Beyond neural scaling laws: beating power law scaling via data pruning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Widely observed neural scaling laws, in which error falls off as a power of the training set size, model size, or both, have driven substantial performance improvements in deep learning. However, these improvements through scaling alone require considerable costs in compute and energy. Here we focus on the scaling of error with dataset size and show how both in theory and practice we can break beyond power law scaling and reduce it to exponential scaling instead if we have access to a high-quality data pruning metric that ranks the order in which training examples should be discarded to achieve any pruned dataset size. We then test this new exponential scaling prediction with pruned dataset size empirically, and indeed observe better than power law scaling performance on ResNets trained on CIFAR-10, SVHN, and ImageNet. Given the importance of finding high-quality pruning metrics, we perform the first large-scale benchmarking study of ten different data pruning metrics on ImageNet. We find most existing high performing metrics scale poorly to ImageNet, while the best are computationally intensive and require labels for every image. We therefore developed a new simple, cheap and scalable self-supervised pruning metric that demonstrates comparable performance to the best supervised metrics. Overall, our work suggests that the discovery of good data-pruning metrics may provide a viable path forward to substantially improved neural scaling laws, thereby reducing the resource costs of modern deep learning.

1 Introduction

Empirically observed neural scaling laws [1, 2, 3, 4, 5, 6, 7] in many domains of machine learning, including vision, language, and speech, demonstrate that test error often falls off as a power law with either the amount of training data, model size, or compute. Such power law scaling has motivated significant societal investments in data collection, compute, and associated energy consumption. However, power law scaling is extremely weak and unsustainable. For example, a drop in error from 3% to 2% might require an *order of magnitude* more data, compute, or energy. In language modeling with large transformers, a drop in cross entropy loss from about 3.4 to 2.8 nats requires *10 times* more training data (Fig. 1 in [2]). Also, for large vision transformers, an additional 2 *billion* pre-training data points (starting from 1 billion) leads to an accuracy gain on ImageNet of a few percentage points (Fig. 1 in [6]). Here we ask whether we might be able to do better. For example, can we achieve exponential scaling instead, with a good strategy for selecting training examples? Such vastly superior scaling would mean that we could go from 3% to 2% error by only adding a few carefully chosen training examples, rather than collecting $10\times$ more random ones.

Focusing on scaling of performance with training dataset size, we demonstrate that exponential scaling is possible, both in theory and practice. The key idea is that power law scaling of error

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

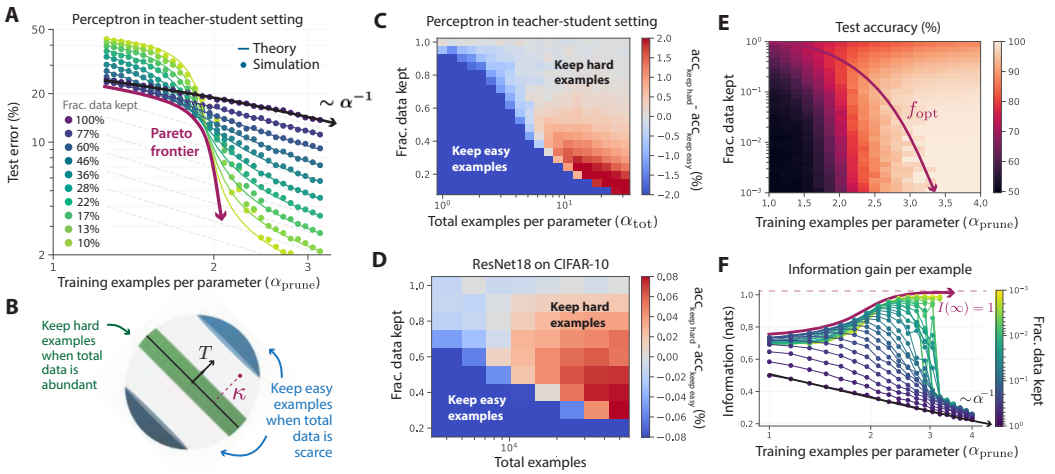


Figure 1: Our analytic theory of data pruning predicts that power law scaling of test error with respect to dataset size can be beaten. **A:** Test error as a function of $\alpha_{\text{prune}} = f\alpha_{\text{tot}}$ with $\theta = 0$. We observe an excellent match between our analytic theory (solid curves) and numerical simulations of perceptron learning at N=200 (dots). The red curve indicates the Pareto optimal test error ε achievable from a tradeoff between α_{tot} and f at fixed α_{prune}. **B:** We find that when data is abundant (scarce) corresponding to large (small) α_{tot}, the better pruning strategy is to keep the hard (easy) examples. **C:** Color indicates difference in test error in keeping hard versus easy examples, revealing the change in strategy in (B). **D:** We tested this prediction on a ResNet18 trained on CIFAR-10, finding remarkably the same shift in optimal pruning strategy under the EL2N metric. **E:** Test accuracy as a function of f and α_{prune}. For every fixed α_{prune}, there is an optimal f_{opt} (purple curve). **F:** $I(\alpha_{\text{prune}})$ for different f.

with respect to data suggests that many training examples are highly redundant. Thus one should in principle be able to prune training datasets to much smaller sizes and train on the smaller pruned datasets without sacrificing performance. Indeed some recent works [8, 9, 10] have demonstrated this possibility by suggesting various metrics to sort training examples in order of their difficulty or importance, ranging from easy or redundant examples to hard or important ones, and pruning datasets by retaining some fraction of the hardest examples. However, these works leave open fundamental theoretical and empirical questions: When and why is successful data pruning possible? What are good metrics and strategies for data pruning? Can such strategies beat power law scaling? Can they scale to ImageNet? Can we leverage large *unlabeled* datasets to successfully prune labeled datasets? We address these questions through both theory and experiment. Our main contributions are:

1. Employing statistical mechanics, we develop a new analytic theory of data pruning in the student-teacher setting for perceptron learning, where examples are pruned based on their teacher margin, with large (small) margins corresponding to easy (hard) examples. Our theory quantitatively matches numerical experiments and reveals two striking predictions:
   (a) The optimal pruning strategy changes depending on the amount of initial data; with abundant (scarce) initial data, one should retain only hard (easy) examples.
   (b) Exponential scaling is possible with respect to pruned dataset size provided one chooses an increasing Pareto optimal pruning fraction as a function of initial dataset size.
2. We show that the two striking predictions derived from theory hold also in practice in much more general settings. Indeed we empirically demonstrate signatures of exponential scaling of error with respect to pruned dataset size for ResNets trained from scratch on SVHN, CIFAR-10 and ImageNet, and Vision Transformers fine-tuned on CIFAR-10.
3. Motivated by the importance of finding good quality metrics for data pruning, we perform a large scale benchmarking study of 10 different data pruning metrics at scale on ImageNet, finding that most perform poorly, with the exception of the most compute intensive metrics.
4. We leveraged self-supervised learning (SSL) to developed a new, cheap *unsupervised* data pruning metric that does *not* require labels, unlike prior metrics. We show this unsupervised metric performs comparably to the best supervised pruning metrics that require labels and

much more compute. This result opens the door to the exciting possibility of leveraging pre-trained foundation models to prune new datasets even before they are labeled.

Overall these results shed theoretical and empirical insights into the nature of data in deep learning and our ability to prune it, and suggest our current practice of collecting extremely large datasets may be highly inefficient. Our initial results in beating power law scaling motivate further studies and investments in not just inefficently collecting large amounts of random data, but rather, intelligently collecting much smaller amounts of carefully selected data, potentially leading to the creation and dissemination of *foundation datasets*, in addition to foundation models [11].

2 Background and related work

Our work brings together 3 largely disparate strands of intellectual inquiry in machine learning: (1) explorations of different metrics for quantifying differences between individual training examples; (2) the empirical observation of neural scaling laws; and (3) the statistical mechanics of learning.

2.1 Pruning metrics: not all training examples are created equal

Several recent works have explored various metrics for quantifying individual differences between data points. To describe these metrics in a uniform manner, we will think of all of them as ordering data points by their difficulty, ranging from "easiest" to "hardest." When these metrics have been used for data pruning, the hardest examples are retained, while the easiest ones are pruned away.

EL2N scores. For example [9] trained small ensembles (of about 10) networks for a very short time (about 10 epochs) and computed for every training example the average L_2 norm of the error vector (EL2N score). Data pruning by retaining only the hardest examples with largest error enabled training from scratch on only 50% and 75% of CIFAR-10 and CIFAR-100 respectively without any loss in final test accuracy. However the performance of EL2N on ImageNet has not yet been explored.

Forgetting scores and classification margins. [8] noticed that over the entire course of training, some examples are learned early and never forgotten, while others can be learned and unlearned (i.e. forgotten) repeatedly. They developed a forgetting score which measures the degree of forgetting of each example. Intuitively examples with low (high) forgetting scores can be thought of as easy (hard) examples. [8] explored data pruning using these metrics, but not at ImageNet scale.

Memorization and influence. [12] defined a memorization score for each example, corresponding to how much the probability of predicting the correct label for the example increases when it is present in the training set relative to when it is absent; a large increase means the example must be memorized (i.e. the remaining training data do not suffice to correctly learn this example). Additionally [12] also considered an influence score that quantifies how much adding a particular example to the training set increases the probability of the correct class label of a test example. Intuitively, low memorization and influence scores correspond to easy examples that are redundant with the rest of the data, while high scores correspond to hard examples that must be individually learned. [12] did not use these scores for data pruning as their computation is expensive. We note since memorization explicitly approximates the increase in test loss due to removing each individual example, it is likely to be a good pruning metric (though it does not consider interactions).

Ensemble active learning. Active learning iterates between training a model and selecting new inputs to be labeled [13, 14, 15, 16, 17]. In contrast, we focus on data pruning: one-shot selection of a data subset sufficient to train to high accuracy from scratch. A variety of coreset algorithms (e.g. [18]) have been proposed for this, but their computation is expensive, and so data-pruning has been less explored at scale on ImageNet. An early clustering approach [19] allowed training on 90% of ImageNet without sacrificing accuracy. Notably [10] reduced this to 80% by training a large ensemble of networks on ImageNet and using ensemble uncertainty to define the difficulty of each example, with low (high) uncertainty corresponding to easy (hard) examples. We will show how to achieve similar pruning performance without labels or the need to train a large ensemble.

Diverse ensembles (DDD). [20] assigned a score to every ImageNet image, given by the number of models in a diverse ensemble (10 models) that misclassified the image. Intuitively, low (high) scores correspond to easy (hard) examples. The pruning performance of this metric remains unexplored.

Summary. We note: (1) only one of these metrics has tested well for its efficacy in data pruning at scale on ImageNet; (2) *all* of these metrics require label information; (3) there is no theory of when and why data pruning is possible for any of these metrics; and (4) none of these works suggest the possibility of exponential scaling. We thus go beyond this prior work by benchmarking the data pruning efficacy of not only these metrics but also a new unsupervised metric we introduce that does not require label information, all at scale on ImageNet. We also develop an analytic theory for data-pruning for the margin metric that predicts not only the possibility of exponential scaling but also the novel finding that retaining easy instead of hard examples is better when data is scarce.

2.2 Neural scaling laws and their potential inefficiency

Recent work [1, 2, 3, 4, 5, 7] has demonstrated that test loss $\mathcal{L}$ often falls off as a power law with different resources like model parameters (N), number of training examples (P), and amount of compute (C). However, the exponents ν of these power laws are often close to 0, suggesting potentially inefficient use of resources. For example, for large models with lots of compute, so that the amount of training data constitutes a performance bottleneck, the loss scales as $\mathcal{L} \approx P^{-\nu}$. Specifically for a large transformer based language model, $\nu = 0.095$, which implies *an order of magnitude* increase in training data drops cross-entropy loss by only about 0.6 nats (Fig. 1 in [2]). In neural machine translation experiments ν varies across language pairs from 0.35 to 0.48 (Table 1 in [4]). Interestingly, [7] explored a fixed computation budget C and optimized jointly over model size N and training set size P, revealing that scaling both N and P commensurately as C increases is compute optimal, and can yield smaller high performing models (trained on more data) than previous work. Nevertheless, for a transformer based language model, a $100\times$ increase in compute, corresponding to $10\times$ increases in *both* model size and training set size, leads to a drop in cross-entropy loss of only about 0.5 nats (Fig. 2 in [7]). Similar slow scaling holds for large vision transformers where adding 2 billion pre-training images reduces ImageNet performance by a few percentage points (Fig. 1 in [6]). While all of these results constitute significant improvements in performance, they do come at a substantial resource cost whose fundamental origin arises from power law scaling with small exponents. Here, in the specific case of data scaling, we explore whether we can beat power law scaling through careful data selection.

2.3 Statistical mechanics of perceptron learning

Statistical mechanics has long played a role in analyzing machine learning problems (see e.g. [21, 22, 23, 24] for reviews). One of the most fundamental applications is perceptron learning in the student-teacher setting [25, 26], in which random i.i.d. Gaussian inputs are labeled by a teacher perceptron to construct a training set. The test error for another student perceptron learning from this training set then scales as a power law with exponent -1 for such data. Such perceptrons have also been analyzed in an active learning setting where the learner is free to design *any* new input to be labeled [27, 28], rather than choose from a fixed set of inputs, as in data-pruning. Recent work [29] has analyzed this scenario but focused on message passing algorithms that are tailored to the case of Gaussian inputs and perceptrons, and are hard to generalize to real world settings. In contrast we analyze margin based pruning algorithms that are used in practice in diverse settings, as in [8, 9].

3 An analytic theory of data pruning

To better understand data pruning, we employed the replica method from statistical mechanics [30] to develop an analytic theory of pruning for the perceptron in the student-teacher setting [21] (see App. A for detailed derivations of all results). Consider a training dataset of P examples $\{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\ldots,P}$ where $\mathbf{x}^\mu \in \mathbb{R}^N$ are i.i.d. zero mean unit variance random Gaussian inputs and $y^\mu = \text{sign}(\mathbf{T} \cdot \mathbf{x}^\mu)$ are labels generated by a teacher perceptron with weight vector $\mathbf{T} \in \mathbb{R}^N$. We work in the high dimensional statistics limit where $N, P \to \infty$ but the ratio $\alpha_{\text{tot}} = \frac{P}{N}$ of the number of total training examples to parameters remains $O(1)$. We then consider a pruning algorithm used in [8, 9], namely: (1) train a probe student perceptron for very few epochs on the training data, obtaining weights $\mathbf{J}_{\text{probe}}$; (2) compute the margin $m^\mu = \mathbf{J}_{\text{probe}} \cdot (y^\mu \mathbf{x}^\mu)$ of each training example, where large (small) margins correspond to easy (hard) examples; (3) construct a pruned dataset of size $P_{\text{prune}} = fP$, where f is the fraction of examples kept, by retaining the P_{prune} hardest examples, (4) train a new perceptron to completion on the smaller dataset with a smaller ratio $\alpha_{\text{prune}} = \frac{P_{\text{prune}}}{N}$ of examples to parameters.

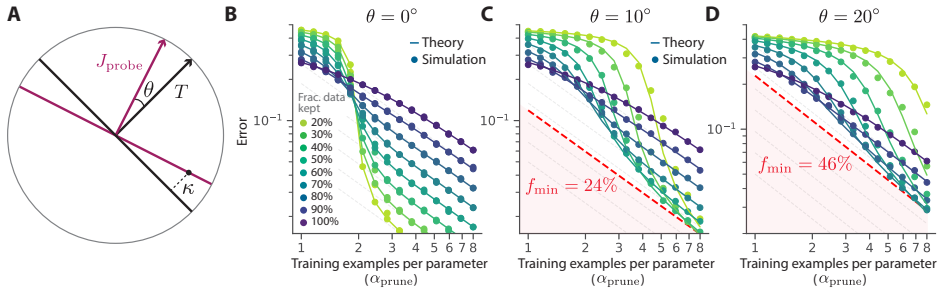


Figure 2: Data pruning with an imperfect metric. **A:** Weight vectors and decision boundaries for a teacher (black) and probe student (red) separated by angle θ. The black point has margin 0 (κ) w.r.t. the probe (teacher). **B–D:** Test error as a function of α_{prune} for different f and different θ.

We are interested in the test error ε of this final perceptron as a function of α_{tot}, f, and the angle θ between the probe student $\mathbf{J}_{\text{probe}}$ and the teacher $\mathbf{T}$. Our theory approximates $\mathbf{J}_{\text{probe}}$ as simply a random Gaussian vector conditioned to have angle θ with the teacher $\mathbf{T}$. Under this approximation we obtain an analytic theory for $\varepsilon(\alpha_{\text{tot}}, f, \theta)$ that is asymptotically exact in the high dimensional limit (App. A). We first examine results when $\theta = 0$, so we are pruning training examples according to their veridical margins with respect to the teacher (Fig. 1A). We find two striking phenomena, each of which constitute predictions in real-world settings that we will successfully confirm empirically.

The best pruning strategy depends on the amount of initial data. First, we note the test error curve for $f = 1$ in Fig. 1A corresponding to no pruning, or equivalently to *randomly* pruning a larger dataset of size α_{tot} down to a size α_{prune}, exhibits the well known classical perceptron learning power law scaling $\varepsilon \propto \alpha_{\text{prune}}^{-1}$. Interestingly though, for small α_{tot}, keeping the hardest examples performs *worse* than random pruning (lighter curves above darkest curve for small α_{prune} in Fig. 1A). However, for large α_{tot}, keeping the hardest examples performs *substantially better* than random pruning (lighter curves below darkest curve for large α_{prune} in Fig. 1A). It turns out keeping the *easiest* rather than hardest examples is a better pruning strategy when α_{tot} is small (Fig. 1C). Intuitively, if one does not have much data to start with, it is better to keep the easiest examples with largest margins (i.e. the blue regions of Fig. 1B) to avoid overfitting. Otherwise, if one starts with lots of data, so overfitting is less of an issue, it is best to keep the hardest examples with smallest margin that provide more information about the teacher's decision boundary (i.e. the green region of Fig. 1B). Fig. 1C reveals which pruning strategy is best as a joint function of α_{tot} and f. Note the transition between optimal strategies becomes sharper at small fractions f of data kept. This transition between optimal pruning strategies can be viewed as a prediction in more general settings. To test this prediction we trained a ResNet18 on pruned subsets of the CIFAR-10 dataset (Fig. 1D), and observed strikingly similar behavior, indicating the prediction can hold far more generally, beyond perceptron learning. Interestingly, [8, 9] missed this transition, likely because they started pruning from large datasets.

Pareto optimal data pruning can beat power law scaling. A second prediction of our theory is that when keeping a *fixed* fraction f of the hardest examples as α_{tot} increases (i.e. constant color curves in Fig. 1A), the error initially drops exponentially in $\alpha_{\text{prune}} = f\alpha_{\text{tot}}$, but then settles into the universal power law $\varepsilon \propto \alpha_{\text{prune}}^{-1}$ for all fixed f. Thus there is no asymptotic advantage to data pruning at a fixed f. However, by pruning more aggressively (smaller f) when given more initial data (larger α_{tot}), one can achieve a Pareto optimal test error as a function of pruned dataset size α_{prune} that remarkably traces out at least an exponential scaling law (Fig. 1A, purple curve). Indeed our theory predicts for each α_{prune} a Pareto optimal point in α_{tot} and f (subject to $\alpha_{\text{prune}} = f\alpha_{\text{tot}}$), yielding for every fixed α_{prune} an optimal f_{opt}, plotted in Fig. 1E. Note f_{opt} decreases with α_{prune} indicating more aggressive pruning (smaller f_{opt}) of original datasets of larger size α_{tot} is required to obtain larger Pareto optimal pruned datasets of size α_{prune}. We will test this striking scaling prediction in Fig. 3.

Beating power law scaling: an information-theoretic perspective. Classical randomly selected data generates slow power law error scaling because each extra training example provides less new information about the correct decision boundary than the previous example. More formally, let $S(\alpha_{\text{tot}})$ denote the typical entropy of the posterior distribution over student perceptron weights consistent with a training set of size α_{tot}. The information gain $I(\alpha_{\text{tot}})$ due to additional examples beyond α_{tot}

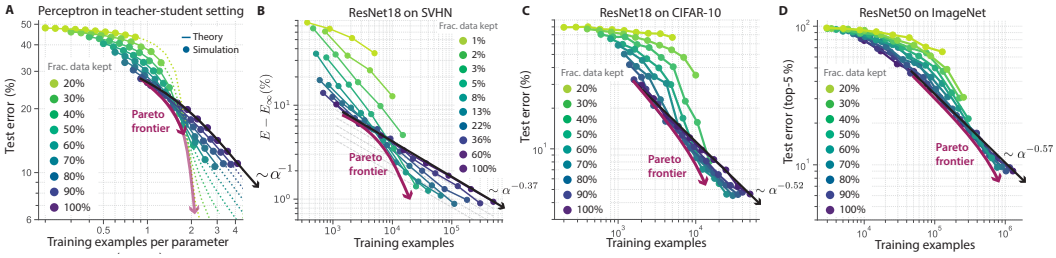


Figure 3: Beating power law scaling in practice. **A–D:** Curves of test error against pruned dataset size in 4 settings. Pruning scores were EL2N [9] for CIFAR-10 and SVHN and memorization [12] for ImageNet. See App. B for all pruning/training details and App. D for similar ImageNet plots with EL2N. Note solid curves reflect performance with a fixed total dataset size; if we prune more aggressively with even larger datasets, scaling could improve further (e.g., dashed lines in **A**). Error curves with no data pruning ($f = 1$) are labeled with their best-fit power law scaling $\sim \alpha^{-\nu}$. (Note that for SVHN in **B** an asymptotic constant error $E(P \to \infty) = 1.1\%$ is subtracted from each of the curves to visualize the power law scaling more clearly.)

can be defined as the rate at which the posterior entropy is reduced: $I(\alpha_{\text{tot}}) = -\frac{d}{d\alpha_{\text{tot}}} S(\alpha_{\text{tot}})$. In classical perceptron learning $I(\alpha_{\text{tot}})$ decays to zero as a power law in α_{tot}, reflecting a vanishing amount of information per each new example, leading to the slow power law decay of test error $\varepsilon \propto \alpha_{\text{tot}}^{-1}$. However, data pruning can increase the information gained per example by pruning away the uninformative examples. To show this, we generalized the replica calculation of the posterior entropy S and information gain I from random datasets of size α_{tot} to pruned datasets of size α_{prune} (App. A). We plot the resulting information gain $I(\alpha_{\text{prune}})$ for different f in Fig. 1F. For any fixed f, $I(\alpha_{\text{prune}})$ will eventually decay as a power law as $\alpha_{\text{prune}}^{-1}$. However, by more aggressively pruning (smaller f) datasets of larger size α_{tot}, $I(\alpha_{\text{prune}})$ can converge to a finite value $I(\infty) = 1$ nat/example, resulting in larger pruned datasets only adding useful non-redundant information. Since each new example under Pareto optimal data pruning conveys finite information about the target decision boundary, as seen in Fig. 1F, the test error can decay at least exponentially in pruned dataset size as in Fig. 1A.

