-A appear to be perfectly reasonable predictions that are caused by dataset label issues as opposed to model mistakes. More randomly selected ImageNet-A samples, along with nearest neighbors, are shown in Figure 16. To quantify the issue, we performed a human experiment on a randomly selected subset of ImageNet-A images (N=100) where the dataset label and the prediction from DinoV2 ViT-L14 with JFT memory disagree. We presented the image alongside the original ImageNet-A label and our model-predicted label to three human observers, asking them to identify which of the labels best describes the image (of course, without telling them which of the labels is the dataset label). The result was that in 39.3% (!) of cases (std: $\pm 1.25\%$), the DinoV2 label was assessed as being better/more suitable than the original dataset label—i.e., roughly 2 out of 5 model “errors” are in fact dataset label errors, quantifying the ImageNet-A label quality issue we alluded to in Figure 6. This percentage can be used to estimate how correcting problematic labels influences performance. Instead of the original model’s 61.1% accuracy on ImageNet-A, due to label errors the ‘corrected’ accuracy is instead 76.4% (a delta of $+15.3\%$ in absolute terms or $+25.0\%$ in relative terms).

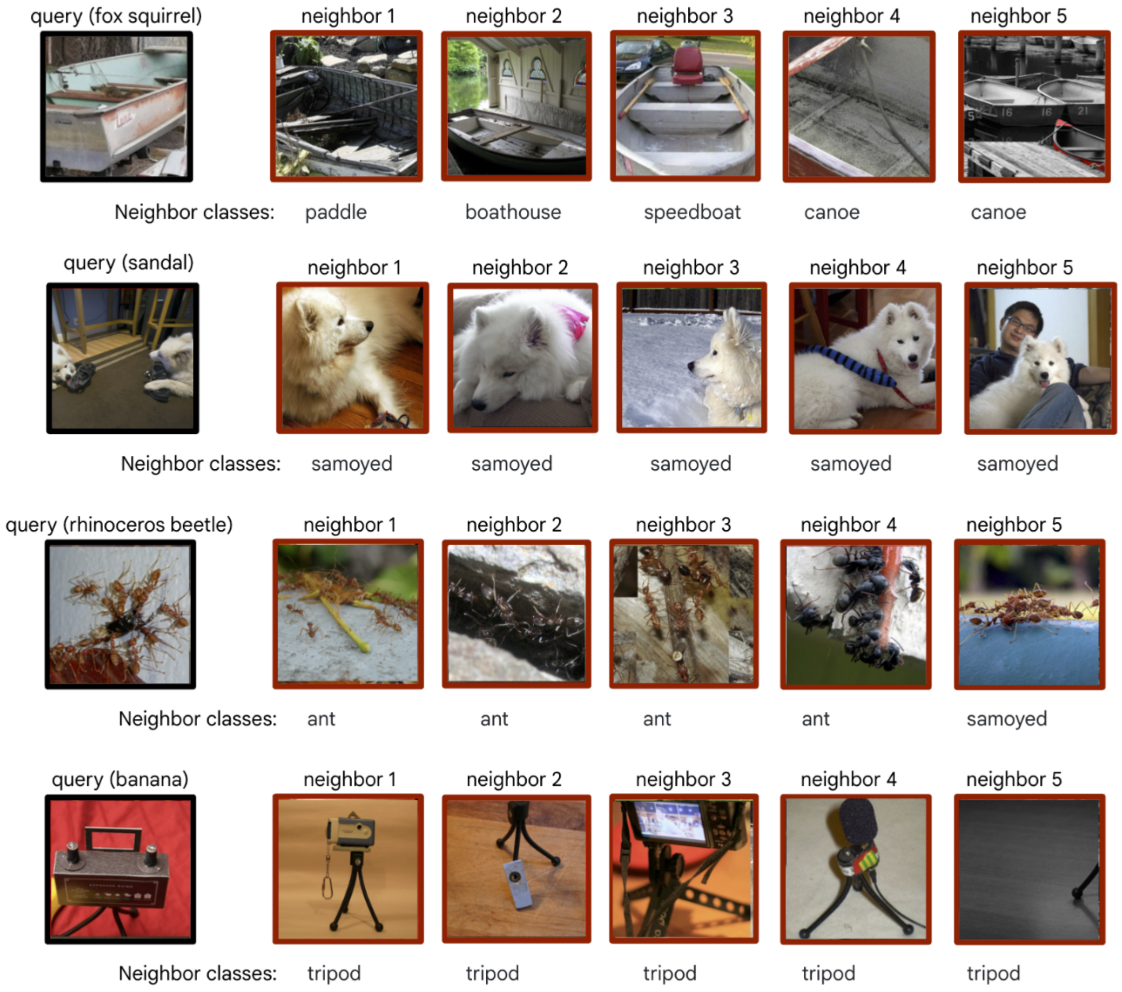


Figure 16: **Interpretable decision-making.** A retrieval-based visual memory enables a clear visual understanding of why a model makes a certain prediction. Here, we show four randomly selected misclassified query images from ImageNet-A (Hendrycks et al., 2021) along with five nearest neighbors from DinoV2 ViT-L14 using the ImageNet-1K training set as visual memory. All labels are from the respective datasets (ImageNet-A for query and ImageNet-train for neighbors). While all neighbors visually look reasonable, not all labels do.

