# Bad Students Make Great Teachers: Active Learning Accelerates Large-Scale Visual Understanding

Talfan Evans[1,*] Shreya Pathak[1,*] Hamza Merzic[1,2,*]
Jonathan Schwarz[1,†] Ryutaro Tanno[1] Olivier J. Hénaff[1,*]

[1]Google DeepMind [2]University College London

**Abstract.** Power-law scaling indicates that large-scale training with uniform sampling is prohibitively slow. Active learning methods aim to increase data efficiency by prioritizing learning on the most relevant examples. Despite their appeal, these methods have yet to be widely adopted since no one algorithm has been shown to a) generalize across models and tasks b) scale to large datasets and c) yield overall FLOP savings when accounting for the overhead of data selection. In this work we propose a method which satisfies these three properties, leveraging small, cheap proxy models to estimate "learnability" scores for datapoints, which are used to prioritize data for training much larger models. As a result, models trained using our methods – *ClassAct* and *ActiveCLIP* – require 46% and 51% fewer training updates and up to 25% less total computation to reach the same performance as uniformly-trained visual classifiers on JFT and multimodal models on ALIGN, respectively. Finally, we find our data-prioritization scheme to be complementary with recent data-curation and learning objectives, yielding a new state-of-the-art in several multimodal transfer tasks.

## 1 Introduction

Power-law scaling for vision and language models [19,43] indicates that incremental improvements in model performance require orders of magnitude increases in computation. One of the key features of these empirical power-laws is that training data is sampled uniformly. In contrast, active data selection prioritizes computation on the data that maximally contributes to task performance [21,24,34], with the ultimate goals of improving data efficiency and reducing the cost of training. However, active data selection has yet to become a mainstay of large model training, since no existing algorithm satisfies the three properties of being a) robust to the choice of model and training task, b) scalable to large datasets and architectures, and c) more compute efficient end-to-end than training with uniform samples.

*Equal technical contribution. Email correspondence to <talfan@deepmind.com> and <henaff@deepmind.com>. †Current affiliation: Harvard University, work done while at Google DeepMind.

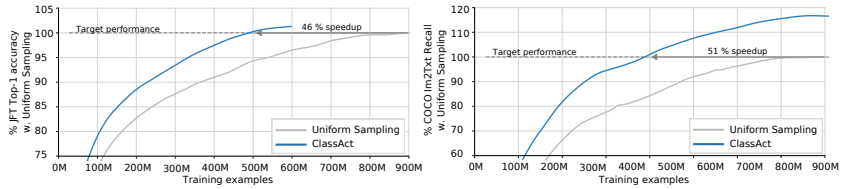


**Fig. 1: Active learning accelerates large-scale visual understanding**. For large-scale classification and multimodal learning tasks, prioritised training on data selected using our active selection methods *ClassAct* (**left**) and *ActiveCLIP* (**right**) requires significantly fewer updates to reach the final performance of uniform training.

In the first instance, data selection based on hand-engineered filters (e.g. removing incorrectly shaped images or that only contain a single colour [2]) can trivially improve training efficiency at minimal computational overhead. However, such heuristics are limited in their effectiveness by the expertise of the human designer, and are not guaranteed to transfer to training of different models, data modalities, or tasks and incur significant effort to develop and tune.

In contrast, model-based curation methods, which use the loss of the model itself to score examples, have shown promise by focusing training on 'hard' and omitting 'easy' data [35], but also the opposite (to exclude noise or other low-quality data) in both language modeling [13] and multimodal learning [15]. However, these methods often spend as much computation on the curation of datasets as is gained from subsequent pretraining, making them less compute-efficient than training on uniformly-sampled data. Finally, while several compute-efficient methods have been successfully deployed at small scale [8], they generally do not scale to even medium-sized datasets such as ImageNet.

In this work, we propose an algorithm that satisfies the three properties of generality, scalability, and *compute-positivity*. The proposed framework uses small proxy models to compute *learnability* scores for candidate training data, resulting in significant training efficiency gains at an almost negligible overhead over standard uniform-training. We test two instantiations of this framework, *ClassAct* and *ActiveCLIP* / *ActiveSigLIP*, for large scale classification and multimodal pretraining, respectively. Our findings are summarized below:

**Benchmarking heuristics for large-scale pretraining:** We investigate loss- and learnability-based prioritization [15, 27, 35] for large-scale classification and find that pretrained reference models are an essential component for accelerating learning, producing efficiency gains of up to 46%.

**Generalizing selection policies across model scale:** Secondly, we show that smaller models act as effective proxies for much larger ($\sim$1000$\times$) models in the context of learnability but *not* loss-based scoring, resulting in *compute-positive* gains of up to 25% over uniform training, a first in the context of large-scale pretraining.