An imperfect pruning metric yields a cross over from exponential to power law scaling. We next examine the case of nonzero angle θ between the probe student $\mathbf{J}_{\text{probe}}$ and the teacher $\mathbf{T}$, such that the ranking of training examples by margin is no longer completely accurate (Fig. 2A). Retaining the hard examples with smallest margin with respect to the probe student will always result in pruned datasets lying near the probe's decision boundary. But if θ is large, such examples might be far from the teacher's decision boundary, and therefore could be less informative about the teacher (Fig. 2A). As a result our theory, confirmed by simulations, predicts that under nonzero angles θ, the Pareto optimal lower envelope of test error over both α_{tot} and f initially scales exponentially as a function of $\alpha_{\text{prune}} = f\alpha_{\text{tot}}$ but then crosses over to a power law (Fig. 2BCD). Indeed, at any given nonzero θ, our theory reveals that as α_{tot} (and therefore α_{prune}) becomes large, one cannot decrease test error any further by retaining less than a minimum fraction $f_{\text{min}}(\theta)$ of all available data. For example when $\theta = 10^\circ$ ($\theta = 20^\circ$) one can do no better asymptotically than pruning down to 24% (46%) of the total data (Fig. 2CD). As θ approaches 0, $f_{\text{min}}(\theta)$ approaches 0, indicating that one can prune extremely aggressively to arbitrarily small f while still improving performance, leading to at least exponential scaling for arbitrarily large α_{prune} in Fig. 2B. However, for nonzero θ, the lack of improvement for $f < f_{\text{min}}(\theta)$ at large α_{prune} renders aggressive pruning ineffective. This result highlights the importance of finding high quality pruning metrics with $\theta \approx 0$. Such metrics can delay the cross over from exponential to power law scaling as pruned dataset size α_{prune} increases, by making aggressive pruning with very small f highly effective. Strikingly, in App. Fig. 10 we demonstrate this cross-over in a real-world setting by showing that the test error on SVHN is bounded below by a power law when the dataset is pruned by a probe ResNet18 under the EL2N metric, trained for 4 epochs (weak pruning metric) but not a probe ResNet18 trained for 40 epochs (strong pruning metric).

4 Data pruning can beat power law scaling in practice

Our theory of data pruning for the perceptron makes three striking predictions which can be tested in more general settings, such as deep neural networks trained on benchmark datasets: (1) relative to random data pruning, keeping only the hardest examples should *help* when the initial dataset size

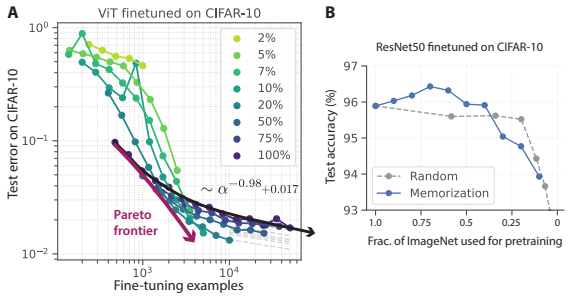


Figure 4: Data pruning improves transfer learning. **A**: CIFAR-10 performance of a ViT pre-trained on all of ImageNet21K and fine-tuned on different pruned subsets of CIFAR-10 under the EL2N metric. **B:** CIFAR-10 performance of ResNet50s pre-trained on different pruned subsets of ImageNet1K and fine-tuned on all of CIFAR-10.

is large, but *hurt* when it is small; (2) data pruning by retaining a fixed fraction f of the hardest examples should yield power law scaling, with exponent equal to that of random pruning, as the initial dataset size increases; (3) the test error optimized over both initial data set size and fraction of data kept can trace out a Pareto optimal lower envelope that beats power law scaling of test error as a function of pruned dataset size, through more aggressive pruning at larger initial dataset size. We verified all three of these predictions on ResNets trained on SVHN, CIFAR-10, and ImageNet using varying amounts of initial dataset size and fractions of data kept under data pruning (compare theory in Fig. 3A with deep learning experiments in Fig. 3BCD). In each experimental setting we see better than power law scaling at larger initial data set sizes and more aggressive pruning. Moreover we would likely see even better scaling with even larger initial datasets (as in Fig.3A dashed lines).

Data pruning improves transfer learning. Modern foundation models are pre-trained on a large initial dataset, and then transferred to other downstream tasks by fine-tuning on them. We therefore examined whether data-pruning can be effective for both reducing the amount of fine-tuning data and the amount of pre-training data. To this end, we first analyzed a vision transformer (ViT) pre-trained on ImageNet21K and then fine-tuned on different pruned subsets of CIFAR-10. Interestingly, pre-trained models allow for far more aggressive data pruning; fine-tuning on only 10% of CIFAR-10 can match or exceed performance obtained by fine tuning on *all* of CIFAR-10 (Fig. 4A). Furthermore Fig. 4A provides a new example of beating power law scaling in the setting of fine-tuning. Additionally, we examined the efficacy of pruning pre-training data by pre-training ResNet50s on different pruned subsets of ImageNet1K (exactly as in Fig. 3D) and then fine-tuning them on all of CIFAR-10. Fig. 4B demonstrates pre-training on as little as 50% of ImageNet can match or exceed CIFAR-10 performance obtained by pre-training on all of ImageNet. Thus intriguingly pruning pre-training data on an upstream task can still maintain high performance on a different downstream task. Overall these results demonstrate the promise of data pruning in transfer learning for both the pre-training and fine-tuning phases.

5 Benchmarking supervised pruning metrics on ImageNet

We note that the majority of data pruning experiments have been performed on small-scale datasets (i.e. variants of MNIST and CIFAR), while the few pruning metrics proposed for ImageNet have rarely been compared against baselines designed on smaller datasets. Therefore, it is currently unclear how most pruning methods scale to ImageNet and which method is best. Motivated by how strongly the quality of a pruning metric can impact performance in theory (Fig. 2), we decided to fill this knowledge gap by performing a systematic evaluation of 8 different supervised pruning metrics on ImageNet: two variants of influence scores [12], two variants of EL2N [9], DDD [20], memorization [12], ensemble active learning [10], and forgetting [8]. See Section 2 for a review of these metrics. Additionally, we include two new prototypicality metrics that we introduce in the next section.

We first asked how consistent the rankings induced by different metrics are by computing the Spearman rank correlation between each pair of metrics (Fig. 5A). Interestingly, we found substantial diversity across metrics, though some (EL2N, DDD, and memorization) were fairly similar with rank correlations above 0.7. However, we observed marked performance differences between metrics: Fig 5BC shows test performance when a fraction f of the hardest examples under each metric are kept in the training set. Despite the success of many of these metrics on smaller datasets, only a few still match performance obtained by training on the full dataset, when selecting a significantly smaller training subset (i.e. about 80% of ImageNet). Nonetheless, most metrics continue to beat random pruning, with memorization in particular demonstrating strong performance (Fig. 5C).

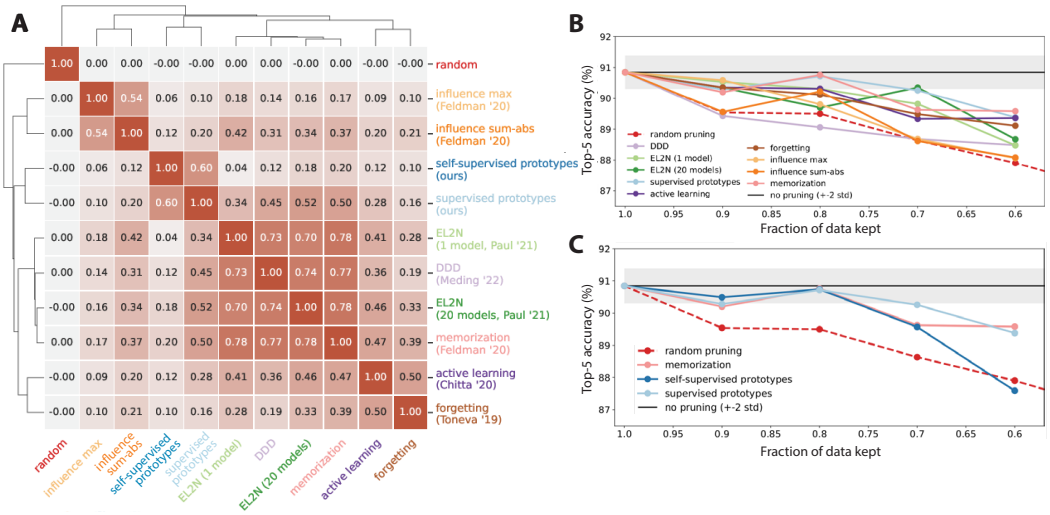


Figure 5: Dataset pruning at ImageNet scale. **A:** Spearman's rank correlation between all pairs of ImageNet metric scores, along with hierarchical clustering (as provided by `seaborn.clustermap`). **B:** Benchmarking supervised metrics on ImageNet (top-5 validation accuracy). **C:** Comparing top-5 performance on ImageNet when pruning according to the best existing supervised metric (memorization) and our supervised and self-supervised prototype metrics. In all 3 cases, training on 80% of ImageNet matches training on 100%. See App. B for pruning and training details.

We found that all pruning metrics amplify class imbalance, which results in degraded performance. To solve this we used a simple 50% class balancing ratio for all ImageNet experiments. Further details and baselines without class balancing are shown in App. H.

6 Self-supervised data pruning through a prototypicality metric

Fig. 5 shows many data pruning metrics do not scale well to ImageNet, while the few that do require substantial amounts of compute. Furthermore, all these metrics require labels, thereby limiting their ability to prune data for large-scale foundation models trained on massive unlabeled datasets [11]. Thus there is a clear need for simple, scalable, self-supervised pruning metrics.

To compute a self-supervised pruning metric for ImageNet, we perform k-means clustering in the embedding space of an ImageNet pre-trained self-supervised model (here: SWaV [31]), and define the difficulty of each data point by the distance to its nearest cluster centroid, or prototype. Thus easy (hard) examples are the most (least) prototypical. Encouragingly, in Fig. 5C, we find our self-supervised prototype metric matches or exceeds the performance of the best supervised metric, memorization, until only 70% of the data is kept, despite the fact that our metric does not use labels and is much simpler and cheaper to compute than many previously proposed supervised metrics. See App. Fig. 9 for further scaling experiments using the self-supervised metric.

To assess whether the clusters found by our metric align with ImageNet classes, we compared their overlaps in Fig. 6A. Interestingly, we found alignment for some but not all classes. For example, class categories such as snakes were largely aligned to a small number of unsupervised clusters, while other classes were dispersed across many such clusters. If class information is available, we can enforce alignment between clusters and classes by simply computing a single prototype for each class (by averaging the embeddings of all examples of this class). While originally intended to be an additional baseline metric (called supervised prototypes, light blue in Fig 5BC), this metric remarkably outperforms all other supervised metrics and largely matches the performance of memorization, which is prohibitively expensive to compute. Moreover, when more than 70% of the data are kept, the performance of the self-supervised and supervised metrics are similar, demonstrating the promise of self-supervised pruning.

One important choice for the self-supervised prototype metric is the number of clusters k. We found, reassuringly, our results were robust to this choice: k can deviate one order of magnitude more or less than the true number of classes (i.e. 1000 for ImageNet) without affecting performance (App. F).

To better understand example difficulty under various metrics, we visualize extremal images for our self-supervised prototype metric and the memorization metric for one class (Fig 6B,C). Qualitatively, easy examples correspond to highly similar, redundant images, while hard examples look like idiosyncratic outliers. See App. E, Figs. 11,12,13,14,15,16,17,18 for more classes and metrics.

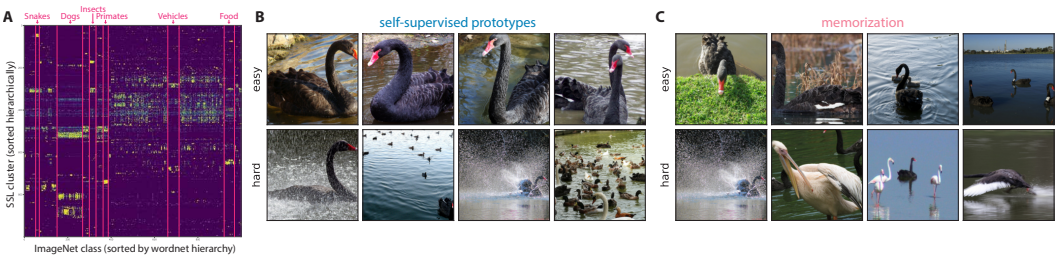


Figure 6: **A:** Heat map where each row denotes the probability that images in a given cluster come from each ImageNet class. **B:** The four easiest and hardest images under our self-supervised pruning metric and the best previously published supervised metric (memorization, shown in **C**) for ImageNet class 100 (`black swan`).

7 Discussion

Summary. We have shown, both in theory and practice, how to break beyond slow power law scaling of error versus dataset size to faster exponential scaling, through data pruning. Additionally we have developed a simple self-supervised pruning metric that enables us to discard 20% of ImageNet without sacrificing performance, on par with the best and most compute intensive supervised metric.

Limitations. The most notable limitation is that achieving exponential scaling requires a high quality data pruning metric. Since most metrics developed for smaller datasets scale poorly to ImageNet, our results emphasize the importance of future work in identifying high quality, scalable metrics. Our self-supervised metric provides a strong initial baseline. Moreover, a key advantage of data pruning is reduced computational cost due to training on a smaller dataset for the same number of epochs as the full dataset (provided the computational cost of computing the pruning metric does not dominate). However, we found that performance often increased when the same number training *iterations* on the full dataset was carried out on the pruned dataset, resulting in the same training time, but additional training epochs. However, this performance gain saturated *before* training time on the pruned dataset approached that on the whole dataset (App. J) thereby still yielding a computational efficiency gain. Overall this tradeoff between accuracy and training time on pruned data is important to consider in evaluating potential gains due to data pruning. Finally, we found that class-balancing was essential to maintain performance on data subsets (App. H). Future work will be required to identify ways to effectively select the appropriate amount of class-balancing.

Ethical considerations. A potential negative societal impact could be that data-pruning leads to unfair outcomes for certain groups. We have done a preliminary analysis of how data-pruning affects performance on individual ImageNet classes (App. I), finding no substantial differential effects across classes. However proper fairness tests specific to deployment settings should always be conducted on every model, whether trained on pruned data or not. Additionally, we analyzed the impact of pruning on OOD performance (App. K).

Outlook: Towards foundation datasets. We believe the most promising future direction is the further development of scalable, unsupervised data pruning metrics. Indeed our theory predicts that the application of pruning metrics on larger scale datasets should yield larger gains by allowing more aggressive pruning. This makes data pruning especially exciting for use on the massive unlabeled datasets used to train large foundation models (e.g. 400M image-text pairs for CLIP [32], 3.5B Instagram images [33], 650M images for the DALLE-2 encoder [34], 780B tokens for PALM [35]). If highly pruned versions of these datasets can be used to train a large number of different models, one can conceive of such carefully chosen data subsets as *foundation datasets* in which the initial computational cost of data pruning can be amortized across efficiency gains in training many downstream models, just at the initial computational cost of training foundation models is amortized across the efficiency gains of fine-tuning across many downstream tasks. Together, our results demonstrate the promise and potential of data pruning for large-scale training and pretraining.

References

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

[2] 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.

[3] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

[4] Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[5] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

[6] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021.

[7] 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.

[8] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

[9] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Adv. Neural Inf. Process. Syst.*, 34, December 2021.

[10] Kashyap Chitta, José M Álvarez, Elmar Haussmann, and Clément Farabet. Training data subset search with ensemble active learning. *IEEE Trans. Intell. Transp. Syst.*, pages 1–12, 2021.

[11] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[12] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Adv. Neural Inf. Process. Syst.*, 33:2881–2891, 2020.

[13] Burr Settles. Active learning literature survey. *Technical Report*, 2009.

[14] Antoine Bordes, Seyda Ertekin, Jason Weston, Léon Botton, and Nello Cristianini. Fast kernel classifiers with online and active learning. *J. Mach. Learn. Res.*, 6(9), 2005.

[15] Zeyad Ali Sami Emam, Hong-Min Chu, Ping-Yeh Chiang, Wojciech Czaja, Richard Leapman, Micah Goldblum, and Tom Goldstein. Active learning at the ImageNet scale. *arXiv preprint arXiv:2111.12880*, 2021.

[16] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[17] Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D Manning. Mind your outliers! Investigating the negative impact of outliers on active learning for visual question answering. *arXiv preprint arXiv:2107.02331*, 2021.

[18] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In Hal Daumé Iii and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6950–6960. PMLR, 2020.

[19] V Birodkar, H Mobahi, and S Bengio. Semantic redundancies in Image-Classification datasets: The 10% you don't need. *arXiv preprint arXiv:1901.11409*, 2019.

[20] Kristof Meding, Luca M. Schulze Buschoff, Robert Geirhos, and Felix A. Wichmann. Trivial or impossible—dichotomous data difficulty masks model differences (on ImageNet and beyond). In *International Conference on Learning Representations*, 2022.

[21] A Engel and C V den Broeck. *Statistical Mechanics of Learning*. Cambridge Univ. Press, 2001.

[22] Madhu Advani, Subhaneil Lahiri, and Surya Ganguli. Statistical mechanics of complex neural systems and high dimensional data. *J. Stat. Mech: Theory Exp.*, 2013(03):P03014, 2013.

[23] Yasaman Bahri, Jonathan Kadmon, Jeffrey Pennington, Sam S Schoenholz, Jascha Sohl-Dickstein, and Surya Ganguli. Statistical mechanics of deep learning. *Annual Review of Condensed Matter Physics*, March 2020.

[24] Lenka Zdeborová and Florent Krzakala. Statistical physics of inference: thresholds and algorithms. *Adv. Phys.*, 65(5):453–552, September 2016.

[25] E Gardner. The space of interactions in neural network models. *J. of Physics A*, 21:257–270, 1988.

[26] H S Seung, H Sompolinsky, and N Tishby. Statistical mechanics of learning from examples. *Phys. Rev. A*, 45(8):6056, 1992.

[27] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Information, prediction, and query by committee. *Adv. Neural Inf. Process. Syst.*, 5, 1992.

[28] Hai-Jun Zhou. Active online learning in the binary perceptron problem. *Commun. Theor. Phys.*, 71(2):243, February 2019.

[29] Hugo Cui, Luca Saglietti, and Lenka Zdeborovà. Large deviations in the perceptron model and consequences for active learning, 2021.

[30] M Mezard, G Parisi, and M A Virasoro. *Spin glass theory and beyond*. World scientific Singapore, 1987.

[31] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020.

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

[33] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[35] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[36] Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Ishan Misra. VISSL. `https://github.com/facebookresearch/vissl`, 2021.

[37] 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(3):211–252, 2015.

[38] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the ImageNet hierarchy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 547–558, 2020.

[39] Yuki M Asano, Christian Rupprecht, Andrew Zisserman, and Andrea Vedaldi. PASS: An ImageNet replacement for self-supervised pretraining without humans. *arXiv preprint arXiv:2109.13228*, 2021.

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

[41] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[42] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. *Advances in Neural Information Processing Systems*, 34:23885–23899, 2021.

[43] Felix A Wichmann, David HJ Janssen, Robert Geirhos, Guillermo Aguilar, Heiko H Schütt, Marianne Maertens, and Matthias Bethge. Methods and measurements to compare men against machines. *Electronic Imaging, Human Vision and Electronic Imaging*, 2017(14):36–45, 2017.

[44] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, 2018.

[45] 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 *International Conference on Learning Representations*, 2019.

[46] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019.

[47] Robert Geirhos, Kristof Meding, and Felix A Wichmann. Beyond accuracy: quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency. *Advances in Neural Information Processing Systems*, 33, 2020.

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[49] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See discussion section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See discussion section 7 and App. I.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See App. A
   (b) Did you include complete proofs of all theoretical results? [Yes] See App. A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] All code for theory plots and numerical perceptron simulations is packaged in a reproducible colab notebook.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See App. B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Since we train a very large (order 100) number of models on computationally intensive tasks (e.g. ImageNet), many of our plots contain only results from a single random seed. However, variability due to random seeds can be inferred by the smooth progression of the relevant quantity on each plot.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See App. B
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See App. B
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See App. B
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See App. B
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

Table of Contents

A A theory of data-pruning for the perceptron: detailed derivations 14

A.1 Problem setup . . . 14
A.2 Main result and overview . . . 15
A.3 Replica calculation of the generalization error . . . 15
A.4 Quenched entropy . . . 24
A.5 Perfect teacher-probe overlap . . . 24
A.6 Information gain per example . . . 25
A.7 Imperfect teacher-probe overlap . . . 26
A.8 Optimal pruning policy . . . 27
A.9 Exact saddle point equations . . . 28

B Model training method details & dataset information 31

C Breaking compute scaling laws via data pruning 32

D Additional scaling experiments 33

E Extremal images according to different metrics 34

F Impact of number of clusters k on self-supervised prototypes 43

G Impact of ensemble prototypes 43

H Relationship between pruning and class (im-)balance 43

I Effect of pruning on class-conditional accuracy and fairness 49

J Interaction between data pruning and training duration 49

K Out-of-distribution (OOD) analysis of dataset pruning 50

A A theory of data-pruning for the perceptron: detailed derivations

All code required to reproduce the theory figures and numerical simulations throughout this paper can be run in the Colab notebook at `https://colab.research.google.com/drive/1in35C6jh7y_ynwuWLBmGOWAgmUgpl8dF?usp=sharing`.

A.1 Problem setup

In this section we introduce a theory for data pruning in the teacher-student perceptron setting, using the tools of statistical mechanics. We study the problem of classifying a dataset of P examples $\{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\dots,P}$, where $\mathbf{x}^\mu \sim \mathcal{N}(0, I_N)$ are i.i.d. zero mean unit variance random Gaussian inputs, and $y^\mu = \text{sign}(\mathbf{T} \cdot x)$ are labels generated by a teacher perceptron $\mathbf{T} \in \mathbb{R}^N$, which we will assume is randomly drawn from a uniform distribution on the sphere $\mathbf{T} \sim \text{Unif}(\mathbb{S}^{N-1}(\sqrt{N}))$. We work in the high-dimensional statistics limit where $N, P \to \infty$ but the ratio $\alpha_{\text{tot}} = P/N$ remains $O(1)$. The generalization error of a perceptron trained on such an isotropic dataset is a classical problem (see e.g. [21]). However, we are interested in the setting where the training dataset is not isotropic, but instead has inherited some structure due to data pruning.

In particular, consider pruning the training dataset by keeping only the examples with the smallest margin $|z^\mu| = |\mathbf{J}_{\text{probe}} \cdot \mathbf{x}^\mu|$ along a probe student $\mathbf{J}_{\text{probe}}$. The pruned dataset will follow some distribution $p(z)$ along the direction of $\mathbf{J}_{\text{probe}}$, and remain isotropic in the nullspace of $\mathbf{J}_{\text{probe}}$. In what follows we will derive a general theory for an arbitrary data distribution $p(z)$, and specialize to the case of small-margin pruning only at the very end (in which case $p(z)$ will take the form of a truncated Gaussian). We will also make no assumptions on the form of the probe student $\mathbf{J}_{\text{probe}}$ or the learning rule used to train it; only that $\mathbf{J}_{\text{probe}}$ has developed some overlap with the teacher, quantified by the angle $\theta = \cos^{-1}\left(\frac{\mathbf{J}_{\text{probe}} \cdot \mathbf{T}}{\|\mathbf{J}_{\text{probe}}\|_2 \|\mathbf{T}\|_2}\right)$ (Fig. 2**A**).

After the dataset has been pruned, we consider training a new student J from scratch on the pruned dataset. A typical training algorithm (used in support vector machines and the solution to which SGD converges on separable data) is to find the solution J which classifies the training data with the maximal margin $\kappa = \min_\mu \mathbf{J} \cdot (y^\mu \mathbf{x}^\mu)$. Our goal is to compute the generalization error ε_g of this student, which is simply governed by the overlap between the student and the teacher, $\varepsilon_g = \cos^{-1}(R)/\pi$, where $R = \mathbf{J} \cdot \mathbf{T} / \|\mathbf{J}\|_2 \|\mathbf{T}\|_2$.