O. Compositionality analysis

A flexible visual memory also provides a path to analyze representations of various models, particularly, how different models represent multiple concepts in an image. We study this for an ImageNet-train visual memory of DinoV2 ViT-L14 and CLIP ViT-L14. We use manually selected query images from outside the ImageNet dataset that have multiple objects from the ImageNet labels. We query the visual memory for nearest neighbors of the query image. Subsequently, we obtain the *residual image* by subtracting the features of the nearest neighbor from the features of the query image. We, then, obtain the nearest neighbors for the residual image from the visual memory. We plot the results in Figure 17 which shows that DinoV2 ViT-L14 and CLIP ViT-L14 represent concepts in their features in a different manner. The nearest neighbors for DinoV2 are mostly images with a single concept (or object) from the query image. The residual image, subsequently, leads to nearest neighbors dominated by another single object in the query image. In contrast, CLIP often results in nearest neighbors that are generally a blend of concepts from the query image. These qualitative explorations are simple demonstrations of the advantages of an interpretable decision-making process provided by a flexible visual memory.

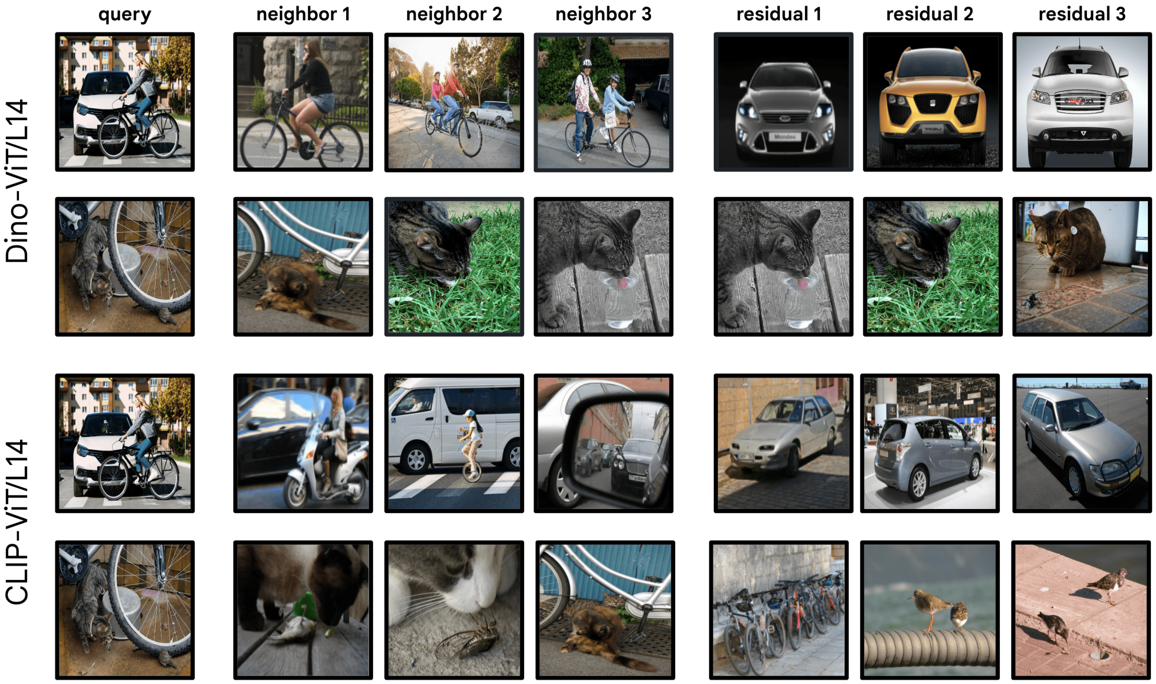


Figure 17: **Compositionality of representations.** The first column indicates a query image; the next three columns are the three nearest neighbors from the training set. The last three columns are the *residual* images, obtained by subtracting the features of the nearest neighbor (2nd column from the left) from the features of the query image (1st column from the left). The nearest neighbors for DinoV2 are mostly images with a single concept (or object) from the query image. The residual image, subsequently, leads to nearest neighbors dominated by another single object in the query image. In contrast, CLIP often finds neighbors that are a blend of concepts from the query image.

P. Connection to bias removal and shortcut learning

This section contains a brief discussion on a connection between our method and bias removal—since the additional flexibility that our memory approach brings (compared to a standard, inflexible classifier), it is an open question whether this could enable better bias removal.

If the image encoder exploited a shortcut during training, this will influence image similarity and thus nearest neighbor selection. There are cases where bias removal is possible, and cases where it is impossible:

- Removal impossible: If the encoder is biased towards textures, a test image of “cat shape + elephant texture” (cf. Geirhos et al., 2019, Figure 1) would pull up elephant nearest neighbors, and removing all elephants from memory would come at an unreasonably high cost (not being able to identify elephants anymore). Here, encoder-level debiasing is necessary.

- Removal possible: If only a part of the memory is biased, memory-level debiasing is feasible. If “fingers” are shortcut predictors for “fish” (due to a dataset bias from proud fishermen holding their catch into the camera, cf. Brendel & Bethge, 2019, Figure 3), then this bias could indeed be rectified by removing the biased “fish+finger” subset from memory. Afterwards, images with “fingers” would no longer lead to “fish” nearest neighbors, demonstrating successful bias removal.