**Accelerating multimodal pretraining:** Using reference models pretrained on small, clean datasets, we substantially accelerate pretraining on much larger,

noisier datasets. Moreover, we find ActiveCLIP to be complementary to recent data-curation techniques [12] and learning objectives [44], yielding a new state-of-the-art in several multimodal understanding tasks.

**Amortizing data selection policies:** Data-selection policies trained on one task can also accelerate the training of subsequent models on different but related tasks, suggesting that such policies can be easily derived from pre-trained models.

**Simplifying active-learning with online reference models:** Lastly, we demonstrate that pre-trained reference models may not be necessary at all, where these models are small and can be trained in parallel on larger batches than the learner model, while remaining *compute-positive*.

## 2 Related Work

**Data pruning.** One approach to data-selection is to identify and sub-select data ahead of training. For example, [28] and [35] show that the training loss and gradients can be used to discard large portions of small-to-medium sized datasets (*e.g.* CIFAR10 and ImageNet) with little loss in performance. These methods have since been deployed for the curation of web-scale datasets in both language modeling [26] and multimodal learning [1, 12, 25], demonstrating large reductions in the amount of data required together with performance improvements. However, in the single-epoch regime that is becoming typical of large model training [16, 19], pre-filtering can be as expensive as learning from it, a shortcoming which we address in this work. Nevertheless, we show that our method for dynamic data selection is complementary to and benefits from such data curation techniques.

**Online active learning.** Unlike data-pruning, online active learning continuously filters data throughout training and applies naturally to the semi-infinite, single-epoch regime. Online Batch Selection [22] scores and filters using the learner model, which has the theoretical advantage that the importance of data can be determined relative to the current state of the learner. In terms of metrics, the Reducible Holdout Loss (RHO) [27] also uses the concept of a reference model to identify learnable data points not yet well represented. Other proposed heuristics include memorization for long-tailed data [11] and assigning "complexity" scores based on the number of times the example is forgotten during training [38]. None of these approaches however have demonstrated that the cost of scoring can be reduced to the point of justifying learner efficiency gains.

**Compute-efficient data selection.** Several works have demonstrated the benefits of selecting data based on simple heuristics, such as low-level image properties [2] or proximity to high-quality text corpora [3, 7, 29, 40]. While cheap to compute, these statistics often require domain-specific knowledge which limits their applicability across tasks. Domain-agnostic methods such as core-sets alleviate this by selecting data based on the geometry of their embeddings [4, 14] which can be efficiently computed, however these algorithms generally do not

scale to large-scale datasets [8]. Most related to our work is DoReMi [39] which uses domain-general, scalable, and compute-efficient proxy models for the simpler problem of determining optimal data-mixtures for the subsequent training of a larger language model.

## 3 Methods

### 3.1 Data selection as prioritized replay

We use online batch selection [22] to apply our scoring heuristics to standard visual learning tasks: firstly, we sample uniformly from the training set $\boldsymbol{x}_i \overset{\mathcal{U}}{\sim} \mathcal{D}$ and compute a score $s_i = s(\boldsymbol{x}_i|\theta) \in \mathbb{R}$ to each data point $\boldsymbol{x}_i$ using model parameters $\theta$. Given a large enough collection of scored examples stored in a memory bank $\mathcal{M} = \{\boldsymbol{x}_i\}_{i\in(0,\dots,M-1)}$, we then sample *non-uniformly* according to their scores $\boldsymbol{x}_i \overset{\pi}{\sim} \mathcal{M}$ [31], where $\pi(\boldsymbol{x}_i) = \text{Softmax}(\{s_i\}_{i\in(0,\dots,M-1)})$. A batch of such examples is used to update the learner model. Following convention in reinforcement learning, we refer to the scoring and target models as *actors* and *learners* respectively.

### 3.2 Statistics for data selection

We explore a few statistics for model-based prioritization, grouped into two categories.

**Example difficulty:** given the current state of the learner, an intuitive prioritization scheme might favour 'difficult' examples (as measured by their training loss), while removing 'easy' examples that are trivially classified and which yield small gradients. This loss-based prioritization:

$$s^{\text{hard}}(\boldsymbol{x}_i|\theta) = \ell(\boldsymbol{x}_i|\theta) \tag{1}$$

can use the current parameters of the learner $\theta^t$ or those of a fixed model $\theta^*$. The opposite argument can been made for favoring examples that are *easily* solved by a well-trained model, as such a prioritization removes the noisy examples present in large-scale datasets:

$$s^{\text{easy}}(\boldsymbol{x}_i|\theta) = -\ell(\boldsymbol{x}_i|\theta) \tag{2}$$

This scheme is commonly used in multimodal learning for identifying high-quality examples with pre-trained models [15, 32, 33].