A.2 Main result and overview

Our main result is a set of self-consistent equations which can be solved to obtain the generalization error $\varepsilon(\alpha, p, \theta)$ for any α and any data distribution $p(z)$ along a probe student at any angle θ relative to the teacher. These equations take the form,

$$\frac{R - \rho\cos\theta}{\sin^2\theta} = \frac{\alpha}{\pi\Lambda}\left\langle \int_{-\infty}^{\kappa} dt \, \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)(\kappa - t)\right\rangle_z \tag{1}$$

$$1 - \frac{\rho^2 + R^2 - 2\rho R\cos\theta}{\sin^2\theta} = 2\alpha\left\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)(\kappa - t)^2\right\rangle_z \tag{2}$$

$$\begin{aligned}\frac{\rho - R\cos\theta}{\sin^2\theta} = 2\alpha\Bigg\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)\left(\frac{z-\rho t}{1-\rho^2}\right)(\kappa - t) \\ + \frac{1}{2\pi\Lambda}\exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)\left(\frac{\rho R - \cos\theta}{1-\rho^2}\right)(\kappa - t)\Bigg\rangle_z\end{aligned} \tag{3}$$

Where,

$$\Lambda = \sqrt{\sin^2\theta - R^2 - \rho^2 + 2\rho R\cos\theta}, \tag{4}$$

$$\Gamma(t,z) = z(\rho R - \cos\theta) - t(R - \rho\cos\theta), \tag{5}$$

$$\Delta(t,z) = z^2\left(\rho^2 + \cos^2\theta - 2\rho R\cos\theta\right) + 2tz(R\cos\theta - \rho) + t^2\sin^2\theta. \tag{6}$$

Where $\langle\cdot\rangle_z$ represents an average over the pruned data distribution $p(z)$ along the probe student. For any $\alpha, p(z), \theta$, these equations can be solved for the order parameters R, ρ, κ, from which the generalization error can be easily read off as $\varepsilon_g = \cos^{-1}(R)/\pi$. This calculation results in the solid theory curves in Figs 1,2,3, which show an excellent match to numerical simulations. In the following section we will walk through the derivation of these equations using replica theory. In Section A.6 we will derive an expression for the information gained per training example, and show that with Pareto optimal data pruning this information gain can be made to converge to a finite rate, resulting in at least exponential decay in test error. In Section A.7, we will show that super-exponential scaling eventually breaks down when the probe student does not match the teacher perfectly, resulting in power law scaling at at a minimum pruning fraction $f_{\text{min}}(\theta)$.

A.3 Replica calculation of the generalization error

To obtain Eqs. 1,2,3, we follow the approach of Elizabeth Gardner and compute the volume $\Omega(\mathbf{x}^\mu, \mathbf{T}, \kappa)$ of solutions J which perfectly classify the training data up to a margin κ (known as the Gardner volume) [25, 21]. As κ grows, the volume of solutions shrinks until it reaches a unique solution at a critical κ, the max-margin solution. The Gardner volume Ω takes the form,

$$\Omega(\mathbf{x}^\mu, \mathbf{T}, \kappa) = \int d\mu(\mathbf{J}) \prod_\mu \Theta\left(\frac{\mathbf{T}\cdot\mathbf{x}^\mu}{\sqrt{N}}\left(\frac{\mathbf{J}\cdot\mathbf{x}^\mu}{\sqrt{N}} - \kappa\right)\right) \tag{7}$$

Because the student's decision boundary is invariant to an overall scaling of $\mathbf{J}$, we enforce normalization of $\mathbf{J}$ via the measure $d\mu(\mathbf{J})$,

$$d\mu(\mathbf{J}) = \frac{1}{(2\pi e)^{N/2}}\delta(\|\mathbf{J}\|^2 - N) \tag{8}$$

In the thermodynamic limit $N, P \to \infty$ the typical value of the entropy $S(\kappa) = \langle\langle \log \Omega(\mathbf{x}^\mu, \mathbf{T}, \kappa)\rangle\rangle$ is dominated by particular values of R, κ, where the double angle brackets $\langle\langle\cdot\rangle\rangle$ denote a quenched average over disorder introduced by random realizations of the training examples $\mathbf{x}^\mu$ and the teacher $\mathbf{T}$. However, computing this quenched average is intractable since the integral over $\mathbf{J}$ cannot be performed analytically for every individual realization of the examples. We rely on the replica trick from statistical physics,

$$\ln(x) = \lim_{n\to 0} \frac{x^n - 1}{n} \tag{9}$$

Which allows us to evaluate $S(\kappa)$ in terms of easier-to-compute powers of Ω,

$$S(\kappa) = \langle\langle \ln \Omega(\mathbf{x}^\mu, \mathbf{T}, \kappa)\rangle\rangle = \frac{\langle\langle \Omega^n(\mathbf{x}^\mu, \mathbf{T}, \kappa)\rangle\rangle - 1}{n} \tag{10}$$

This reduces our problem to computing powers of Ω, which for integer n can be written in terms of $\alpha = 1, \ldots, n$ replicated copies of the original system,

$$\Omega^{(n)} \equiv \langle\langle \Omega^n(\mathbf{x}^\mu, \mathbf{T}, \kappa)\rangle\rangle = \left\langle\left\langle \int \prod_{\alpha=1}^n d\mu(\mathbf{J}^\alpha) \prod_{\alpha,\mu} \Theta\left(\frac{\mathbf{T}\cdot\mathbf{x}^\mu}{\sqrt{N}}\left(\frac{\mathbf{J}\cdot\mathbf{x}^\mu}{\sqrt{N}} - \kappa\right)\right)\right\rangle\right\rangle \tag{11}$$

We begin by introducing auxiliary variables,

$$\lambda_\mu^\alpha = \frac{\mathbf{J}^\alpha\cdot\mathbf{x}^\mu}{\sqrt{N}}, \quad u_\mu = \frac{\mathbf{T}\cdot\mathbf{x}^\mu}{\sqrt{N}} \tag{12}$$

by $\delta-$functions, to pull the dependence on $\mathbf{J}$ and $\mathbf{T}$ outside of the Heaviside function,

$$\begin{aligned}\Omega^{(n)} = \int \prod_{\alpha=1}^n d\mu(\mathbf{J}^\alpha) \int \prod_{\alpha,\mu} d\lambda_\mu^\alpha \int \prod_\mu du_\mu \prod_{\alpha,\mu} \Theta\big(u_\mu(\lambda_\mu^\alpha - \kappa)\big) \\ \times \left\langle\left\langle \delta\left(\lambda_\mu^\alpha - \frac{1}{\sqrt{N}}\mathbf{J}^\alpha\cdot\mathbf{x}^\mu\right)\delta\left(u_\mu - \frac{1}{\sqrt{N}}\mathbf{T}\cdot\mathbf{x}^\mu\right)\right\rangle\right\rangle\end{aligned} \tag{13}$$

Using the integral representation of the δ-functions,

$$\Omega^{(n)} = \int \prod_{\alpha=1}^n d\mu(\mathbf{J}^\alpha) \int \prod_{\alpha,\mu} \frac{d\lambda_\mu^\alpha d\hat{\lambda}_\mu^\alpha}{2\pi} \int \prod_\mu \frac{du_\mu d\hat{u}_\mu}{2\pi} \tag{14}$$

$$\times \prod_{\alpha,\mu} \Theta\big(u_\mu(\lambda_\mu^\alpha - \kappa)\big) \exp\left(i\sum_{\mu,\alpha} \lambda_\mu^\alpha \hat{\lambda}_\mu^\alpha + i\sum_\mu u_\mu \hat{u}_\mu\right) \tag{15}$$

$$\times \langle\langle \exp\left(-\frac{i}{\sqrt{N}}\sum_{\mu,\alpha} \hat{\lambda}_\mu^\alpha \mathbf{J}^\alpha\cdot\mathbf{x}^\mu - \frac{i}{\sqrt{N}}\sum_\mu \hat{u}_\mu \mathbf{T}\cdot\mathbf{x}^\mu\right)\rangle\rangle \tag{16}$$

The data obeys some distribution $p(z)$ along the direction of $\mathbf{J}_{\text{probe}}$ and is isotropic in the nullspace of $\mathbf{J}_{\text{probe}}$. Hence we can decompose a training example $\mathbf{x}^\mu$ as follows, $\mathbf{x}^\mu = \mathbf{J}_{\text{probe}} z^\mu + (I - \mathbf{J}_{\text{probe}}\mathbf{J}_{\text{probe}}^T)\mathbf{s}^\mu$, where $z^\mu \sim p(z)$ and $\mathbf{s}^\mu \sim \mathcal{N}(0, I_N)$,

$$\Omega^{(n)} = \int \prod_{\alpha=1}^{n} d\mu(\mathbf{J}^\alpha) \int \prod_{\alpha,\mu} \frac{d\lambda_\mu^\alpha d\hat{\lambda}_\mu^\alpha}{2\pi} \int \prod_{\mu} \frac{du_\mu d\hat{u}_\mu}{2\pi} \tag{17}$$

$$\times \prod_{\alpha,\mu} \Theta\big(u_\mu(\lambda_\mu^\alpha - \kappa)\big) \exp\left(i \sum_{\mu,\alpha} \lambda_\mu^\alpha \hat{\lambda}_\mu^\alpha + i \sum_{\mu} u_\mu \hat{u}_\mu \right) \tag{18}$$

$$\times \left\langle\left\langle \exp\left(-\frac{i}{\sqrt{N}} \sum_{\mu,\alpha} \hat{\lambda}_\mu^\alpha (\mathbf{J}^\alpha \cdot \mathbf{J}_{\text{probe}} z^\mu + \mathbf{J}_\perp^\alpha \cdot \mathbf{s}^\mu) - \frac{i}{\sqrt{N}} \sum_{\mu} \hat{u}_\mu (\mathbf{T} \cdot \mathbf{J}_{\text{probe}} z^\mu + \mathbf{T}_\perp \cdot \mathbf{s}^\mu) \right) \right\rangle\right\rangle \tag{19}$$

Where $\mathbf{J}_\perp = (1 - \mathbf{J}_{\text{probe}}\mathbf{J}_{\text{probe}}^T)\mathbf{J}$ and $\mathbf{T}_\perp = (1 - \mathbf{J}_{\text{probe}}\mathbf{J}_{\text{probe}}^T)\mathbf{T}$. Now we can average over the patterns $\mathbf{s}^\mu \sim \mathcal{N}(0, I_N)$,

$$\left\langle \exp\left(-\frac{i}{\sqrt{N}} \sum_{\mu,\alpha} \hat{\lambda}_\mu^\alpha \mathbf{J}_\perp^\alpha \cdot \mathbf{s}^\mu - \frac{i}{\sqrt{N}} \sum_{\mu} \hat{u}_\mu \mathbf{T}_\perp \cdot \mathbf{s}^\mu \right) \right\rangle_{s^\mu} = \exp\left(-\frac{1}{2N} \| \sum_{\mu,\alpha} \hat{\lambda}_\mu^\alpha \mathbf{J}_\perp^\alpha + \hat{u}_\mu \mathbf{T}_\perp \|^2 \right) \tag{20}$$

$$= \exp\left(-\frac{1}{2N} \sum_{\mu} \left(\sum_{\alpha\beta} \hat{\lambda}_\mu^\alpha \hat{\lambda}_\mu^\beta \mathbf{J}_\perp^\alpha \cdot \mathbf{J}_\perp^\beta + 2 \sum_{\alpha} \hat{\lambda}_\mu^\alpha \hat{u}_\mu \mathbf{J}_\perp^\alpha \cdot \mathbf{T}_\perp + \hat{u}_\mu^2 \|\mathbf{T}_\perp\|^2 \right) \right). \tag{21}$$

Inserting this back into our expression for the Gardner volume,

$$\begin{aligned}
\Omega^{(n)} = &\int \prod_{\alpha=1}^{n} d\mu(\mathbf{J}^\alpha) \int \prod_{\alpha,\mu} \frac{d\lambda_\mu^\alpha d\hat{\lambda}_\mu^\alpha}{2\pi} \int \prod_{\mu} \frac{du_\mu d\hat{u}_\mu}{2\pi} \\
&\times \prod_{\alpha,\mu} \Theta\big(u_\mu(\lambda_\mu^\alpha - \kappa)\big) \exp\left(i \sum_{\mu,\alpha} \lambda_\mu^\alpha \hat{\lambda}_\mu^\alpha + i \sum_{\mu} u_\mu \hat{u}_\mu \right) \\
&\times \left\langle\left\langle \exp\left[-\frac{1}{2N} \sum_{\mu} \left(\sum_{\alpha\beta} \hat{\lambda}_\mu^\alpha \hat{\lambda}_\mu^\beta \mathbf{J}_\perp^\alpha \cdot \mathbf{J}_\perp^\beta + 2 \sum_{\alpha} \hat{\lambda}_\mu^\alpha \hat{u}_\mu \mathbf{J}_\perp^\alpha \cdot \mathbf{T}_\perp + \hat{u}_\mu^2 \|\mathbf{T}_\perp\|^2 \right) \right.\right. \\
&\left.\left. - i \sum_{\mu} \left(\sum_{\alpha} \hat{\lambda}_\mu^\alpha \mathbf{J}^\alpha \cdot \mathbf{J}_{\text{probe}} + \hat{u}_\mu \mathbf{T} \cdot \mathbf{J}_{\text{probe}} \right) z^\mu \right] \right\rangle\right\rangle_{T, z^\mu}
\end{aligned} \tag{22}$$

As is typical in replica calculations of this type, we now introduce order parameters,

$$q^{\alpha\beta} = \frac{\mathbf{J}^\alpha \cdot \mathbf{J}^\beta}{N}, \quad R^\alpha = \frac{\mathbf{T} \cdot \mathbf{J}^\alpha}{N} \tag{23}$$

which will allow us to decouple the $\mathbf{J}$- from the λ-μ-z- integrals. $q^{\alpha\beta}$ represents the overlaps between replicated students, and R^α the overlap between each replicated student and the teacher. However, because our problem involves the additional role of the probe student, we must introduce an additional order parameter,

$$\rho^\alpha = \frac{\mathbf{J}^\alpha \cdot \mathbf{J}_{\text{probe}}}{N} \tag{24}$$

which represents the overlap between each replicated student and the probe student. Notice that,

$$\mathbf{J}_\perp^\alpha \cdot \mathbf{J}_\perp^\beta = \mathbf{J}^\alpha \cdot \mathbf{J}^\beta - \mathbf{J}_\parallel^\alpha \cdot \mathbf{J}_\parallel^\beta = N(q^{\alpha\beta} - \rho^\alpha \rho^\beta) \tag{25}$$

$$\mathbf{J}_\perp^\alpha \cdot \mathbf{T}_\perp = \mathbf{J}^\alpha \cdot \mathbf{T} - \mathbf{J}_\parallel^\alpha \cdot \mathbf{T}_\parallel = N(R^\alpha - \rho^\alpha \cos\theta) \tag{26}$$

With this new set of order parameters in hand, we can decouple the **J** from the $\lambda - u - z-$integrals.

$$\begin{aligned}
\Omega^{(n)} = &\int \prod_{\alpha<\beta} dq^{\alpha\beta} \int \prod_\alpha dR^\alpha \int \prod_\alpha d\rho^\alpha \\
&\times \int \prod_{\alpha=1}^n d\mu(\mathbf{J}^\alpha) \Big\langle \prod_\alpha \delta(\mathbf{T}\cdot\mathbf{J}^\alpha - NR^\alpha) \Big\rangle_{\mathbf{T}} \prod_{\alpha<\beta} \delta(\mathbf{J}^\alpha \cdot \mathbf{J}^\beta - Nq^{\alpha\beta}) \prod_\alpha \delta(\mathbf{J}^\alpha \cdot \mathbf{J}_{\text{probe}} - N\rho^\alpha) \\
&\times \int \prod_{\alpha,\mu} \frac{d\lambda_\mu^\alpha d\hat{\lambda}_\mu^\alpha}{2\pi} \int \prod_\mu \frac{du_\mu d\hat{u}_\mu}{2\pi} \prod_{\alpha,\mu} \Theta\big(u_\mu(\lambda_\mu^\alpha - \kappa)\big) \prod_\mu \exp\Big(i\sum_\alpha \lambda_\mu^\alpha \hat{\lambda}_\mu^\alpha + iu_\mu \hat{u}_\mu\Big) \\
&\times \Big\langle \exp\Big[-\frac{1}{2}\sum_{\alpha\beta} \hat{\lambda}_\mu^\alpha \hat{\lambda}_\mu^\beta (q^{\alpha\beta} - \rho^\alpha\rho^\beta) - \sum_\alpha \hat{\lambda}_\mu^\alpha \hat{u}_\mu (R^\alpha - \rho^\alpha\cos\theta) - \frac{1}{2}\hat{u}_\mu^2 \sin^2\theta \\
&- i\Big(\sum_\alpha \hat{\lambda}_\mu^\alpha \rho^\alpha + \hat{u}_\mu \cos\theta\Big) z^\mu \Big]\Big\rangle_{z^\mu}
\end{aligned} \tag{27}$$

We can now perform the gaussian integral over $\hat{u}_\mu$,

$$\begin{aligned}
\Omega^{(n)} = &\int \prod_{\alpha<\beta} dq^{\alpha\beta} \int \prod_\alpha dR^\alpha \int \prod_\alpha d\rho^\alpha \\
&\times \int \prod_{\alpha=1}^n d\mu(\mathbf{J}^\alpha) \Big\langle \prod_\alpha \delta(\mathbf{T}\cdot\mathbf{J}^\alpha - NR^\alpha) \Big\rangle_{\mathbf{T}} \prod_{\alpha<\beta} \delta(\mathbf{J}^\alpha \cdot \mathbf{J}^\beta - Nq^{\alpha\beta}) \prod_\alpha \delta(\mathbf{J}^\alpha \cdot \mathbf{J}_{\text{probe}} - N\rho^\alpha) \\
&\times \int \prod_{\alpha,\mu} \frac{d\lambda_\mu^\alpha d\hat{\lambda}_\mu^\alpha}{2\pi} \int \prod_\mu \frac{du_\mu d\hat{u}_\mu}{2\pi} \prod_{\alpha,\mu} \Theta\big(u_\mu(\lambda_\mu^\alpha - \kappa)\big) \prod_\mu \exp\Big(i\sum_\alpha \lambda_\mu^\alpha \hat{\lambda}_\mu^\alpha\Big) \\
&\times \Big\langle \exp\Big[-\frac{1}{2}\sum_{\alpha\beta} \hat{\lambda}_\mu^\alpha \hat{\lambda}_\mu^\beta (q^{\alpha\beta} - \rho^\alpha\rho^\beta) - i\sum_\alpha \hat{\lambda}_\mu^\alpha \rho^\alpha z^\mu \\
&\qquad + \frac{1}{2\sin^2\theta}\Big(i(u_\mu - z^\mu\cos\theta) - \sum_\alpha \hat{\lambda}_\mu^\alpha (R^\alpha - \rho^\alpha\cos\theta)\Big)^2 \Big]\Big\rangle_{z^\mu}
\end{aligned} \tag{28}$$

Expanding,

$$
\begin{aligned}
\Omega^{(n)} = &\int \prod_{\alpha<\beta} dq^{\alpha\beta} \int \prod_{\alpha} dR^{\alpha} \int \prod_{\alpha} d\rho^{\alpha} \\
&\times \int \prod_{\alpha=1}^{n} d\mu(\mathbf{J}^{\alpha}) \Big\langle \prod_{\alpha} \delta(\mathbf{T}\cdot\mathbf{J}^{\alpha} - NR^{\alpha}) \Big\rangle_{\mathbf{T}} \prod_{\alpha<\beta} \delta(\mathbf{J}^{\alpha}\cdot\mathbf{J}^{\beta} - Nq^{\alpha\beta}) \prod_{\alpha} \delta(\mathbf{J}^{\alpha}\cdot\mathbf{J}_{\text{probe}} - N\rho^{\alpha}) \\
&\times \int \prod_{\alpha,\mu} \frac{d\lambda_{\mu}^{\alpha} d\hat{\lambda}_{\mu}^{\alpha}}{2\pi} \int \prod_{\mu} \frac{du_{\mu} d\hat{u}_{\mu}}{2\pi} \prod_{\alpha,\mu} \Theta\big(u_{\mu}(\lambda_{\mu}^{\alpha} - \kappa)\big) \prod_{\mu} \exp\left(i \sum_{\alpha} \lambda_{\mu}^{\alpha} \hat{\lambda}_{\mu}^{\alpha} - \frac{1}{2} \frac{(u_{\mu} - z^{\mu}\cos\theta)^2}{\sin^2\theta} \right) \\
&\times \Big\langle \exp\Big[-\frac{1}{2} \sum_{\alpha\beta} \hat{\lambda}_{\mu}^{\alpha} \hat{\lambda}_{\mu}^{\beta} (q^{\alpha\beta} - \rho^{\alpha}\rho^{\beta}) - i \sum_{\alpha} \hat{\lambda}_{\mu}^{\alpha} \rho^{\alpha} z^{\mu} \\
&- \frac{i}{\sin^2\theta} \big(u_{\mu} - z^{\mu}\cos\theta\big) \sum_{\alpha} \hat{\lambda}_{\mu}^{\alpha} (R^{\alpha} - \rho^{\alpha}\cos\theta) + \frac{1}{2\sin^2\theta} \sum_{\alpha\beta} \hat{\lambda}_{\mu}^{\alpha} \hat{\lambda}_{\mu}^{\beta} (R^{\alpha} - \rho^{\alpha}\cos\theta)(R^{\beta} - \rho^{\beta}\cos\theta) \Big] \Big\rangle_{z^{\mu}}
\end{aligned}
\tag{29}
$$

Simplifying,

$$
\begin{aligned}
\Omega^{(n)} = &\int \prod_{\alpha<\beta} dq^{\alpha\beta} \int \prod_{\alpha} dR^{\alpha} \int \prod_{\alpha} d\rho^{\alpha} \\
&\times \int \prod_{\alpha=1}^{n} d\mu(\mathbf{J}^{\alpha}) \Big\langle \prod_{\alpha} \delta(\mathbf{T}\cdot\mathbf{J}^{\alpha} - NR^{\alpha}) \Big\rangle_{\mathbf{T}} \prod_{\alpha<\beta} \delta(\mathbf{J}^{\alpha}\cdot\mathbf{J}^{\beta} - Nq^{\alpha\beta}) \prod_{\alpha} \delta(\mathbf{J}^{\alpha}\cdot\mathbf{J}_{\text{probe}} - N\rho^{\alpha}) \\
&\times \int \prod_{\alpha,\mu} \frac{d\lambda_{\mu}^{\alpha} d\hat{\lambda}_{\mu}^{\alpha}}{2\pi} \int \prod_{\mu} \frac{du_{\mu} d\hat{u}_{\mu}}{2\pi} \prod_{\alpha,\mu} \Theta\big(u_{\mu}(\lambda_{\mu}^{\alpha} - \kappa)\big) \exp\left(i \sum_{\mu,\alpha} \lambda_{\mu}^{\alpha} \hat{\lambda}_{\mu}^{\alpha} - \frac{1}{2} \frac{(u_{\mu} - z^{\mu}\cos\theta)^2}{\sin^2\theta} \right) \\
&\times \Big\langle \exp\Big[-\frac{1}{2} \sum_{\mu} \sum_{\alpha\beta} \hat{\lambda}_{\mu}^{\alpha} \hat{\lambda}_{\mu}^{\beta} \Big(q^{\alpha\beta} - \rho^{\alpha}\rho^{\beta} - \frac{(R^{\alpha} - \rho^{\alpha}\cos\theta)(R^{\beta} - \rho^{\beta}\cos\theta)}{\sin^2\theta} \Big) - i \sum_{\mu} \sum_{\alpha} \hat{\lambda}_{\mu}^{\alpha} \rho^{\alpha} z^{\mu} \\
&- \frac{i}{\sin^2\theta} \big(u_{\mu} - z^{\mu}\cos\theta\big) \sum_{\alpha} \hat{\lambda}_{\mu}^{\alpha} (R^{\alpha} - \rho^{\alpha}\cos\theta) \Big] \Big\rangle_{z^{\mu}}
\end{aligned}
\tag{30}
$$

Now we introduce integral representations for the remaining delta functions, including the measure $d\mu(\mathbf{J}^{\alpha})$, for which we introduce the parameter $\hat{k}^{\alpha}$,

$$
\begin{aligned}
\Omega^{(n)} = &\int \prod_\alpha \frac{d\hat{k}^\alpha}{4\pi} \int \prod_{\alpha<\beta} \frac{dq^{\alpha\beta} d\hat{q}^{\alpha\beta}}{2\pi/N} \int \prod_\alpha \frac{dR^\alpha d\hat{R}^\alpha}{2\pi/N} \int \prod_\alpha \frac{d\rho^\alpha d\hat{\rho}^\alpha}{2\pi/N} \\
&\times \exp\left(i\frac{N}{2} \sum_\alpha \hat{k}^\alpha + iN \sum_{\alpha<\beta} q^{\alpha\beta}\hat{q}^{\alpha\beta} + iN \sum_\alpha R^\alpha \hat{R}^\alpha + iN \sum_\alpha \rho^\alpha \hat{\rho}^\alpha \right) \\
&\times \int \prod_{i,\alpha} \frac{dJ_i^\alpha}{\sqrt{2\pi e}} \exp\left(-\frac{i}{2} \sum_\alpha \hat{k}^\alpha \|\mathbf{J}^\alpha\|^2 - i \sum_{\alpha<\beta} \hat{q}^{\alpha\beta} \mathbf{J}^\alpha \cdot \mathbf{J}^\beta - i \sum_\alpha \hat{R}_\alpha \mathbf{J}^\alpha \cdot \mathbf{T} - i \sum_\alpha \hat{\rho}_\alpha \mathbf{J}^\alpha \cdot \mathbf{J}_{\text{probe}} \right) \\
&\times \int \prod_{\alpha,\mu} \frac{d\lambda_\mu^\alpha d\hat{\lambda}_\mu^\alpha}{2\pi} \int \prod_\mu \frac{du_\mu d\hat{u}_\mu}{2\pi} \prod_{\alpha,\mu} \Theta\big(u_\mu(\lambda_\mu^\alpha - \kappa)\big) \exp\left(i \sum_{\mu,\alpha} \lambda_\mu^\alpha \hat{\lambda}_\mu^\alpha - \frac{1}{2} \frac{(u_\mu - z^\mu \cos\theta)^2}{\sin^2\theta} \right) \\
&\times \Bigg\langle \exp\Bigg[-\frac{1}{2} \sum_\mu \sum_{\alpha\beta} \hat{\lambda}_\mu^\alpha \hat{\lambda}_\mu^\beta \left(q^{\alpha\beta} - \rho^\alpha \rho^\beta - \frac{(R^\alpha - \rho^\alpha \cos\theta)(R^\beta - \rho^\beta \cos\theta)}{\sin^2\theta} \right) - i \sum_\mu \sum_\alpha \hat{\lambda}_\mu^\alpha \rho^\alpha z^\mu \\
&- \frac{i}{\sin^2\theta} \big(u_\mu - z^\mu \cos\theta\big) \sum_\alpha \hat{\lambda}_\mu^\alpha (R^\alpha - \rho^\alpha \cos\theta) \Bigg] \Bigg\rangle_{z^\mu}
\end{aligned}
\tag{31}
$$

Notice that the $u_\mu - \lambda_\mu^\alpha - \hat{\lambda}_\mu^\alpha - z_\mu$-integrals factorize in μ, and can be written as a single integral to the power of $P = \alpha N$.

$$
\begin{aligned}
\Omega^{(n)} = k &\int \prod_\alpha \frac{d\hat{k}^\alpha}{4\pi} \int \prod_{\alpha<\beta} \frac{dq^{\alpha\beta} d\hat{q}^{\alpha\beta}}{2\pi/N} \int \prod_\alpha \frac{dR^\alpha d\hat{R}^\alpha}{2\pi/N} \int \prod_\alpha \frac{d\rho^\alpha d\hat{\rho}^\alpha}{2\pi/N} \\
&\times \exp\Bigg(N \Bigg[\frac{i}{2} \sum_\alpha \hat{k}^\alpha + i \sum_{\alpha<\beta} q^{\alpha\beta}\hat{q}^{\alpha\beta} + i \sum_\alpha R^\alpha \hat{R}^\alpha + i \sum_\alpha \rho^\alpha \hat{\rho}^\alpha \\
&+ G_S(\hat{k}^\alpha, \hat{q}^{\alpha\beta}, \hat{R}^\alpha, \hat{\rho}^\alpha) + \alpha G_E(q^{\alpha\beta}, R^\alpha, \rho^\alpha) \Bigg] \Bigg)
\end{aligned}
\tag{32}
$$

Where we have written the Gardner volume in terms of an *entropic* part G_S, which measures how many spherical couplings satisfy the constraints,

$$
G_S = \frac{1}{N} \log \int \prod_\alpha \frac{d\mathbf{J}^\alpha}{\sqrt{2\pi e}} \exp\left(-\frac{i}{2} \sum_\alpha \hat{k}^\alpha \|\mathbf{J}^\alpha\|^2 - i \sum_{\alpha<\beta} \hat{q}^{\alpha\beta} \mathbf{J}^\alpha \cdot \mathbf{J}^\beta - i \sum_\alpha \hat{R}^\alpha \mathbf{J}^\alpha \cdot \mathbf{T} - i \sum_\alpha \hat{\rho}^\alpha \mathbf{J}^\alpha \cdot \mathbf{J}_{\text{probe}} \right)
\tag{33}
$$

And an *energetic* part G_E,

$$
\begin{aligned}
G_E = \log &\int \frac{du}{\sqrt{2\pi}} \int \prod_\alpha \frac{d\lambda^\alpha d\hat{\lambda}^\alpha}{2\pi} \prod_\alpha \Theta\big(u(\lambda^\alpha - \kappa)\big) \exp\left(i \sum_\alpha \lambda^\alpha \hat{\lambda}^\alpha - \frac{1}{2} \frac{(u_\mu - z^\mu \cos\theta)^2}{\sin^2\theta} \right) \\
\times \Bigg\langle \exp\Bigg[&-\frac{1}{2} \sum_{\alpha\beta} \hat{\lambda}^\alpha \hat{\lambda}^\beta \left(q^{\alpha\beta} - \rho^\alpha \rho^\beta - \frac{(R^\alpha - \rho^\alpha \cos\theta)(R^\beta - \rho^\beta \cos\theta)}{\sin^2\theta} \right) - i \sum_\alpha \hat{\lambda}^\alpha \rho^\alpha z \\
&- \frac{i}{\sin^2\theta} \big(u - z\cos\theta\big) \sum_\alpha \hat{\lambda}^\alpha (R^\alpha - \rho^\alpha \cos\theta) \Bigg] \Bigg\rangle_z
\end{aligned}
\tag{34}
$$

We first evaluate the entropic part, G_S, by introducing the $n \times n$ matrices A, B,

$$
A_{\alpha\beta} = i\hat{k}^\alpha \delta_{\alpha\beta} + i\hat{q}^{\alpha\beta}(1 - \delta_{\alpha\beta}) \tag{35}
$$

$$
B_{\alpha\beta} = \delta_{\alpha\beta} + q^{\alpha\beta}(1 - \delta_{\alpha\beta}) \tag{36}
$$

Inserting this our expression for G_S becomes

$$G_S = \frac{1}{N}\log\int\prod_\alpha \frac{d\mathbf{J}^\alpha}{\sqrt{2\pi e}}\exp\left(-\frac{1}{2}\sum_{\alpha,\beta}\mathbf{J}^{\alpha T}A_{\alpha\beta}\mathbf{J}^\beta - i\sum_\alpha \mathbf{J}^\alpha\cdot(\mathbf{T}\hat{R}^\alpha + \mathbf{J}_{\text{probe}}\hat{\rho}^\alpha)\right) \tag{37}$$

Integrating over $\mathbf{J}^\alpha$,

$$G_S = -\frac{n}{2} - \frac{1}{2}\log(\det A) - \frac{1}{2N}\sum_{\alpha,\beta}(\mathbf{T}\hat{R}^\alpha + \mathbf{J}_{\text{probe}}\hat{\rho}^\alpha)^T A^{-1}_{\alpha\beta}(\mathbf{T}\hat{R}^\beta + \mathbf{J}_{\text{probe}}\hat{\rho}^\beta) \tag{38}$$

Now we can include the remaining terms in the expression for $\Omega^{(n)}$ outside of G_E and G_S by noting that

$$tr(AB) = \sum_{\alpha\beta} A_{\alpha\beta}B_{\beta\alpha} \tag{39}$$

$$= \sum_{\alpha\beta}(i\hat{k}^\alpha\delta_{\alpha\beta} + i\hat{q}^{\alpha\beta}(1-\delta_{\alpha\beta}))(\delta_{\alpha\beta} + q^{\alpha\beta}(1-\delta_{\alpha\beta})) \tag{40}$$

$$= \sum_\alpha i\hat{k}^\alpha + 2\sum_{\alpha<\beta} iq^{\alpha\beta}\hat{q}^{\alpha\beta} \tag{41}$$

Additionally, we can use $\log\det A = tr(\log A)$. Thus all terms in the exponent except G_E can be written as

$$-\frac{n}{2}-\frac{1}{2}tr(\log A)-\frac{1}{2N}\sum_{\alpha,\beta}(\mathbf{T}\hat{R}^\alpha+\mathbf{J}_{\text{probe}}\hat{\rho}^\alpha)^T A^{-1}_{\alpha\beta}(\mathbf{T}\hat{R}^\beta+\mathbf{J}_{\text{probe}}\hat{\rho}^\beta)+\frac{1}{2}tr(AB)+i\sum_\alpha R^\alpha\hat{R}^\alpha+i\sum_\alpha\rho^\alpha\hat{\rho}^\alpha\mathbf{J}_{\text{probe}} \tag{42}$$

Now we extremize wrt $\hat{R}^\alpha$ and the elements of A by setting the derivatives wrt $\hat{R}^\gamma$, $\hat{\rho}^\gamma$ and $A^{\gamma\delta}$ equal to zero:

$$0 = -\sum_\alpha A^{-1}_{\alpha\gamma}\mathbf{T}\cdot(\mathbf{T}\hat{R}^\alpha + \mathbf{J}_{\text{probe}}\hat{\rho}^\alpha) + iR^\gamma = -\sum_\alpha A^{-1}_{\alpha\gamma}(\hat{R}^\alpha + \hat{\rho}^\alpha\cos\theta) + iR^\gamma \tag{43}$$

$$0 = -\sum_\alpha A^{-1}_{\alpha\gamma}\mathbf{J}_{\text{probe}}\cdot(\mathbf{T}\hat{R}^\alpha + \mathbf{J}_{\text{probe}}\hat{\rho}^\alpha) + i\rho^\gamma = -\sum_\alpha A^{-1}_{\alpha\gamma}(\hat{R}^\alpha\cos\theta + \hat{\rho}^\alpha) + i\rho^\gamma \tag{44}$$

$$0 = -\frac{1}{2}A^{-1}_{\gamma\delta} + \frac{1}{2}\sum_{\alpha,\beta}(\mathbf{T}\hat{R}^\alpha + \mathbf{J}_{\text{probe}}\hat{\rho}^\alpha)^T A^{-1}_{\alpha\gamma}A^{-1}_{\beta\delta}(\mathbf{T}\hat{R}^\beta + \mathbf{J}_{\text{probe}}\hat{\rho}^\beta) + \frac{1}{2}B_{\gamma\delta} \tag{45}$$

Solving these gives

$$\hat{R}^\alpha = i\sum_\beta A_{\alpha\beta}\frac{R^\beta - \rho^\beta\cos\theta}{\sin^2\theta} \tag{46}$$

$$\hat{\rho}^\alpha = i\sum_\beta A_{\alpha\beta}\frac{\rho^\beta - R^\beta\cos\theta}{\sin^2\theta} \tag{47}$$

and

$$A^{-1}_{\gamma\delta} = B_{\gamma\delta} - \frac{R^\gamma R^\delta - R^\gamma\rho^\delta\cos\theta - R^\delta\rho^\gamma\cos\theta + \rho^\gamma\rho^\delta}{\sin^2\theta} \equiv C_{\gamma\delta} \tag{48}$$

and now we are left with

$$\Omega^{(n)} \sim \exp\left(N\text{extr}_{q^{\alpha\beta},R^\alpha,\rho^\alpha}\left[\frac{1}{2}tr(\log C) + \alpha G_E(q^{\alpha\beta}, R^\alpha)\right]\right) \tag{49}$$

A.3.1 Replica symmetry ansatz

In order to extremize wrt $q^{\alpha\beta}, R^\alpha, \rho^\alpha$, we take the replica symmetry ansatz [21],

$$q^{\alpha\beta} = q, \quad R^\alpha = R, \quad \rho^\alpha = \rho \tag{50}$$

Then C takes the form

$$C_{\alpha\beta} = \delta_{\alpha\beta} - \frac{R^2 - 2R\rho\cos\theta + \rho^2}{\sin^2\theta} + q(1-\delta_{\alpha\beta}) \tag{51}$$

A matrix with E on the diagonal and F elsewhere, $C_{\alpha\beta} = E\delta_{\alpha\beta} + F(1-\delta_{\alpha\beta})$, has $n-1$ degenerate eigenvalues $E-F$ and one eigenvalue $E+(n-1)F$. Hence in our case C has $n-1$ degenerate eigenvalues

$$\left(1 - \frac{R^2 - 2R\rho\cos\theta + \rho^2}{\sin^2\theta}\right) - \left(q - \frac{R^2 - 2R\rho\cos\theta + \rho^2}{\sin^2\theta}\right) = 1 - q \tag{52}$$

and one other eigenvalue,

$$\left(1 - \frac{R^2 - 2R\rho\cos\theta + \rho^2}{\sin^2\theta}\right) + (n-1)\left(q - \frac{R^2 - 2R\rho\cos\theta + \rho^2}{\sin^2\theta}\right) = 1 - q + n\left(q - \frac{R^2 - 2R\rho\cos\theta + \rho^2}{\sin^2\theta}\right) \tag{53}$$

Therefore,

$$\begin{aligned} tr(\log C) &= (n-1)\log\left(1-q\right) + \log\left[1 - q + n\left(q - \frac{R^2 - 2R\rho\cos\theta + \rho^2}{\sin^2\theta}\right)\right] \\ &= n\log\left(1-q\right) + \log\left[1 + n\left(\frac{q\sin^2\theta - (R^2 - 2R\rho\cos\theta + \rho^2)}{(1-q)\sin^2\theta}\right)\right] \end{aligned} \tag{54}$$

We next evaluate the energetic part, G_E,

$$\begin{aligned} G_E = \log \int \frac{du}{\sqrt{2\pi}} \int \prod_\alpha \frac{d\lambda^\alpha d\hat{\lambda}^\alpha}{2\pi} \prod_\alpha \Theta\big(u(\lambda^\alpha - \kappa)\big) \exp\left(i\sum_\alpha \lambda^\alpha \hat{\lambda}^\alpha - \frac{1}{2}\frac{(u - z\cos\theta)^2}{\sin^2\theta}\right) \\ \times \Bigg\langle \exp\Bigg[-\frac{1}{2}\sum_\alpha (\hat{\lambda}^\alpha)^2 \left(1 - \rho^2 - \frac{(R-\rho\cos\theta)^2}{\sin^2\theta}\right) - \frac{1}{2}\sum_{\alpha\neq\beta} \hat{\lambda}^\alpha\hat{\lambda}^\beta \left(q - \rho^2 - \frac{(R-\rho\cos\theta)^2}{\sin^2\theta}\right) \\ - i\sum_\alpha \hat{\lambda}^\alpha \rho^\alpha z - \frac{i}{\sin^2\theta}\big(u - z\cos\theta\big)\sum_\alpha \hat{\lambda}^\alpha (R^\alpha - \rho^\alpha\cos\theta)\Bigg]\Bigg\rangle_z \end{aligned} \tag{55}$$

First note that we can rewrite the terms

$$\begin{aligned} &-\frac{1}{2}\sum_\alpha \left(1 - \rho^2 - \frac{(R-\rho\cos\theta)}{\sin^2\theta}\right)(\hat{\lambda}^\alpha)^2 - \frac{1}{2}\sum_{\alpha\neq\beta}\hat{\lambda}^\alpha\hat{\lambda}^\beta\left(q - \rho^2 - \frac{(R-\rho\cos\theta)^2}{\sin^2\theta}\right) \\ &= -\frac{1}{2}\sum_\alpha \left(1 - \rho^2 - \frac{(R-\rho\cos\theta)^2}{\sin^2\theta}\right)(\hat{\lambda}^\alpha)^2 - \frac{1}{2}\left(q - \rho^2 - \frac{(R-\rho\cos\theta)^2}{\sin^2\theta}\right)\left[(\sum_\alpha \hat{\lambda}^\alpha)^2 - \sum_\alpha (\hat{\lambda}^\alpha)^2\right] \\ &= -\frac{1}{2}\sum_\alpha (1-q)(\hat{\lambda}^\alpha)^2 - \frac{1}{2}\left(q - \rho^2 - \frac{(R-\rho\cos\theta)^2}{\sin^2\theta}\right)(\sum_\alpha \hat{\lambda}^\alpha)^2 \end{aligned} \tag{56}$$

To simplify the last term we apply the Hubbard-Stratonovich transformation, $e^{b^2/2} = \int Dt e^{bt}$, introducing auxiliary field t,

$$= \log \int \frac{du}{\sqrt{2\pi}} \int \prod_\alpha \frac{d\lambda^\alpha d\hat{\lambda}^\alpha}{2\pi} \prod_\alpha \Theta\big(u(\lambda^\alpha - \kappa)\big) \int Dt \Big\langle \exp\Big[-\frac{1-q}{2} \sum_\alpha (\hat{\lambda}^\alpha)^2 + i \sum_\alpha \hat{\lambda}^\alpha \Big(\lambda^\alpha - \rho^\alpha z - \frac{(u - z\cos\theta)(R^\alpha - \rho^\alpha \cos\theta)}{\sin^2\theta} - \sqrt{q - \rho^2 - \frac{(R - \rho\cos\theta)^2}{\sin^2\theta}} t \Big) - \frac{1}{2} \frac{(u - z\cos\theta)^2}{\sin^2\theta} \Big] \Big\rangle_z \tag{57}$$

Using the Θ-function to restrict the bounds of integration,

$$= \log 2 \int_0^\infty \frac{du}{\sqrt{2\pi}} \int_\kappa^\infty \prod_\alpha \frac{d\lambda^\alpha}{\sqrt{2\pi}} \int \prod_\alpha \frac{d\hat{\lambda}^\alpha}{\sqrt{2\pi}} \int Dt \Big\langle \exp\Big[-\frac{1-q}{2} \sum_\alpha (\hat{\lambda}^\alpha)^2 + i \sum_\alpha \hat{\lambda}^\alpha \Big(\lambda^\alpha - \rho^\alpha z - \frac{(u - z\cos\theta)(R^\alpha - \rho^\alpha \cos\theta)}{\sin^2\theta} - \sqrt{q - \rho^2 - \frac{(R - \rho\cos\theta)^2}{\sin^2\theta}} t \Big) - \frac{1}{2} \frac{(u - z\cos\theta)^2}{\sin^2\theta} \Big] \Big\rangle_z \tag{58}$$

Now we can perform the gaussian integrals over $\hat{\lambda}^\alpha$,

$$= \log 2 \int_0^\infty \frac{du}{\sqrt{2\pi}} \int_\kappa^\infty \prod_\alpha \frac{d\lambda^\alpha}{\sqrt{2\pi}} \int Dt \Big\langle \exp\Big[-\frac{1}{2(1-q)} \Big(\lambda^\alpha - \rho^\alpha z - \frac{(u - z\cos\theta)(R^\alpha - \rho^\alpha \cos\theta)}{\sin^2\theta} - \sqrt{q - \rho^2 - \frac{(R - \rho\cos\theta)^2}{\sin^2\theta}} t \Big)^2 - \frac{1}{2} \frac{(u - z\cos\theta)^2}{\sin^2\theta} \Big] \Big\rangle_z \tag{59}$$

And λ^α,

$$= \log 2 \int_0^\infty \frac{du}{\sqrt{2\pi}} \int Dt \Big\langle H^n \Big[-\frac{1}{\sqrt{1-q}} \Big(\kappa - \rho^\alpha z + \frac{(u - z\cos\theta)(R^\alpha - \rho^\alpha \cos\theta)}{\sin^2\theta} + \sqrt{q - \rho^2 - \frac{(R - \rho\cos\theta)^2}{\sin^2\theta}} t \Big)^2 \Big] \times \exp\Big[-\frac{1}{2} \frac{(u - z\cos\theta)^2}{\sin^2\theta} \Big] \Big\rangle_z \tag{60}$$

Shifting the integration variable $t \to (\rho^\alpha z + \frac{(u - z\cos\theta)(R^\alpha - \rho^\alpha \cos\theta)}{\sin^2\theta} + \sqrt{q - \rho^2 - \frac{(R - \rho\cos\theta)^2}{\sin^2\theta}} t)/\sqrt{q}$, we can finally perform the gaussian integral over u,

$$= \log 2 \int \frac{dt}{\sqrt{2\pi}} \Big\langle \exp\Big(-\frac{(\sqrt{q}t - z\rho)^2}{2(q-\rho)^2} \Big) \sqrt{\frac{q}{q-\rho^2}} H^n \Big(-\sqrt{\frac{q}{1-q}} t \Big) \times H\Big(\frac{1}{\sqrt{q-\rho^2}} \frac{\kappa - (qR_0 z + z\rho(R - 2\rho\cos\theta) - \sqrt{q}t(R - \rho\cos\theta))}{\sqrt{q\sin^2\theta + 2R\rho\cos\theta - R^2 - \rho}} \Big) \Big\rangle_z \tag{61}$$

We can simplify this further by taking $t \to (\sqrt{q}t - z\rho)/\sqrt{q - \rho^2}$,

$$= \log 2 \int Dt \Big\langle H^n \Big(\frac{\kappa - (z\rho + \sqrt{q-\rho^2})t}{\sqrt{1-q}} \Big) H\Big(\frac{t\left(\sqrt{q-\rho^2} + \rho z\right)(R - \rho\cos\theta) + qz\cos\theta - \rho R z}{\sqrt{-(q-\rho^2)\left(\rho^2 - q\sin^2\theta + R^2 - 2\rho R\cos\theta\right)}} \Big) \Big\rangle_z \tag{62}$$

A.4 Quenched entropy

Putting everything together, and using the replica identity, $\langle \log \Omega \rangle$=$\lim_{n\to 0}(\langle \Omega^n \rangle - 1)/n$, we obtain an expression for the quenched entropy of the teacher-student perceptron under data pruning:

$$\boxed{\begin{aligned}\frac{1}{N}\langle \log \Omega \rangle = \text{extr}_{q,R,\rho}\Bigg[&\frac{1}{2}\log\Big(1-q\Big) + \frac{1}{2}\Bigg(\frac{q-(R^2-2R\rho\cos\theta+\rho^2)/\sin^2\theta}{1-q}\Bigg) \\ &+ 2\alpha\Bigg\langle \int Dt \log H\Bigg(\frac{\kappa-(z\rho+\sqrt{q-\rho^2})t}{\sqrt{1-q}}\Bigg) \\ &\times H\Bigg(\frac{t\left(\sqrt{q-\rho^2}+\rho z\right)(R-\rho\cos\theta)+z(q\cos\theta-\rho R)}{\sqrt{(q-\rho^2)\left(R^2+\rho^2-q\sin^2\theta-2\rho R\cos\theta\right)}}\Bigg)\Bigg\rangle_z\Bigg]\end{aligned} \tag{63}}$$

We will now unpack this equation and use it to make predictions in several specific settings.

A.5 Perfect teacher-probe overlap

We will begin by considering the case where the probe student has learned to perfectly match the teacher, $J_{\text{probe}} = T$, which we can obtain by the limit $\theta \to 0$, $\rho \to R$. In this limit the second H-function in Eq. 63 becomes increasingly sharp, approaching a step function:

$$H\Bigg(\frac{t\left(\sqrt{q-\rho^2}+\rho z\right)(R-\rho\cos\theta)+z(q\cos\theta-\rho R)}{\sqrt{(q-\rho^2)\left(R^2+\rho^2-q\sin^2\theta-2\rho R\cos\theta\right)}}\Bigg) \to \Theta(z) \tag{64}$$

Hence we are left with,

$$\boxed{\begin{aligned}\frac{1}{N}\langle\langle \ln \Omega(\mathbf{x}^\mu, T, \kappa)\rangle\rangle = \text{extr}_{q,R}\Bigg[&\frac{1}{2}\log\Big(1-q\Big) + \frac{1}{2}\Bigg(\frac{q-R^2}{1-q}\Bigg) \\ &+ 2\alpha\int Dt \int dz p(z)\Theta(z)\log H\Bigg(-\frac{\sqrt{q-R^2}t+Rz-\kappa}{\sqrt{1-q}}\Bigg)\Bigg]\end{aligned} \tag{65}}$$

A.5.1 Saddle point equations

We can now obtain a set of self-consistent saddle point equations by setting set to zero the derivatives with respect to R and q of the right side of Eq. 65. As κ approaches its critical value, the space of solutions shrinks to a unique solution, and hence the overlap between students q approaches one. In the limit $q \to 1$, after some partial integration, we find,

$$R = 2\alpha\int_{-\infty}^{\kappa}\frac{dt}{\sqrt{2\pi}\sqrt{1-R^2}}\int_0^\infty dz p(z)\exp\Bigg(-\frac{(t-Rz)^2}{2(1-R^2)}\Bigg)\Bigg(\frac{z-Rt}{1-R^2}\Bigg)(\kappa-t) \tag{66}$$

$$1-R^2 = 2\alpha\int_{-\infty}^{\kappa}\frac{dt}{\sqrt{2\pi}\sqrt{1-R^2}}\int_0^\infty dz p(z)\exp\Bigg(-\frac{(t-Rz)^2}{2(1-R^2)}\Bigg)\big(\kappa-t\big)^2 \tag{67}$$

These saddle point equations can be solved numerically to find R and κ as a function of α for a student perceptron trained on a dataset with an arbitrary distribution along the teacher direction $p(z)$. We can specialize to the case of data pruning by setting $p(z)$ to the distribution found after pruning an initially Gaussian-distributed dataset so that only a fraction f of those examples with the smallest margin along the teacher are kept, $p(z) = \frac{e^{-z^2/2}}{\sqrt{2\pi}f}\Theta(\gamma - |z|)$, where the threshold $\gamma = H^{-1}\big(\frac{1-f}{2}\big)$.

$$R = \frac{2\alpha}{f\sqrt{2\pi}\sqrt{1-R^2}} \int_{-\infty}^{\kappa} Dt \, \exp\left(-\frac{R^2t^2}{2(1-R^2)}\right)\left[1-\exp\left(-\frac{\gamma(\gamma-2Rt)}{2(1-R^2)}\right)\right](\kappa-t) \quad (68)$$

$$1-R^2 = \frac{2\alpha}{f}\int_{-\infty}^{\kappa} Dt \left[H\left(-\frac{Rt}{\sqrt{1-R^2}}\right) - H\left(-\frac{Rt-\gamma}{\sqrt{1-R^2}}\right)\right](\kappa-t)^2 \quad (69)$$

Solving these saddle point equations numerically for R and κ yields an excellent fit to numerical simulations, as can be seen in Fig. 1**A**. It is also easy to verify that in the limit of no data pruning ($f \to 1, \gamma \to \infty$) we recover the saddle point equations for the classical teacher-student perceptron (Eqs. 4.4 and 4.5 in [21]),

$$R = \frac{2\alpha}{\sqrt{2\pi}\sqrt{1-R^2}} \int Dt \exp\left(-\frac{R^2t^2}{2(1-R^2)}\right)(\kappa-t) \quad (70)$$

$$1-R^2 = 2\alpha \int Dt \, H\left(-\frac{Rt}{\sqrt{1-R^2}}\right)\big(\kappa-t\big)^2 \quad (71)$$

A.6 Information gain per example

Why does data pruning allow for super-exponential performance with dataset size α? We can define the amount of information gained from each new example, $I(\alpha)$,as the fraction by which the space of solutions which perfectly classify the data is reduced when a new training example is added, $I(\alpha) = \Omega(\frac{P+1}{N})/\Omega(\frac{P}{N})$. Or, equivalently, the rate at which the entropy is reduced, $I(\alpha) = -\frac{d}{d\alpha}S(\alpha)$. Of coure, the volume of solutions shrinks to zero at the max-margin solution; so to study the volume of solutions which perfectly classify the data we simply set the margin to zero $\kappa = 0$. In [27] the information gain for a perceptron in the classical teacher-student setting is shown to take the form,

$$I(\alpha) = -2\int Dt \, H\left(\sqrt{\frac{R}{1-R}}t\right)\log H\left(\sqrt{\frac{R}{1-R}}t\right) \quad (72)$$

Which goes to zero in the limit of large α as $I(\alpha) \sim 1/\alpha$. Data pruning can increase the information gained per example by pruning away the uninformative examples. To show this, we generalize the calculation of the information gain to pruned datasets, using the expression for the entropy we obtained in the previous section (Eq. 65).

$$S(\alpha) = \frac{1}{N}\langle \log \Omega \rangle = \text{extr}_{q,R}\left[\frac{1}{2}\log\Big(1-R\Big)+\frac{1}{2}R+2\alpha\int Dt\int_0^{\infty} dz\, p(z)\log H\left(-\sqrt{R}t-\frac{R}{\sqrt{1-R}}z\right)\right] \quad (73)$$

Hence the information gain $I(\alpha) = -\frac{d}{d\alpha}S(\alpha)$ is given by

$$I(\alpha) = 2\alpha\int Dt\int dz p(z)\Theta(z)\log H\left(-\sqrt{R}t - \frac{R}{\sqrt{1-R}}z\right) \quad (74)$$

Changing variables to $t \to -(\sqrt{R}t + \frac{R}{\sqrt{1-R}}z)/\sqrt{q}$,

$$I(\alpha) = 2\alpha\int Dt\int_0^{\infty} dz p(z)\frac{1}{\sqrt{1-R}}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z^2+2\sqrt{R}tz}{2(1-R)}\right)\log H\left(\sqrt{\frac{R}{1-R}}t\right) \quad (75)$$

Now, assuming that we prune to a fraction f, so that $p(z) = \Theta\big(|z|-\gamma\big)\frac{\exp(-z^2/2)}{\sqrt{2\pi}f}$, where $\gamma = H^{-1}\left(\frac{1-f}{2}\right)$

$$I(\alpha) = \frac{2\alpha}{f} \int Dt \left[H\left(\sqrt{\frac{R}{1-R}}t\right) - H\left(\frac{\gamma + \sqrt{R}t}{\sqrt{1-R}}\right)\right] \log H\left(\sqrt{\frac{R}{1-R}}t\right)\Bigg] \tag{76}$$

$I(\alpha)$ is plotted for varying values of f in Fig. 1**F**. Notice that for $f \to 1$, $\gamma \to \infty$ and we recover Eq. 72. To obtain the optimal pruning fraction f_{opt} for any α, we first need an equation for R, which can be obtain by taking the saddle point of Eq. 73. Next we optimize $I(\alpha)$ by setting the derivative of Eq. 76 with respect to f equal to zero. This gives us a pair of equations which can be solved numerically to obtain f_{opt} for any α.

Finally, Eq. 76 reveals that as we prune more aggressively the information gain per example approaches a finite rate. As $f \to 0$, $\gamma \to 0$, and we obtain,

$$I(\alpha) = -\int Dt \ \log H(\sqrt{R}t) \tag{77}$$

Which allows us to produce to trace the Pareto frontier in Fig. 1**F**. For $R \to 1$, Eq. 77 gives the asymptotic information gain $I(\infty) = 1$ nat/example.

A.7 Imperfect teacher-probe overlap

In realistic settings we expect the probe student to have only partial information about the target function. What happens if the probe student does not perfectly match the teacher? To understand this carefully, we need to compute the full set of saddle point equations over R, q, and ρ, which we will do in the following section. But to first get an idea for what goes wrong, we include in this section a simple sketch which reveals the limiting behavior.

Consider the case where the angle between the probe student and teacher is θ. Rotate coordinates so that the first canonical basis vector aligns with the student $J = (1, 0, \ldots, 0)$, and the teacher lies in the span of the first two canonical basis vectors, $T = (\cos\theta, \sin\theta, 0, \ldots, 0)$. Consider the margin along the teacher of a new training example x drawn from the pruned distribution.

$$\mathbb{E}|T \cdot x|^2 = \mathbb{E}[x_0^2 \cos^2\theta + x_1^2 \sin^2\theta] \tag{78}$$

As the fraction of examples kept goes to zero, $\mathbb{E}x_0^2 \to 0$, and the average margin of a new example converges to a fixed value,

$$\mathbb{E}|T \cdot x|^2 = \sin^2\theta \tag{79}$$

Hence the data ultimately stops concentrating around the teacher's decision boundary, and the information gained from each new example goes to zero. Therefore we expect the generalization error to converge to a power law, where the constant prefactor is roughly that of pruning with a prune fraction f_{min} which yields an average margin of $1 - R^2$. This "minimum" pruning fraction lower bounds the generalization error envelope (see Fig. 2), and satisfies the following equation,

$$\int_{-\gamma_{\text{min}}}^{\gamma_{\text{min}}} dx p(x) x^2 = \frac{1}{2} - \frac{e^{-\gamma_{\text{min}}^2/2}\gamma_{\text{min}}}{\sqrt{2\pi}(1 - 2H(\gamma_{\text{min}}))} = 1 - R^2 \tag{80}$$

where $\gamma_{\text{min}} = H^{-1}\left(\frac{1-f_{\text{min}}}{2}\right)$. Eq. 80 can be solved numerically, and we use it to produce the lower-bounding power laws shown in red in Fig. 2**C,D**. The minimum achievable pruning fraction $f_{\text{min}}(\theta)$ approaches zero as the angle between the probe student and the teacher shrinks, and we can obtain its scaling by taking $R \to 1$, in which case we find,

$$f_{\text{min}}(\theta) \sim \theta \tag{81}$$

A.8 Optimal pruning policy

The saddle point equations Eq. 66,67 reveal that the optimal pruning policy varies as a function of α_{prune}. For α_{prune} large the best policy is to retain only the “hardest" (smallest-margin) examples. But when α_{prune} is small, keeping the “hardest" examples performs worse than chance, suggesting that the best policy in the α_{prune} small regime is to keep the easiest examples. Indeed by switching between the “keep easy” and “keep hard” strategies as α_{prune} grows, one can achieve a lower Pareto frontier than the one shown in Fig. 1A in the small α_{prune} regime (Fig. 7C).

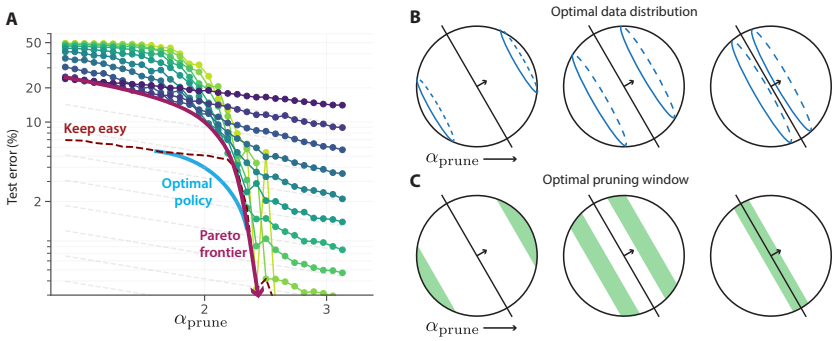


Figure 7: Optimal pruning policy as a function of α.

These observations beg the question: what is the best policy in the intermediate α_{prune} regime? Is there a globally optimal pruning policy which interpolates between the “keep easy” and “keep hard” strategies and achieves the lowest possible Pareto frontier (blue curve in Fig. 7A)?

In this section we investigate this question. Using the calculus of variations, we first derive the optimal data distribution $p(z|\alpha_{\text{prune}}, f)$ along the teacher for a given α_{prune}, f. We begin by framing the problem using the method of Lagrange multipliers. Seeking to optimize R under the constraints imposed by the saddle point equations Eqs. 66,67, we define the Lagrangian,

$$\mathcal{L} = R + \mu\left(R - 2\alpha \int_0^\infty dz\, p(z)\varphi(z; R, k)\right) + \lambda\left(1 - R^2 - 2\alpha \int_0^\infty dz\, p(z)\psi(z; R, k)\right). \quad (82)$$

Where,

$$\varphi(z; R, k) = \int_{-\infty}^{\kappa} \frac{dt}{\sqrt{2\pi}\sqrt{1-R^2}} \exp\left(-\frac{(t-Rz)^2}{2(1-R^2)}\right)\left(\frac{z-Rt}{1-R^2}\right)(\kappa - t), \quad (83)$$

and,

$$\psi(z; R, k) = \int_{-\infty}^{\kappa} \frac{dt}{\sqrt{2\pi}\sqrt{1-R^2}} \exp\left(-\frac{(t-Rz)^2}{2(1-R^2)}\right)(\kappa - t)^2 \quad (84)$$

Taking a variational derivative $\frac{\delta \mathcal{L}}{\delta p}$ with respect to the data distribution p, we obtain an equation for z, indicating that the optimal distribution is a delta function at $z = z^*$. To find the optimal location of the delta function z^*, we take derivatives with respect to the remaining variables R, k, μ, λ and solve the resulting set of equations numerically. The qualitative behavior is shown in Fig. 7A. As α_{prune} grows, the location of the delta function shifts from infinity to zero, confirming that the optimal strategy for small α_{prune} is to keep the "easy" (large-margin) examples, and for large α_{prune} to keep the "hard" (small-margin) examples.

Interestingly, this calculation also reveals that if the location of the delta function is chosen optimally, the student can perfectly recover the teacher ($R = 1$, zero generalization error) for any α_{prune}. This observation, while interesting, is of no practical consequence because it relies on an infinitely large training set from which examples can be precisely selected to perfectly recover the teacher. Therefore,

to derive the optimal pruning policy for a more realistic scenario, we assume a gaussian distribution of data along the teacher direction and model pruning as keeping only those examples which fall inside a window $a < z < b$. The saddle point equations, Eqs. 66,67, then take the form,

$$R = \frac{2\alpha}{f\sqrt{\pi/2}\sqrt{1-R^2}} \int_{-\infty}^{\kappa} Dt \left[\exp\left(-\frac{(a-Rt)^2}{2(1-R^2)}\right) - \exp\left(-\frac{(b-Rt)^2}{2(1-R^2)}\right) \right] (\kappa - t) \tag{85}$$

$$1 - R^2 = \frac{4\alpha}{f} \int_{-\infty}^{\kappa} Dt \left[H\left(\frac{a-Rt}{\sqrt{1-R^2}}\right) - H\left(\frac{b-Rt}{\sqrt{1-R^2}}\right) \right] (\kappa - t)^2 \tag{86}$$

Where a must satisfy $a = H^{-1}(f/2 + H(b))$. For each f, α, we find the optimal location of this window using the method of Lagrange multipliers. Defining the Lagrangian as before,

$$\mathcal{L} = R + \mu\left(R - 2\alpha \int_0^{\infty} dz\, p(z)\varphi(z; R, k)\right) + \lambda\left(1 - R^2 - 2\alpha \int_0^{\infty} dz\, p(z)\psi(z; R, k)\right), \tag{87}$$

Where now,

$$\phi(b; R, k) = \left[\exp\left(-\frac{(a-Rt)^2}{2(1-R^2)}\right) - \exp\left(-\frac{(b-Rt)^2}{2(1-R^2)}\right) \right] (\kappa - t) \tag{88}$$

$$\psi(b; R, k) = \left[H\left(\frac{a-Rt}{\sqrt{1-R^2}}\right) - H\left(\frac{b-Rt}{\sqrt{1-R^2}}\right) \right] (\kappa - t)^2 \tag{89}$$

To find the optimal location of the pruning window, we take derivatives with respect to the remaining variables b, R, k, μ, λ and solve the resulting set of equations numerically. Consistent with the results for the optimal distribution, the location of the optimal window shifts from around infinity to around zero as α_{prune} grows (Fig. 7**C**).

A.9 Exact saddle point equations

To obtain exact expressions for the generalization error for all θ, we can extremize Eq. 63 wrt R, q, ρ.

Derivative wrt R

$$\frac{R - \rho\cos\theta}{(1-q)\sin^2\theta} = \int dt \int dz\, p(z) \frac{\sqrt{2}\alpha}{\pi} \left(\frac{t\sqrt{q-\rho^2}}{\sqrt{2}\sqrt{(\rho^2-q)}\Lambda} - \frac{(R-\rho\cos\theta)\Gamma(t,z)}{\sqrt{2}\sqrt{q-\rho^2}\Lambda^3} \right) \tag{90}$$

$$\times \log\left(H\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{1-q}} \right) \right) \exp\left(-\frac{\Gamma(t,z)^2}{2\left(q-\rho^2\right)\Lambda^2} - \frac{t^2}{2} \right) \tag{91}$$

Where we have defined,

$$\Lambda = \sqrt{q\sin^2\theta - R^2 - \rho^2 + 2R\rho\cos\theta}, \tag{92}$$

$$\Gamma(t,z) = (R - \rho\cos\theta)\left(t\sqrt{q-\rho^2} + \rho z\right) + qz\cos\theta - \rho R z. \tag{93}$$

Integrating the right-hand side by parts,

$$\frac{R - \rho\cos\theta}{(1-q)\sin^2\theta} = -\int dt \int dz\, p(z) \frac{\alpha}{\pi\Lambda} \frac{\sqrt{q-\rho^2} e^{-\frac{\left(\kappa - t\sqrt{q-\rho^2} - \rho z\right)^2}{2(1-q)}}}{\sqrt{2\pi}\sqrt{1-q}\mathrm{H}\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{1-q}}\right)} \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right) \tag{94}$$

where

$$\Delta(t,z) = 2tz\sqrt{q-\rho^2}(R-\rho\cos\theta)\cos\theta + qt^2\sin^2\theta + qz^2\cos^2\theta - \rho^2t^2\sin^2\theta - \rho^2z^2\cos^2\theta \quad (95)$$

Changing variables to $t \to t\sqrt{q-\rho^2} + \rho z$ and taking the limit $q \to 1$,

$$\frac{R-\rho\cos\theta}{\sin^2\theta} = \left\langle \int_{-\infty}^{\kappa} dt \frac{\alpha}{\pi\Lambda} \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)(\kappa - t)\right\rangle_z \quad (96)$$

Where with this change of variables,

$$\Lambda = \sqrt{q\sin^2\theta - R^2 - \rho^2 + 2\rho R\cos\theta} \quad (97)$$

$$\Delta(t,z) = z^2\left(\rho^2 + \cos^2\theta - 2\rho R\cos\theta\right) + 2tz(R\cos\theta - \rho) + t^2\sin^2\theta \quad (98)$$

Derivative wrt q,

$$\frac{q - (\rho^2 + R^2 - 2\rho R\cos\theta)/\sin^2\theta}{2(1-q)^2} = \int dt \int dz\, p(z)\frac{2\alpha}{\pi}\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{(1-q)^{3/2}} - \frac{t}{\sqrt{1-q}\sqrt{q-\rho^2}}\right) \quad (99)$$

$$\times \exp\left(-\frac{\left(\kappa - t\sqrt{q-\rho^2} - \rho z\right)^2}{2(1-q)} - \frac{t^2}{2}\right) \quad (100)$$

$$\times H\left(-\frac{(R-\rho\cos\theta)\left(t\sqrt{q-\rho^2} + \rho z\right) + qz\cos\theta - \rho R z}{\sqrt{(\rho^2 - q)\left(\rho^2 - q\sin^2\theta + R^2 - 2\rho R\cos\theta\right)}}\right) H\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{1-q}}\right) \quad (101)$$

$$-\frac{\sqrt{2}\alpha}{\pi}\left(\frac{\frac{t(R-\rho\cos\theta)}{2\sqrt{q-\rho^2}} + z\cos}{\sqrt{2}\sqrt{(q-\rho^2)}\Lambda} - \frac{\left(\left(q-\rho^2\right)\sin^2\theta + \Lambda^2\right)\Gamma(t,z)}{2\sqrt{2}\left(\rho^2 - q\right)^{3/2}\Lambda^3}\right) \quad (102)$$

$$\times \log\left(H\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{1-q}}\right)\right)\exp\left(-\frac{\left((R-\rho\cos\theta)\left(t\sqrt{q-\rho^2} + \rho z\right) + qz\cos\theta - \rho R z\right)^2}{2\left(q-\rho^2\right)\Lambda^2} - \frac{t^2}{2}\right) \quad (103)$$

Where $\Gamma(t,z) = (R-\rho\cos\theta)\left(t\sqrt{q-\rho^2} + \rho z\right) + qz\cos\theta - \rho R z$. After integating by parts,

$$\frac{q - (\rho^2 + R^2 - 2\rho R\cos\theta)/\sin^2\theta}{2(1-q)^2} = \int dt \int dz\, p(z)\frac{\alpha\exp\left(-\frac{2t\sqrt{q-\rho^2}(\rho z - \kappa) - \left(\rho^2 - 1\right)t^2 + (\kappa - \rho z)^2}{2(1-q)}\right)}{4\pi \mathrm{H}\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{1-q}}\right)^2} \quad (104)$$

$$\times\sqrt{\frac{2}{\pi}}e^{-\frac{\left(\kappa - t\sqrt{q-\rho^2} - \rho z\right)^2}{2(1-q)}}\mathrm{H}\left(-\frac{\Gamma(t,z)}{\sqrt{\left(q-\rho^2\right)\Lambda}}\right) \quad (105)$$

Changing variables to $t \to t\sqrt{q-\rho^2} + \rho z$ and taking the limit $q \to 1$,

$$1 - \frac{\rho^2 + R^2 - 2\rho R\cos\theta}{\sin^2\theta} = 2\alpha\Big\langle \int_{-\infty}^{\kappa} \frac{dt\, e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)(\kappa - t)^2\Big\rangle_z \tag{106}$$

Where now $\Gamma(t,z) = z(\rho R - \cos\theta) - t(R - \rho\cos\theta)$.

Derivative wrt ρ,

$$\frac{\rho - R\cos\theta}{(1-q)\sin^2\theta} = \frac{\alpha}{2\pi}\frac{\left(\frac{\rho t}{\sqrt{q-\rho^2}} - z\right)\exp\left(-\frac{\left(\kappa - t\sqrt{q-\rho^2} - \rho z\right)^2}{2(1-q)} - \frac{t^2}{2}\right) H\left(-\frac{\Gamma(t,z)}{\sqrt{q-\rho^2}\Lambda}\right)}{\sqrt{1-q}\mathrm{H}\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{1-q}}\right)} \tag{107}$$

$$+\frac{\sqrt{2}\alpha}{\pi}\Bigg(\frac{(R-\rho\cos\theta)\left(z - \frac{\rho t}{\sqrt{q-\rho^2}}\right) - \left(t\sqrt{q-\rho^2} + \rho z\right)\cos\theta - Rz}{\sqrt{2}\sqrt{q-\rho^2}\Lambda} \tag{108}$$

$$-\frac{\left(-2\rho\Lambda^2 - \left(q-\rho^2\right)\left(2\rho - 2R\cos\theta\right)\right)\Gamma(t,z)}{2\sqrt{2}\left(q-\rho^2\right)^{3/2}\Lambda^3}\Bigg) \tag{109}$$

$$\times \log\left(\frac{1}{2}\mathrm{erfc}\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{2}\sqrt{1-q}}\right)\right)\exp\left(-\frac{\Gamma(t,z)^2}{2\left(q-\rho^2\right)\Lambda^2} - \frac{t^2}{2}\right) \tag{110}$$

Integrating the second term by parts,

$$\frac{\rho - R\cos\theta}{(1-q)\sin^2\theta} = \frac{\alpha}{\pi}\frac{\left(\frac{\rho t}{\sqrt{q-\rho^2}} - z\right)\exp\left(-\frac{\left(\kappa - t\sqrt{q-\rho^2} - \rho z\right)^2}{2(1-q)} - \frac{t^2}{2}\right) H\left(-\frac{\Gamma(t,z)}{\sqrt{q-\rho^2}\Lambda}\right)}{\sqrt{1-q}\mathrm{H}\left(\frac{\kappa - t\sqrt{q-\rho^2} - \rho z}{\sqrt{1-q}}\right)} \tag{111}$$

$$-\frac{\alpha}{\pi\Delta}\exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)\left(\frac{\rho R - q\cos\theta}{q-\rho^2}\right) \tag{112}$$

Changing variables to $t \to t\sqrt{q-\rho^2} + \rho z$ and taking the limit $q \to 1$,

$$\frac{\rho - R\cos\theta}{\sin^2\theta} = 2\alpha\Big\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)\left(\frac{z-\rho t}{1-\rho^2}\right)(\kappa - t) \tag{113}$$

$$+\frac{1}{2\pi\Lambda}\exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)\left(\frac{\rho R - \cos\theta}{1-\rho^2}\right)(\kappa - t)\Big\rangle_z \tag{114}$$

So together we have three saddle point equations:

$$\frac{R-\rho\cos\theta}{\sin^2\theta} = \frac{\alpha}{\pi\Lambda}\left\langle \int_{-\infty}^{\kappa} dt\ \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)(\kappa-t)\right\rangle_z \tag{115}$$

$$1-\frac{\rho^2+R^2-2\rho R\cos\theta}{\sin^2\theta} = 2\alpha\left\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)(\kappa-t)^2\right\rangle_z \tag{116}$$

$$\frac{\rho-R\cos\theta}{\sin^2\theta} = 2\alpha\left\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)\left(\frac{z-\rho t}{1-\rho^2}\right)(\kappa-t)\right. \tag{117}$$

$$\left. + \frac{1}{2\pi\Lambda}\exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)\left(\frac{\rho R-\cos\theta}{1-\rho^2}\right)(\kappa-t)\right\rangle_z \tag{118}$$

Where

$$\Lambda = \sqrt{\sin^2\theta - R^2 - \rho^2 + 2\rho R\cos\theta}, \tag{119}$$

$$\Gamma(t,z) = z(\rho R-\cos\theta) - t(R-\rho\cos\theta), \tag{120}$$

$$\Delta(t,z) = z^2\left(\rho^2+\cos^2\theta-2\rho R\cos\theta\right) + 2tz(R\cos\theta-\rho) + t^2\sin^2\theta. \tag{121}$$

Solving these equations numerically yields an excellent fit to numerical simulations on structured data (Fig. 2**BCD**).

B Model training method details & dataset information

Perceptron in the teacher-student setting All code to reproduce these simulations can be found at: `https://colab.research.google.com/drive/1in35C6jh7y_ynwuWLBmGOWAgmUgpl8dF?usp=sharing`. Perceptrons were trained on a synthetic dataset of P examples $\{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\dots,P}$, where $\mathbf{x}^\mu \sim \mathcal{N}(0, I_N)$ are i.i.d. zero mean unit variance random Gaussian inputs, and $y^\mu = \text{sign}(\mathbf{T}\cdot x)$ are labels generated by a teacher perceptron $\mathbf{T} \in \mathbb{R}^N$, which was randomly drawn from a uniform distribution on the sphere $\mathbf{T} \sim \text{Unif}(\mathbb{S}^{N-1}(\sqrt{N}))$. For all of our experiments we fixed $N = 200$ and set $P = \alpha N$ where α varied between $10^{0.1}$ and $10^{0.5}$. Each synthetic dataset was pruned to keep a fraction f of the smallest-margin examples, where f varied between 0.1 and 1 in Fig. 1 and between 0.2 and 1 in Figs. 2,3 to match the real-world experiments in Fig. 3. Perceptrons were optimized to find the max-margin separating solution using a standard quadratic programming (QP) algorithm from the CVXPY library (for analysis of the computational complexity of this algorithm see Fig. 8). Results were averaged over 100 independent draws of the teacher and training examples.

ImageNet. ImageNet model training was performed using a standard ResNet-50 through the VISSL library [36] (stable version `v0.1.6`), which provides default configuration files for supervised ResNet-50 training (accessible here; released under the MIT license). Each model was trained on a single node of 8 NVIDIA V100 32GB graphics cards with BATCHSIZE_PER_REPLICA = 256, using the Stochastic Gradient Descent (SGD) optimizer with a base learning rate = 0.1, nesterov momentum = 0.9, and weight decay = 0.001. For our scaling experiments (Fig. 3 and Fig. 9), we trained one model per fraction of data kept (0.1-1.0) for each dataset size. In total, these plot required training 97 models on (potentially a subset of) ImageNet. All the models were trained with matched number of iterations, corresponding to 105 epochs on the full ImageNet dataset. The learning rate was decayed by a factor of 10 after the number of iterations corresponding to 30, 60, 90, and 100 epochs on the full ImageNet dataset.

For our main ImageNet experiments (Fig. 5) we trained one model per fraction of data kept (1.0, 0.9, 0.8, 0.7, 0.6) $\times$ metric (11 metrics in total). In the plot itself, since any variation in the "fraction of data kept = 1.0" setting is due to random variation across runs not due to potential metric differences, we averaged model performances to obtain a single datapoint here (while also keeping track of the variation across models, which is plotted as ± 2 standard deviations). In total, this plot required training 55 models on (potentially a subset of) ImageNet. ImageNet [37] is released under the ImageNet terms of access. It is important to note that ImageNet images are often biased [38, 39]. The SWaV model used to compute our prototypicality metrics was obtained

via `torch.hub.load('facebookresearch/swav:main', 'resnet50')`, which is the original model provided by [31]; we then used the `avgpool` layer's activations.

CIFAR-10 and SVHN. CIFAR-10 and SVHN model training was performed using a standard ResNet-18 through the PyTorch library. Each model was trained on a single NVIDIA TITAN Xp 12GB graphics card with batch size = 128, using the Stochastic Gradient Descent (SGD) optimizer with learning rate = 0.1, nesterov momentum = 0.9, and weight decay = 0.0005. Probe models were trained for 20 epochs each for CIFAR-10 and 40 epochs each for SVHN. Pruning scores were then computed using the EL2Ns metric [9], averaged across 10 independent initializations of the probe models. To evaluate data pruning performance, fresh models were trained from scratch on each pruned dataset for 200 epochs, with the learning rate decayed by a factor of 5 after 60, 120 and 160 epochs.

Data pruning for transfer learning. To assess the effect of pruning downstream finetuning data on transfer learning performance, vision transformers (ViTs) pre-trained on ImageNet21k were fine-tuned on different pruned subsets of CIFAR-10. Pre-trained models were obtained from the timm model library [40]. Each model was trained on a single NVIDIA TITAN Xp 12GB graphics card with batch size = 128, using the Adam optimizer with learning rate = 1e-5 and no weight decay. Probe models were trained for 2 epochs each. Pruning scores were then computed using the EL2Ns metric [9], averaged across 10 independent random seeds. To evaluate data pruning performance, pre-trained models were fine-tuned on each pruned dataset for 10 epochs.

To assess the effect of pruning upstream pretraining data on transfer learning performance, each of the ResNet-50s pre-trained on pruned subsets of ImageNet1k in Fig. 3**D** was fine-tuned on all of CIFAR-10. Each model was trained on a single NVIDIA TITAN Xp 12GB graphics card with batch size = 128, using the RMSProp optimizer with learning rate = 1e-4 and no weight decay. Probe models were trained for 2 epochs each. Pruning scores were then computed using the EL2Ns metric [9], averaged across 10 independent random seeds. To evaluate data pruning performance, pre-trained models were fine-tuned on each pruned dataset for 10 epochs.

C Breaking compute scaling laws via data pruning

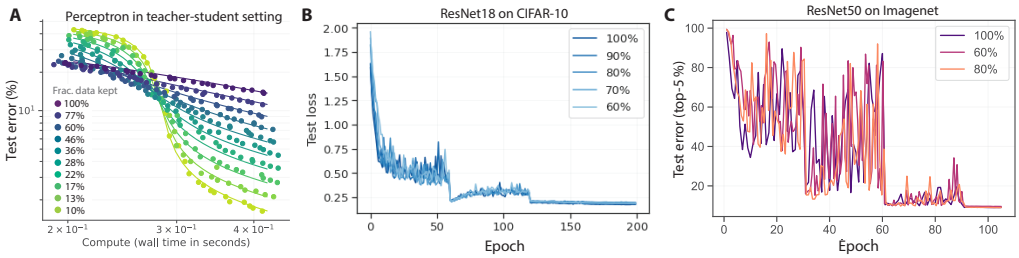


Figure 8: Breaking compute scaling laws via data pruning. **A**, For each experiment in Fig. 1**A** we measure the computational complexity of the quadratic programming optimizer as measured by the wall time to convergence, running on a CPU on a google Colab. **B,C**, ImageNet learning curves.

Do the savings in training dataset size we have identified translate to savings in compute, and can data pruning be used to beat widely observed compute scaling laws [2, 4, 6]? Here we show for the perceptron that data pruning can afford exponential savings in compute, and we provide preliminary evidence that the same is true for ResNets trained on CIFAR-10 and ImageNet. We repeat the perceptron learning experiments in Fig. 1A, keeping track of the computational complexity of each experiment, measured by the time to convergence of the quadratic programming algorithm used to find a max-margin solution (see B for details). Across all experiments, the convergence time T was linearly proportional to α_{prune} with $T = 0.96\alpha_{\text{prune}} + 0.80$, allowing us to replace the x-axis of 1A with compute to produce Fig. 8A, which reveals that data pruning can be used to break compute scaling laws for the perceptron.

Motivated by this, we next investigate whether the convergence time of neural networks trained on pruned datasets depends largely on the number of examples and not their difficulty, potentially

allowing for exponential compute savings. We investigate the learning curves of a ResNet18 trained on CIFAR-10 and a ResNet50 on ImageNet for several different pruning fractions (Fig. 8B). While previous works have fixed the number of iterations [9], here we fix the number of *epochs*, so that the model trained on 60% of the full dataset is trained for only 60% the iterations of the model trained on the full dataset, using only 60% the compute. Nevertheless, we find that the learning curves are strikingly similar across pruning fractions, and appear to converge equally quickly. These results suggest that data pruning could lead to large compute savings in practical settings, and in ongoing experiments we are working to make the analogs of Fig. 8A for ResNets on CIFAR-10 and ImageNet to quantify this benefit.

D Additional scaling experiments

In Fig. 9 we perform additional scaling experiments using the EL2Ns and self-supervised prototypes metrics. In Fig. 10 we give a practical example of a cross over from exponential to power-law scaling when the probe student has limited information about the teacher (here a model trained for only a small number of epochs on SVHN) .

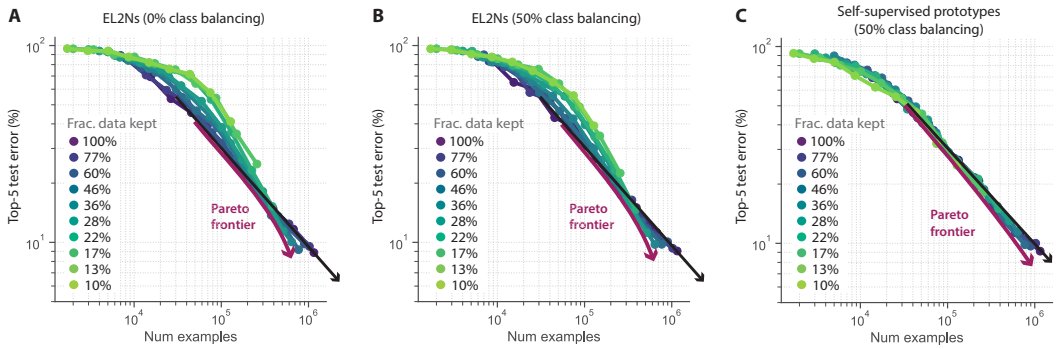


Figure 9: Additional scaling experiments. We reproduce the scaling results on ImageNet in Fig. 3**D** using three additional metrics: two supervised and one self-supervised. Each shows some signatures of breaking power law scaling, although the effect is less dramatic than for the best metric, memorization (Fig. 3**D**). (**A**) EL2Ns with a class balancing fraction of 0% (see App. Section H for details), (**B**) EL2Ns with a class balancing fraction of 50%, and (**C**) Self-supervised prototypes with a class balancing fraction of 50%.

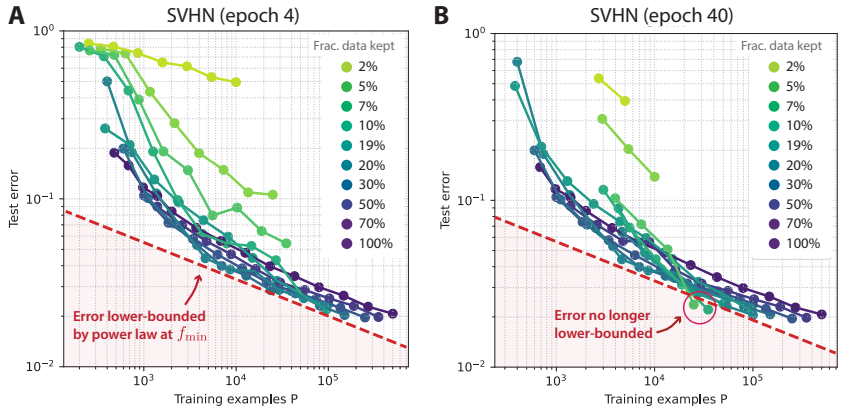


Figure 10: Consistent with a prediction from the perceptron theory (Fig. 2), when SVHN is pruned using a weak metric (a probe trained for only 4 epochs), the learning curve envelope is lower-bounded by a power law at some $f_{\min}$. However, with a stronger pruning metric (a probe trained for 40 epochs), the learning curve can break through this power law to achieve lower generalization error.

E Extremal images according to different metrics

In Fig. 6, we showed extremal images for two metrics (self-supervised prototypes, memorization) and a single class. In order to gain a better understanding of how extremal images (i.e. images that are easiest or hardest to learn according to different metrics) look like for all metrics and more classes, we here provide additional figures. In order to avoid cherry-picking classes while at the same time making sure that we are visualizing images for very different classes, we here show extreme images for classes 100, ..., 500 while leaving out classes 0 and 400 (which would have been part of the visualization) since those classes almost exclusively consist of images containing people (0: tench, 400: academic gown). The extremal images are shown in Figures 11,12,13,14,15,16,17,18.

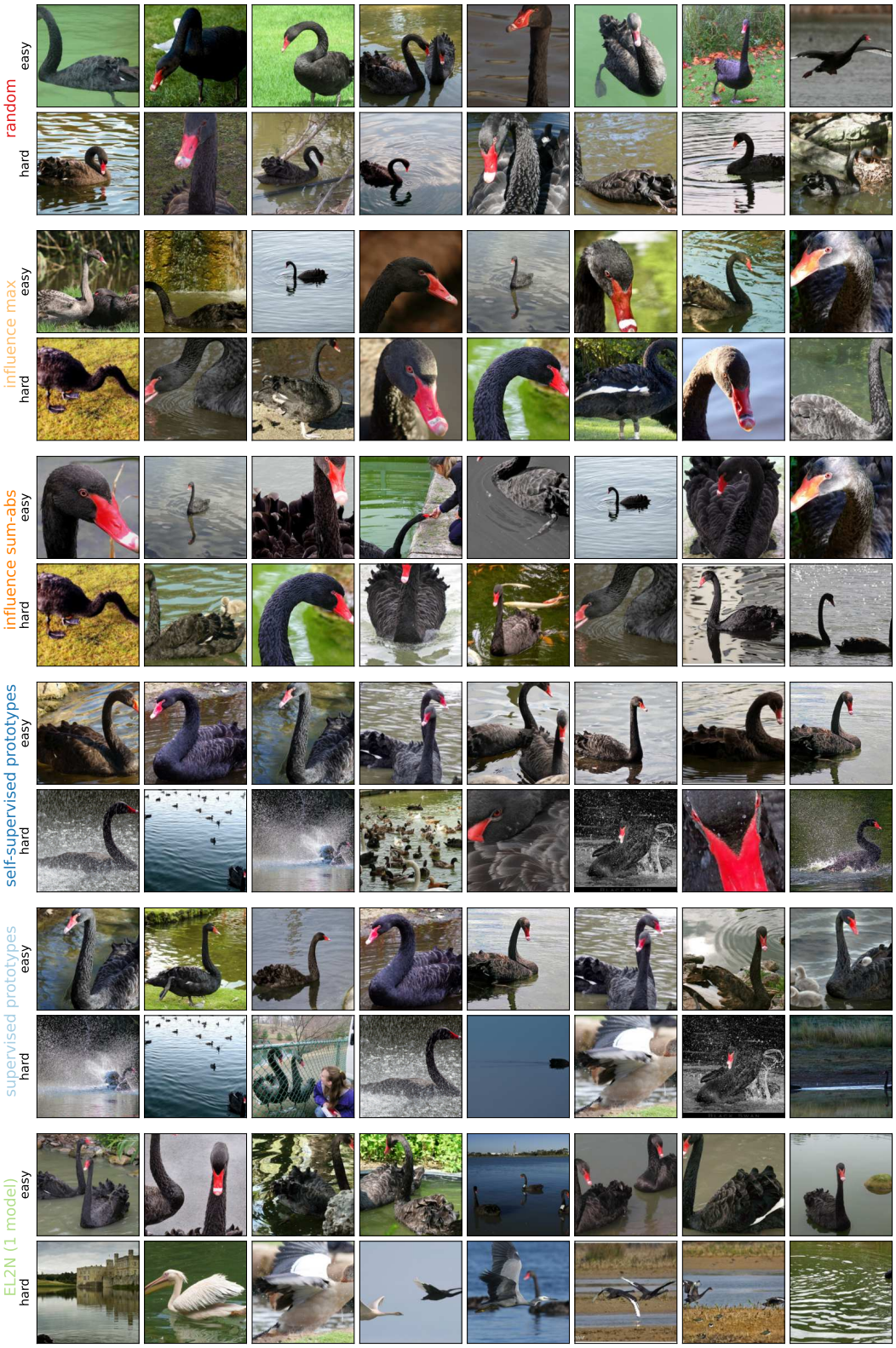


Figure 11: Extreme images according to different metrics for ImageNet class 100 (`black swan`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

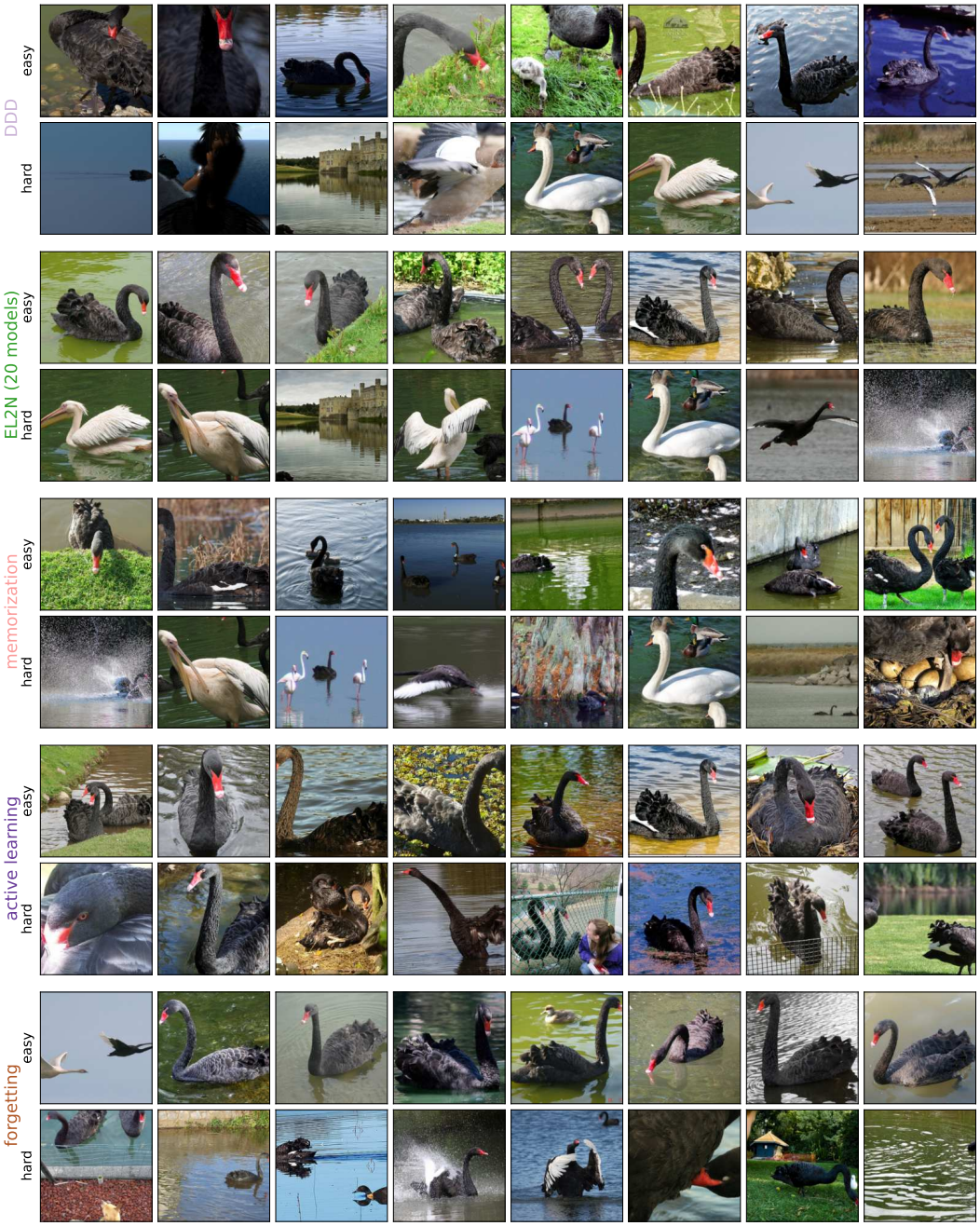


Figure 12: Extreme images according to different metrics for ImageNet class 100 (`black swan`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

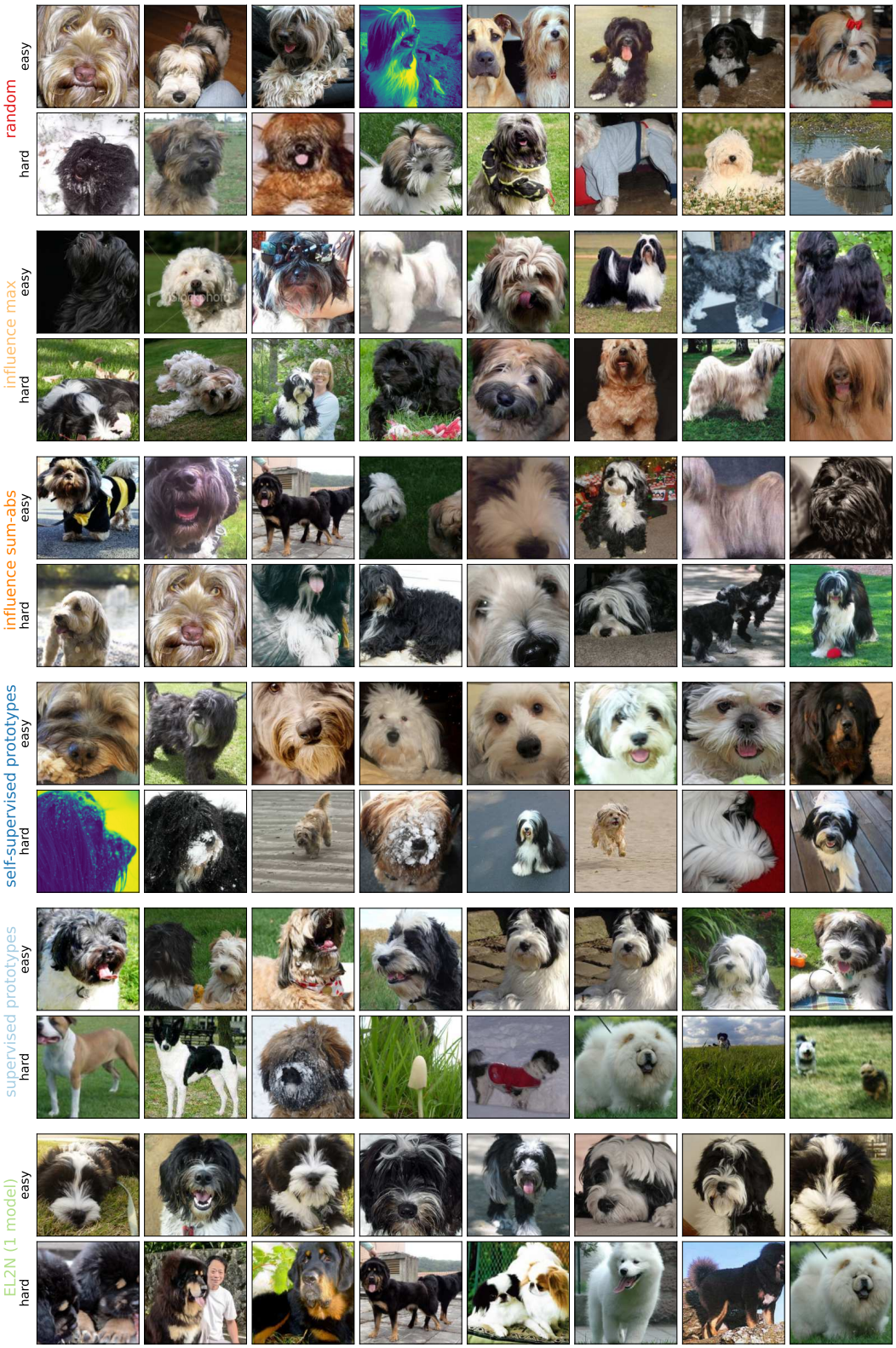


Figure 13: Extreme images according to different metrics for ImageNet class 200 (`Tibetan terrier`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

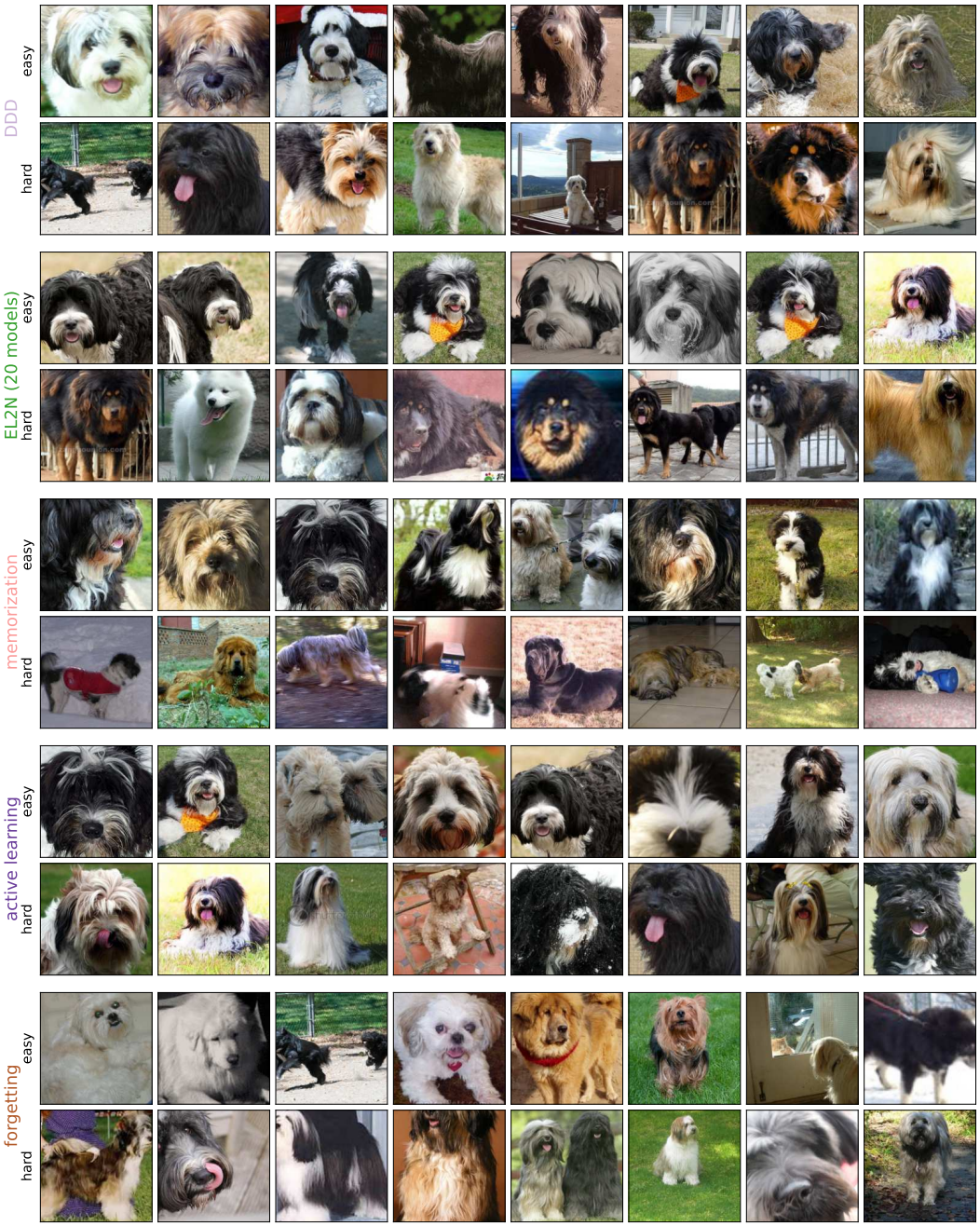


Figure 14: Extreme images according to different metrics for ImageNet class 200 (`Tibetan terrier`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

Figure 15: Extreme images according to different metrics for ImageNet class 300 (`tiger beetle`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

Figure 16: Extreme images according to different metrics for ImageNet class 300 (`tiger beetle`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

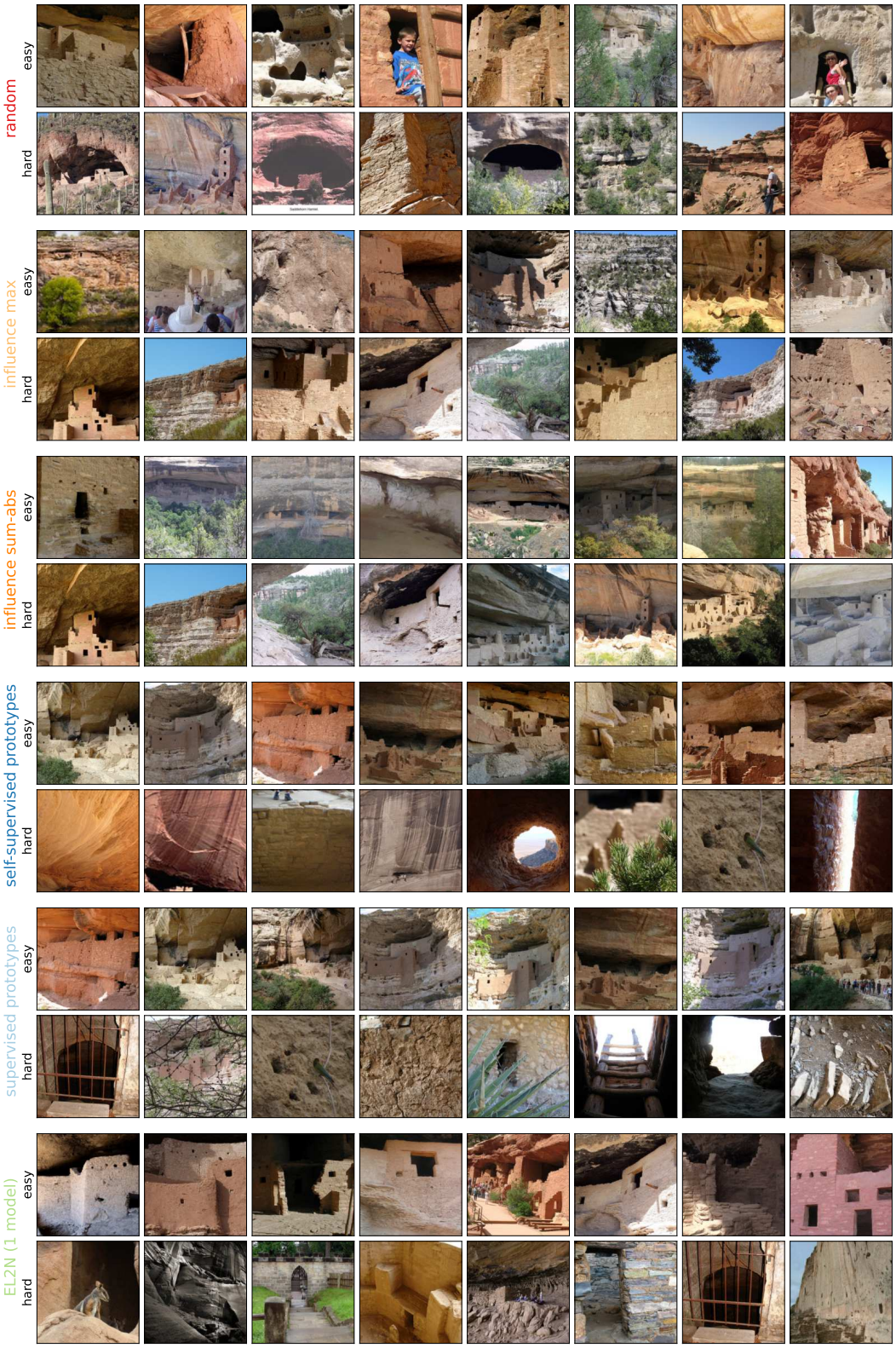


Figure 17: Extreme images according to different metrics for ImageNet class 500 (`cliff dwelling`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

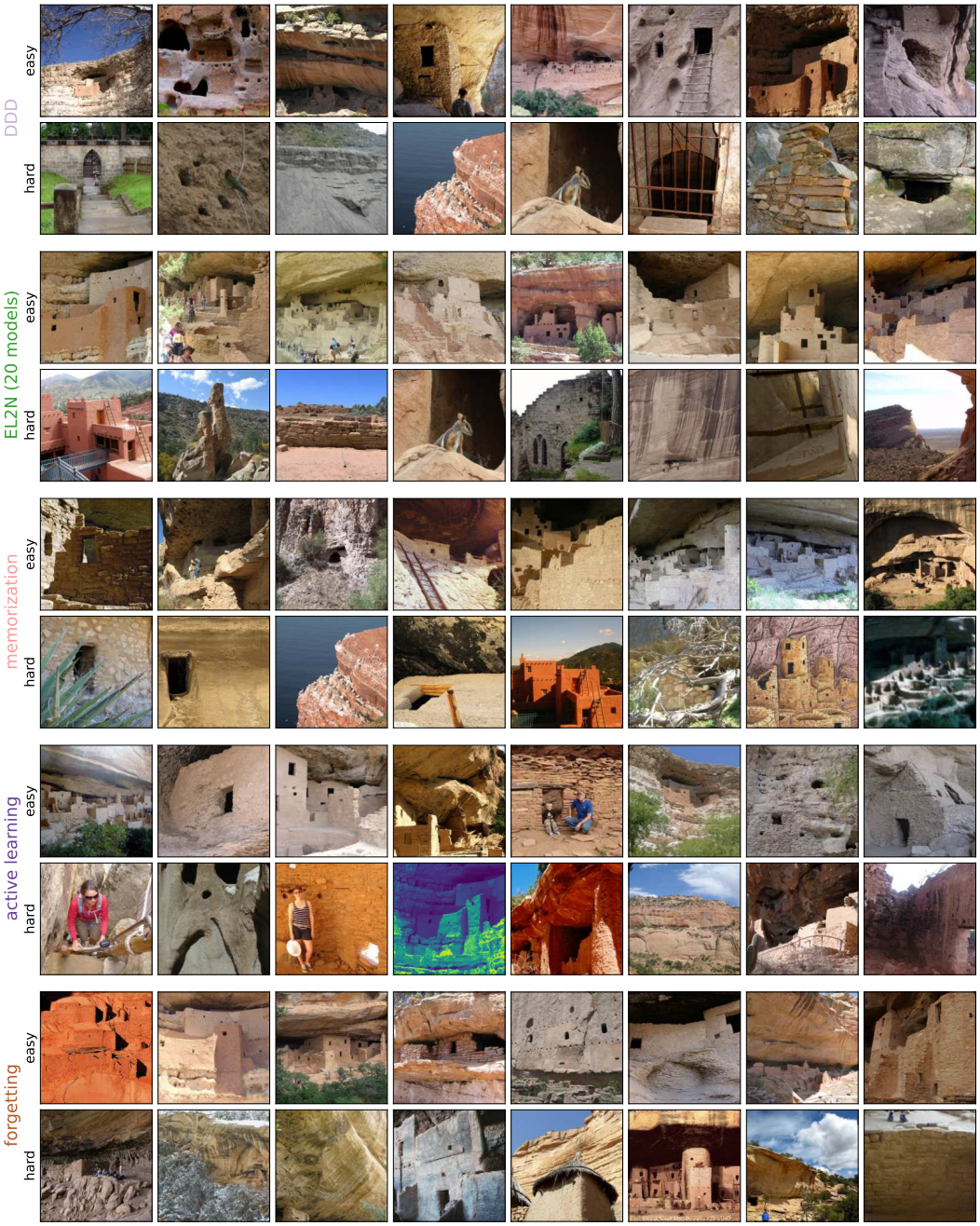


Figure 18: Extreme images according to different metrics for ImageNet class 500 (`cliff dwelling`). For each metric, the top row shows images that are ranked as "easy" (most pruneable) according to the metric, and the bottom row shows images that are ranked as "hard" (least pruneable).

F Impact of number of clusters k on self-supervised prototypes

Our self-supervised prototype metric is based on k-means clustering, which has a single hyperparameter k. By default and throughout the main paper, we set $k = 1000$, corresponding to the number of classes in ImageNet. Here, we investigate other settings of k to understand how this hyperparameter impacts performance. As can be seen in Table 1, k does indeed have an impact on performance, and very small values for k (e.g. $k < 10$) as well as very large values for k (e.g. $k = 50,000$) both lead to performance impairments. At the same time, performance is relatively high across very different in-between settings for k. In order to assess these results, it may be important to keep in mind that $\pm 0.54\%$ corresponds to plus/minus 2 standard deviations of performance when simply training the same model multiple times (with different random initialization). Overall, these results suggest that if the number of clusters k deviates at most by one order of magnitude from the number of classes in the dataset (for ImageNet-1K), the exact choice of k does not matter much.

k	1	5	10	50	100	200	400	600	800	1K	5K	10K	50K
acc	88.35	89.09	88.76	89.04	89.48	90.27	89.85	89.96	90.44	90.56	90.33	90.57	88.67

Table 1: Performance (top-5 accuracy, denoted as acc) when pruning away 20% of ImageNet according to our self-supervised prototype metric as a function of the number of prototypes k (hyperparameter used for self-supervised clustering indicating the number of clusters). Results based on training a ResNet-50 architecture with VISSL without class balancing.

G Impact of ensemble prototypes

The self-supervised prototypes metric is based on k-means clustering in the embedding space of a self-supervised (=SSL) model. Since even otherwise identical models trained with different random seeds can end up with somewhat different embedding spaces, we here investigated how the performance of our self-supervised prototypes metric would change when averaging the scores derived from five models, instead of just using a single model's score. The results, shown in Table 2, indicate that ensembling the self-supervised prototype scores neither improves nor hurts performance. This is both good and bad news: Bad news since naturally any improvement in metric development leads to better data efficiency; on the other hand this is also good news since ensembles increase the computational cost of deriving the metric—and this suggests that ensembling is not necessary to achieve the performance we achieved (unlike in other methods such as ensemble active learning).

Fraction of data kept	0.9	0.8	0.7
score from single model	90.03	90.10	89.38
score from ensemble model	89.93	90.40	89.38

Table 2: Performance (top-5 acuracy) when pruning away 10%, 20% or 30% of ImageNet based on the self-supervised prototype metric derived from either a single SSL model, or an ensemble of SSL models. Supervised model training on the pruned dataset was performed with a ResNet-50 architecture trained using VISSL without class balancing. Overall, we do not observe any performance difference between the two settings: the differences are still well within a single standard deviation of training multiple models with identical settings. In order to reduce the influence of random variations, each data point in this table shows the average of four independent runs.

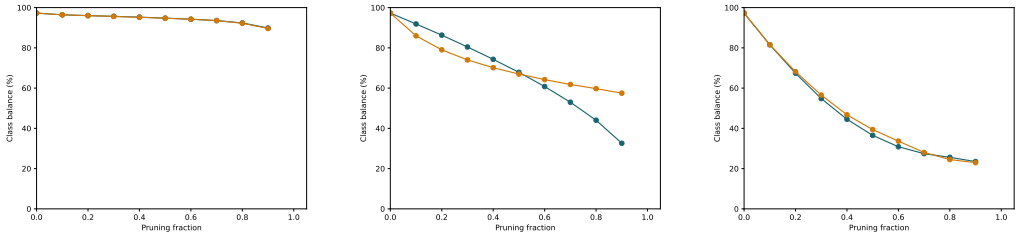
H Relationship between pruning and class (im-)balance

Motivation. It is well-known that strong class imbalance in a dataset is a challenge that needs to be addressed. In order to understand the effect of pruning on class (im-)balance, we quantified this relationship. For context, if pruning according to a certain metric preferentially leads to discarding most (or even all) images from certain classes, it is likely that the performance on those classes will drop as a result if this imbalance is not addressed.

Class imbalance metric: pruning amplifies class imbalance. Since a standard measure of class (im-)balance—dividing the number of images for the majority class by the number of images for the minority class—is highly sensitive to outliers and discards information about the 998 non-extreme ImageNet classes, we instead calculated a class balance score $b \in [0\%, 100\%]$ as the average class imbalance across any two pairs of classes by computing the expectation over taking two random classes, and then computing how many images the minority class has in proportion to the majority class. For instance, a class balance score of 90% means that on average, when selecting two random classes from the dataset, the smaller of those two classes contains 90% of the number of images of the larger class (higher=better; 100% would be perfectly balanced).

In Fig. 19, we observe that dataset pruning strongly increases class imbalance. This is the case both when pruning away easy images and when pruning away hard images, and the effect occurs for all pruning metrics except, of course, for random pruning. Class imbalance is well-known to be a challenge for deep learning models when not addressed properly [41]. The cause for the amplified class imbalance is revealed when looking at class-conditional differences of metric scores (Figs. 21 and 22): The histograms of the class-conditional score distributions show that for many classes, most (if not all) images have very low scores, while for others most (if not all) images have very high scores. This means that as soon as the lowest / highest scoring images are pruned, certain classes are pruned preferentially and thus class imbalance worsens.

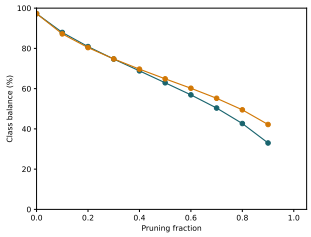
We thus use 50% class balancing for our ImageNet experiments. This ensures that every class has at least 50% of the images that it would have when pruning all classes equally (essentially providing a fixed floor for the minimum number of images per class). This simple fix is an important step to address and counteract class imbalance, although other means could be used as well; and ultimately one would likely want to use a self-supervised version of class (or cluster) balancing when it comes to pruning large-scale unlabeled datasets. For comparison purposes, the results for ImageNet pruning *without* class balancing are shown in supplementary Fig. 20.



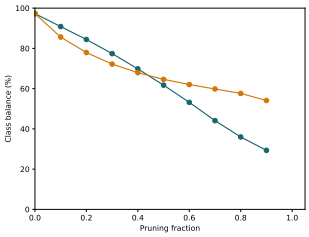
(a) random

(b) influence max

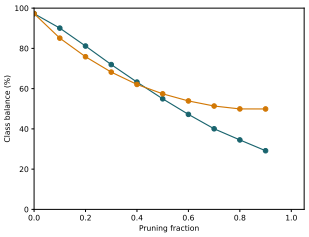
(c) influence sum-abs



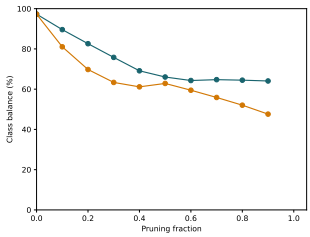
(d) self-supervised prototypes



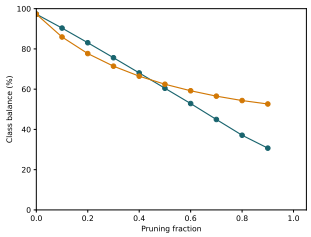
(e) supervised prototypes



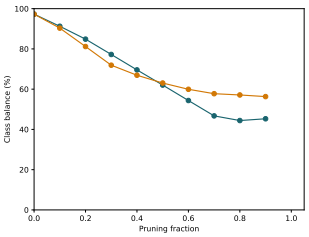
(f) EL2N (1 model)



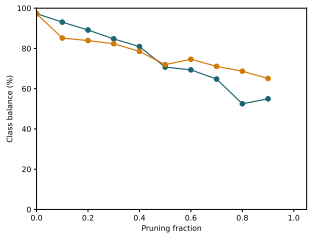
(g) DDD



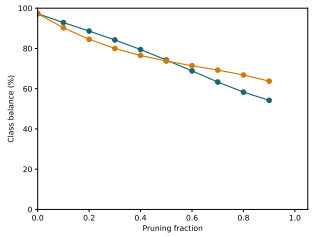
(h) EL2N (20 models)



(i) memorization



(j) active learning



(k) forgetting

Figure 19: Pruning amplifies class imbalance. With larger pruning fractions, class balance decreases—both when pruning "easy" images (turquoise) and when pruning "hard" images (orange). This effect occurs for all pruning metrics except for random pruning (top left). For details on the class imbalance metric see Appendix H. Pruning fraction refers to the fraction of data pruned, from 0 (no pruning) to 0.9 (keeping only 10% of the data).

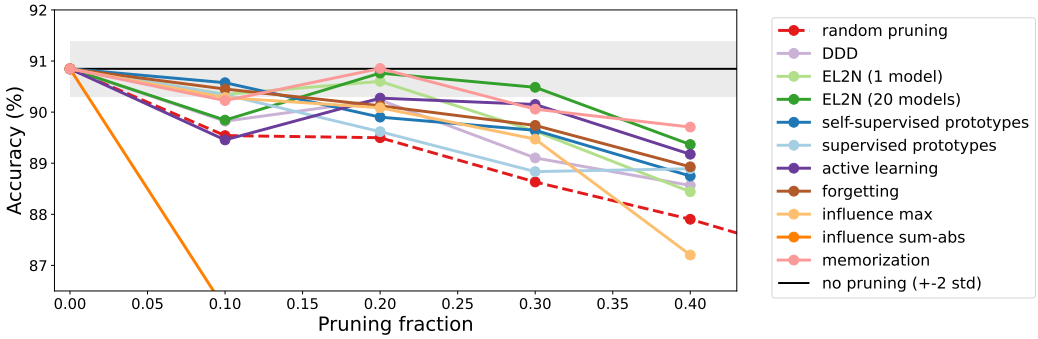


Figure 20: ImageNet-1K pruning results for different metrics, compared against random pruning. 'Pruning fraction' refers to the fraction of the dataset that is pruned away. Results obtained without any class balancing are worse than results with 50% class balancing (Fig. 5BC), confirming the finding that vanilla pruning amplifies class imbalance.

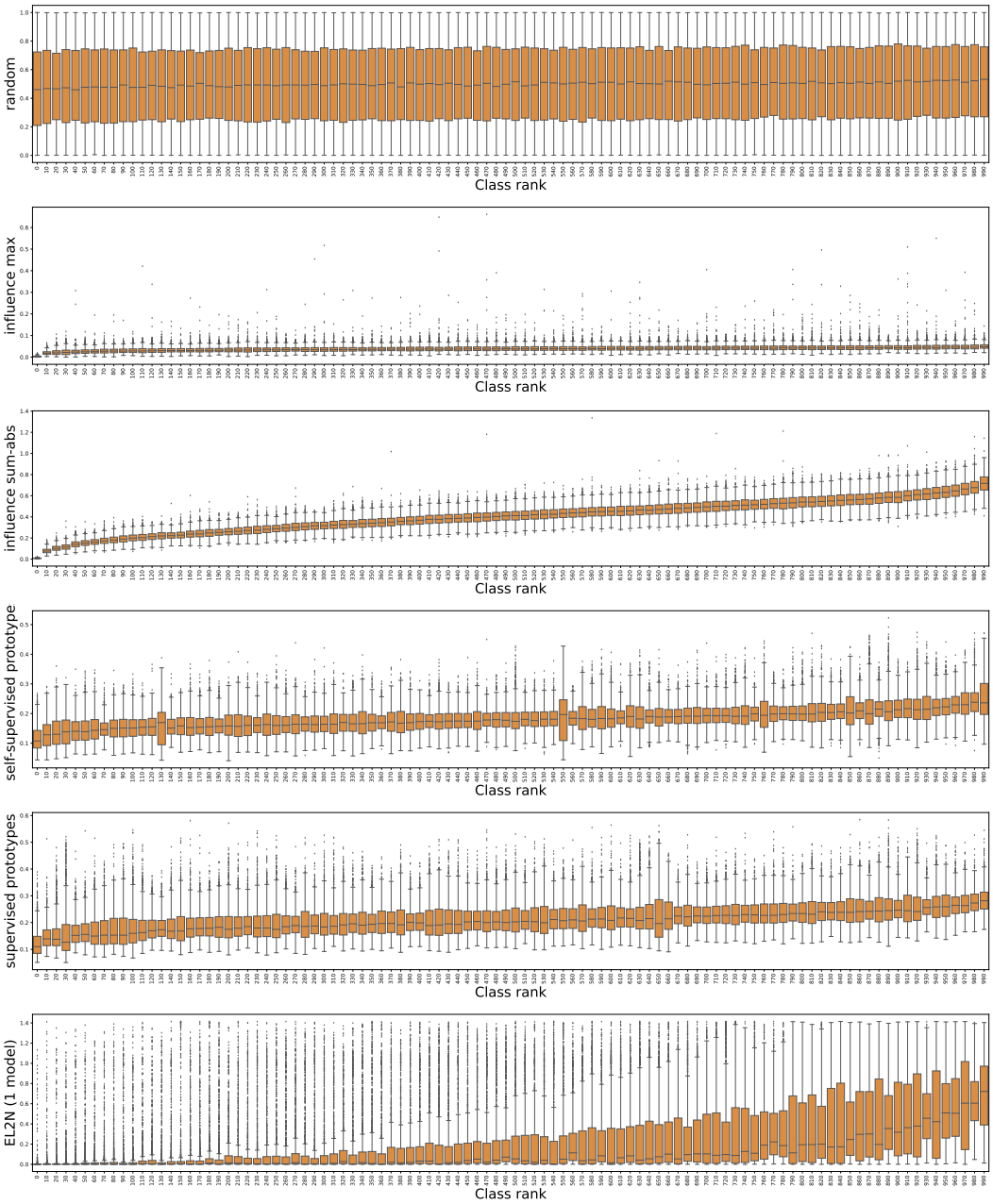


Figure 21: How do ImageNet metric scores differ across classes? Class-conditional score distribution histograms across metrics. For the purpose of visualization, only every 10th class is shown.

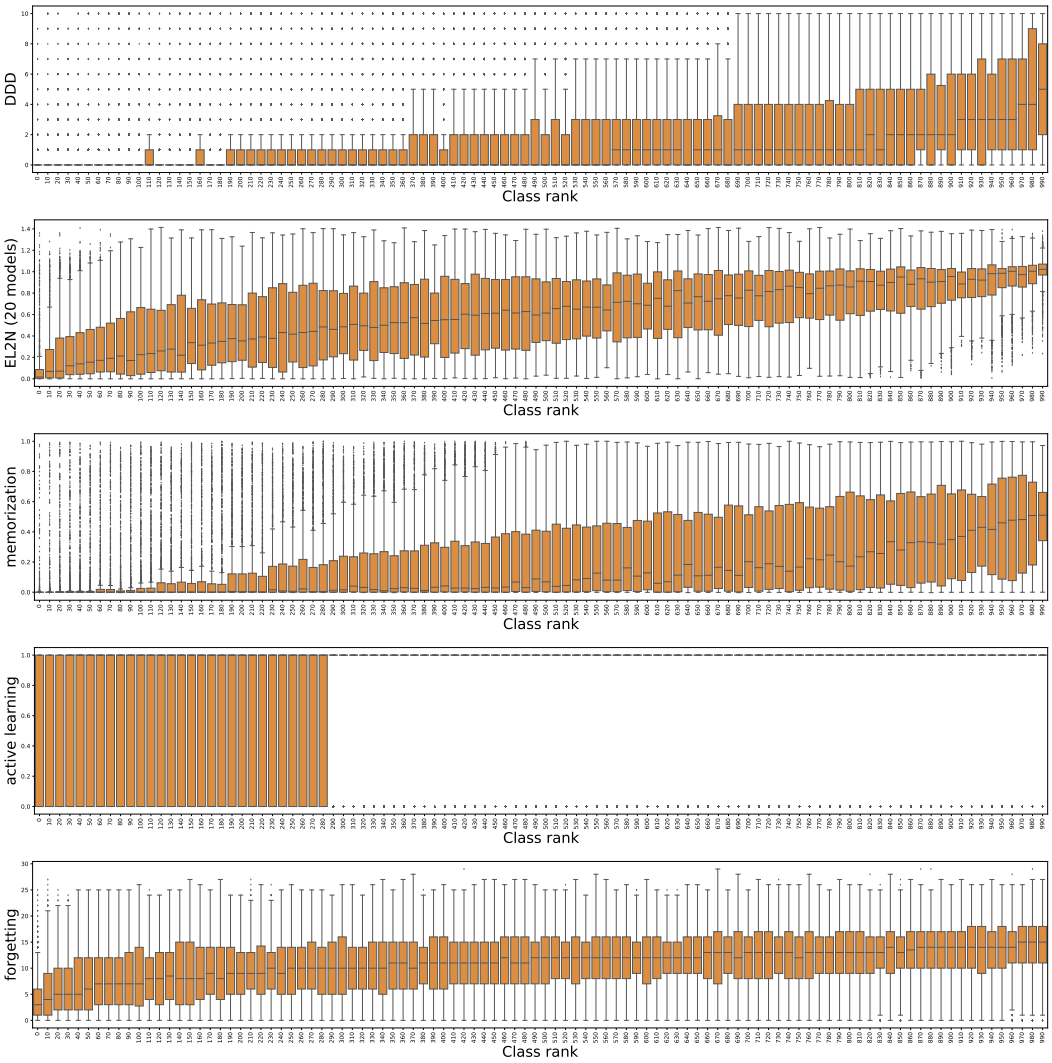
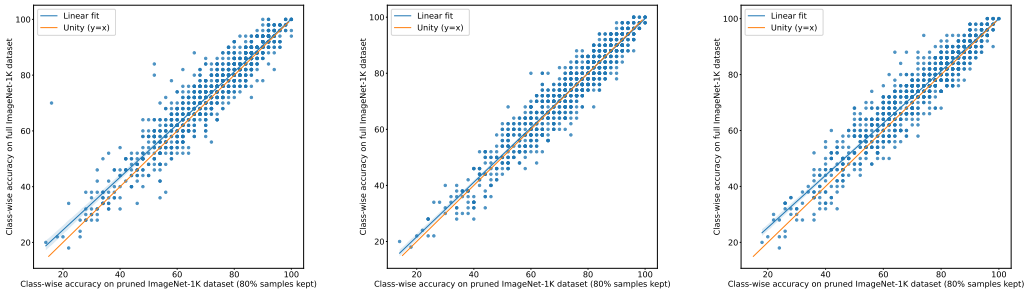


Figure 22: How do ImageNet metric scores differ across classes (continued)? Class-conditional score distribution histograms across metrics. For the purpose of visualization, only every 10th class is shown. Please note that the active learning plot is to be taken with a grain of salt since the authors provided a binary score (included/excluded) for ImageNet corresponding to 80% "important" images, thus the scores are either zero or one and a boxplot fit does not apply here.

I Effect of pruning on class-conditional accuracy and fairness

In order to study the effect of dataset pruning on model fairness, at least with respect to specific ImageNet classes, we compared the class-conditional accuracy of a ResNet-50 model trained on the full ImageNet dataset, versus that of the same model trained on an 80% subset obtained after pruning. We used two supervised pruning metrics (EL2N, memorization) and one self-supervised pruning metric (self-supervised prototypes) for obtaining the pruned dataset. In all three cases, and across all 1000 classes, we found that the class-conditional accuracy of the model trained on a pruned subset of the dataset remains quite similar to that of the model trained on the full dataset (Fig. 23). However, we did notice a very small reduction in class-conditioned accuracy for ImageNet classes that were least accurately predicted by models trained on the entire dataset (blue lines slightly above red unity lines when class-conditioned accuracy is low). This suggests that pruning yields a slight systematic reduction in the accuracy of harder classes, relative to easier classes, though the effect is small.

While we have focused on fairness with respect to individual ImageNet classes, any ultimate test of model fairness should be conducted in scenarios that are specific to the use case of the deployed model. Our examination of the fairness of pruning with respect to individual ImageNet classes constitutes only an initial foray into an exploration of fairness, given the absence of any specific deployment scenario for our current models other than testing them on ImageNet. We leave a full exploration of fairness in other deployment settings for future work.



(a) EL2N (20 models) (b) Memorization (c) Self-supervised prototypes

Figure 23: The effect of data pruning on ImageNet test accuracy on individual classes. For all 3 plots, each point corresponds to an ImageNet class, and for all classes, the class specific test accuracy when training on the entire dataset (y-axis) is plotted against the test accuracy when training on the pruned dataset (x-axis).

J Interaction between data pruning and training duration

Throughout the main paper, our ImageNet experiments are based on a setting where the number of training *epochs* is kept constant (i.e. we train the same number of epochs on the smaller pruned dataset as on the larger original dataset). This means that data pruning directly reduces the number of *iterations* required to train the model specifically by reducing the size of the dataset. However, this simultaneously places two constraints on model performance: not only training on a smaller data set, but also training for fewer iterations.

We therefore investigate how model performance changes if we train longer, as quantified by a *matched iterations factor*. A matched iterations factor of 0 corresponds to the default setting used in the paper of training for the same number of epochs (so that a smaller dataset means proportionally fewer training iterations). In contrast a matched iterations factor of 1 corresponds to training on the smaller pruned dataset for a number of iterations equal to that when training on the larger initial dataset (e.g. when pruning away 50% of the dataset one would train twice as long to match the number of iterations of a model trained on 100% of the dataset). Otherwise the matched iterations factor reflects a linear interpolation in the number of training iterations as the factor varies between 0 and 1.

The results are shown in Table 3 and indicate that training longer does indeed improve performance slightly; however, a matched iterations factor of around 0.4–0.6 may already be sufficient to reap the full benefit. Any matched iterations factor strictly smaller than 1.0 comes with reduced training time compared to training a model on the full dataset.

matched iterations factor	0.0	0.2	0.4	0.6	0.8	1.0
ImageNet top-5 accuracy	90.20	90.41	90.48	90.56	90.45	90.50

Table 3: Comparing different settings for training longer when pruning: Performance (top-5 acuracy) when pruning away 20% of ImageNet based on our self-supervised prototype metric. Supervised model training on the pruned dataset performed with a ResNet-50 architecture trained using VISSL without class balancing. Performance tends to increase when training longer (i.e. with a larger matched iterations factor). Interestingly, the benefit of training longer may already be achieved with a matched iterations factor of around 0.4–0.6.

K Out-of-distribution (OOD) analysis of dataset pruning

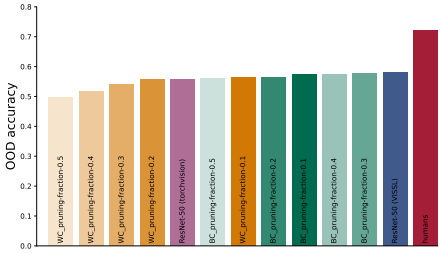
Pruning changes the data regime that a model is exposed to. Therefore, a natural question is how this might affect desirable properties beyond IID performance like fairness (see Appendix I) and out-of-distribution, or OOD, performance which we investigate here. To this end, we use the model-vs-human toolbox [42] based on data and analyses from [43, 44, 45, 46, 47]. This toolbox is comprised of 17 different OOD datasets, including many image distortions and style changes.

In Figure 24a, OOD accuracies averaged across those 17 datasets are shown for a total of 12 models. These models all have a ResNet-50 architecture [48]. Two baseline models are trained on the full ImageNet training dataset, one using torchvision (purple) and the other using VISSL (blue). Human classification data is shown as an additional reference in red. The remaining 10 models are VISSL-trained on pruned versions of ImageNet using pruning fractions in {0.1, 0.2, 0.3, 0.4, 0.5} and our self-supervised prototype metric. A pruning fraction of 0.3 would correspond to "fraction of data kept = 0.7", i.e. to training on 70% of ImageNet while discarding the other 30%. We investigated two different settings: discarding easy examples (the default used throughout the paper), which is denoted as "Best Case" (or BC) in the plots; and the reverse setting, i.e. discarding hard examples denoted as Worst Case, or WC. (These terms should be taken with a grain of salt; examples are only insofar best- or worst case examples as predicted by the metric, which itself is in all likelihood far less than perfect.)

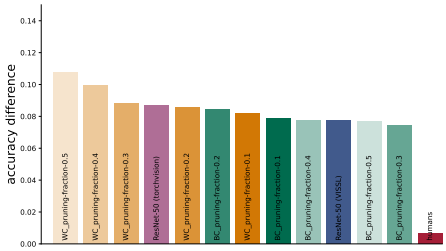
The results are as follows: In terms of OOD accuracy (Figure 24a), best-case pruning in green achieves very similar accuracies to the most relevant baseline, the blue ResNet-50 model trained via VISSL. This is interesting since oftentimes, OOD accuracies closely follow IID accuracies except for a constant offset [49], and we know from Figure 5 that the self-supervised prototype metric has a drastic performance impairment when pruning away 40% of the data, yet the model "BC_pruning-fraction-0.4" still achieves almost the same OOD accuracy as the baseline trained on the full dataset. **The core take-away is: While more analyses would be necessary to investigate whether pruning indeed consistently *helps* on OOD performance, it seems safe to conclude that it does not *hurt* OOD performance on the investigated datasets compared to an accuracy-matched baseline.** (The control setting, pruning away hard examples shown in orange, leads to much lower IID accuracies and consequently also lower OOD accuracies.) For reference, the numerical results from Figure 24a are also shown in Table 4.

Figures 24b, 24c and 24d focus on a related question, the question of whether models show human-like behavior on OOD datasets. Figure 24b shows that two models pruned using our self-supervised prototype metric somewhat more closely match human accuracies compared to the baseline in blue; Figures 24c and 24d specifically focus on image-level consistency with human responses. In terms of overall consistency (c), the baseline scores best; in terms of error consistency pruned models outperform the VISSL-trained baseline. For details on the metrics we kindly refer the interested reader to [42]. Numerical results are again also shown in a Table (Table 5).

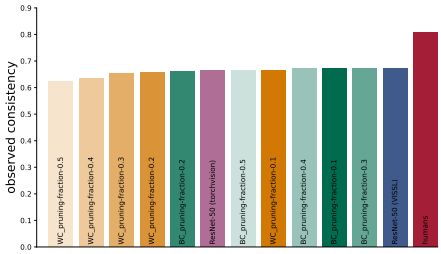
Finally, in Figure 25 we observe that best-case pruning leads to slightly higher shape bias as indicated by green vertical lines plotting the average shape bias across categories, which are shifted to the left of the baseline; while worst-case pruning in orange is shifted to the right. An outlier is the torchvision-trained model in purple with a very strong texture bias; we attribute this to data augmentation differences between VISSL and torchvision training.



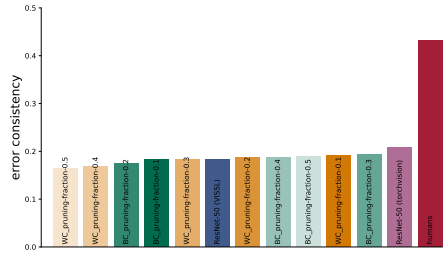
(a) OOD accuracy (higher = better).



(b) Accuracy difference (lower = better).



(c) Observed consistency (higher = better).



(d) Error consistency (higher = better).

Figure 24: OOD benchmark results for different models, aggregated over 17 datasets. All models have a ResNet-50 architecture [48]. Two baseline models are trained on the full ImageNet training dataset, one using torchvision (purple) and the other using VISSL (blue). Human comparison data is shown in red.

Table 4: Benchmark table of model results for highest out-of-distribution robustness.

model	OOD accuracy ↑	rank ↓
ResNet-50 (VISSL)	**0.582**	**1.000**
BC_pruning-fraction-0.3	0.578	2.000
BC_pruning-fraction-0.4	0.574	3.000
BC_pruning-fraction-0.1	0.574	4.000
BC_pruning-fraction-0.2	0.565	5.000
WC_pruning-fraction-0.1	0.565	6.000
BC_pruning-fraction-0.5	0.560	7.000
ResNet-50 (torchvision)	0.559	8.000
WC_pruning-fraction-0.2	0.556	9.000
WC_pruning-fraction-0.3	0.540	10.000
WC_pruning-fraction-0.4	0.516	11.000
WC_pruning-fraction-0.5	0.498	12.000

Table 5: Benchmark table of model results for most human-like behaviour. The three metrics "accuracy difference" "observed consistency" and "error consistency" (plotted in Figure 24) each produce a different model ranking.

model	accuracy diff. ↓	obs. consistency ↑	error consistency ↑	mean rank ↓
BC_pruning-fraction-0.3	**0.075**	0.671	0.194	**1.667**
ResNet-50 (VISSL)	0.077	**0.671**	0.184	3.667
BC_pruning-fraction-0.5	0.077	0.666	0.190	4.000
BC_pruning-fraction-0.4	0.078	0.670	0.187	4.333
WC_pruning-fraction-0.1	0.082	0.666	0.191	4.667
ResNet-50 (torchvision)	0.087	0.665	**0.208**	5.667
BC_pruning-fraction-0.1	0.079	0.671	0.182	5.667
WC_pruning-fraction-0.2	0.086	0.658	0.186	7.667
BC_pruning-fraction-0.2	0.084	0.660	0.175	8.333
WC_pruning-fraction-0.3	0.088	0.652	0.183	9.333
WC_pruning-fraction-0.4	0.099	0.635	0.169	11.000
WC_pruning-fraction-0.5	0.108	0.623	0.165	12.000

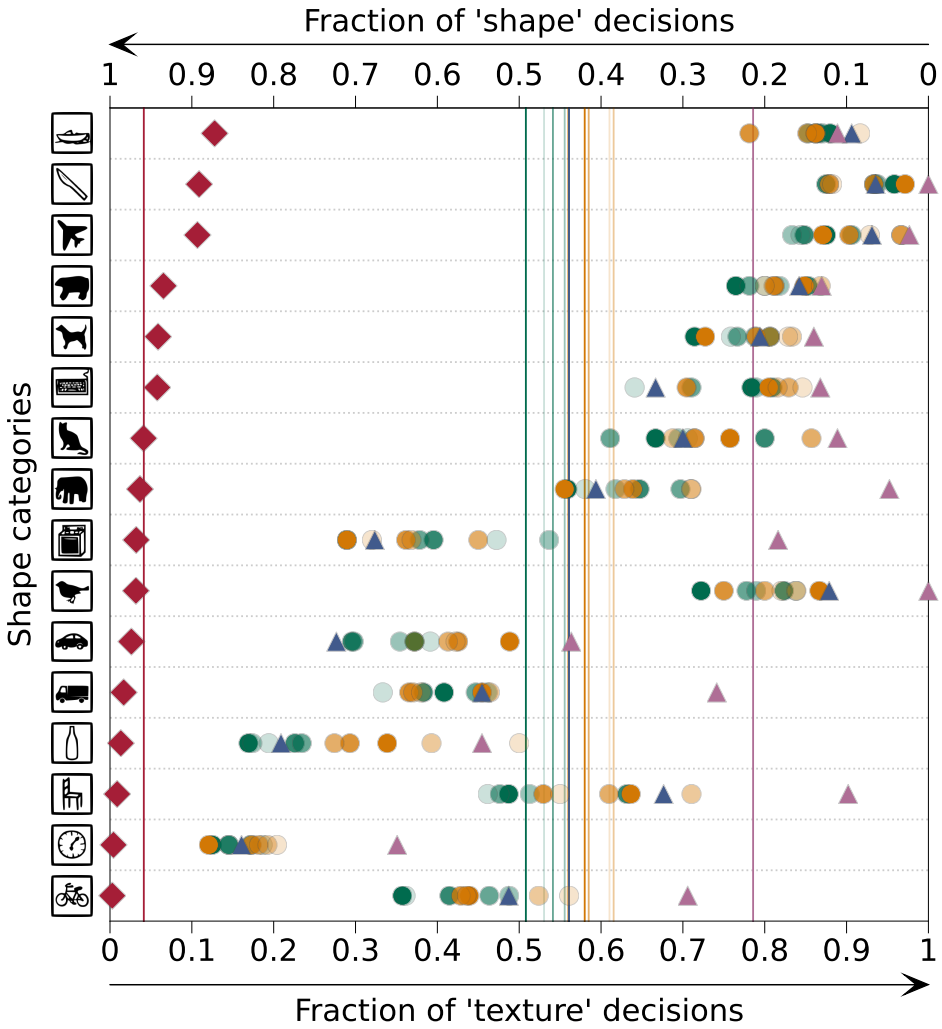


Figure 25: Shape vs. texture bias [45]: category-level plot. Horizontal lines indicate average shape/texture bias; values to the left lean towards a shape bias while values to the right lean towards a texture bias. For details on the plot see [45]; model colors are identical to Figure 24.