**Example *learnability*:** Given that favoring easy and hard examples target different and potentially orthogonal properties of the data, a natural question is whether these policies can be combined. *Learnability* criteria straightforwardly combine the two as

$$s^{\text{learn}}(\boldsymbol{x}_i|\theta^t, \theta^*) = s^{\text{hard}}(\boldsymbol{x}_i|\theta^t) + s^{\text{easy}}(\boldsymbol{x}_i|\theta^*) \tag{3}$$

$$= \ell(\boldsymbol{x}_i|\theta^t) - \ell(\boldsymbol{x}_i|\theta^*), \tag{4}$$

favoring examples that are easily solved by a well-trained model $\theta^*$ but challenging to the learner in its current state $\theta^t$, such that more computation dedicated to this example could lower its loss. Conversely, examples that are trivially classified by the learner (or mislabeled) will yield low (or high) losses for *both* the current learner and the well-trained one, leading to low learnability scores.

A special case of learnability scores (the *RHO* loss, [27]) uses a model $\theta^{\text{ho}}$ specifically trained on a held-out dataset to ensure the independence of its predictions from those of the current learner $s^{\text{learn}}(\boldsymbol{x}_i|\theta^t, \theta^{\text{ho}})$. We assess in Section 4.2 whether this is necessary when training on large-scale image datasets.

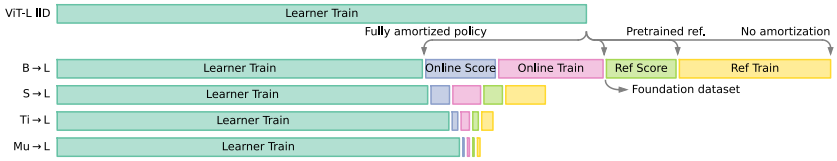


**Fig. 2: Amortizing the cost of data selection**. Drawn to scale: length of bars indicates number of FLOPs required to reach the accuracy of a ViT-L trained with uniform sampling ("ViT-L Uniform Sampling", see Figure 4). Expensive model policies (e.g. a ViT-B scores data for the ViT-L learner, or 'B → L') produce large learner speedups, at the expense of the additional FLOPs associate with data selection. This overhead can be reduced by deriving the data-selection policies from smaller models (e.g. ViT-S, ViT-Ti or ViT-Mu score data for the ViT-L learner), at the expense of marginal decreases in the learner speedup. Costs could be additionally amortized by using off-the-shelf reference models, removing the need to train from scratch (yellow). Since the reference model is fixed throughout training, scores can be assigned once to a 'foundation dataset' and amortized across many training runs (lime green; [27, 35]). Since the online model is independent of the learner model and generalizes across scale, data selection policies can also be distilled as a fixed ordering of a given dataset (a 'foundation curriculum').

### 3.3 Unlocking compute-positive training

Scoring requires inference passes over the *actor* and *learner* models. We assume that gradient updates to cost 3x inference passes. The cost of scoring data $F$ thus scales with the proportion of data which is being rejected (*e.g.* retaining only 20% of the data requires 5 inference passes per trained batch). The requirements for compute-positivity can therefore be expressed as:

$$\underbrace{\Big(3F_{\text{learn}} + \rho F_{\text{act}}\Big)\beta + 3F_{\text{ref}}}_{\text{Active Learning}} < \underbrace{3F_{\text{learn}}}_{\text{Uniform Sampling}} \tag{5}$$

where $F_{\text{act}}$ is the cost of scoring an example, $\rho$ is the number of examples scored per training example, and $\beta$ is the efficiency gain (in terms of the learner update)

relative to training with uniform sampling (see Appendix Section A.1 for more details). The RHS term is the cost per update of uniform sampling. The first LHS term inside the brackets is the cost of the learner training during AL, the second term is the scoring cost and the last term outside the brackets the cost of training the reference model. We illustrate in Figure 2 the different contexts in which parts of this computation may be effectively amortized.

In the large-scale training regime where data is neither repeated nor seen before, compute-positivity requires that either or all of the reference model training, actor scoring and learner efficiency $\beta$ terms must be made smaller to produce net savings relative to uniform sampling. Typical prioritization schemes can produce saving on the order of 50% (ie $\beta = 0.5$), suggesting that savings must also be made by down-scaling the other terms $F_{\text{act}}$ and $F_{\text{ref}}$.

**Cost of easy-reference scoring.** While both the cost of the reference model and example scoring can be scaled down in the case of easy-reference scoring with small models ($F_{\text{act}} = F_{\text{ref}}$, see Equation 2), it is unclear whether the efficiency gains $\beta$ are robust to this down-scaling (see section 4.2).

**Cost of RHO *learnability* scoring.** The original definition of learnability scores [27] requires inference passes through both the learner and a reference model ($F_{\text{act}} = F_{\text{ref}} + F_{\text{learn}}$, see Equation 3), meaning that although the cost of the reference model can be reduce by using a smaller model, the cost of example scoring cannot.

**Cost of ClassAct / ActiveCLIP.** For this reason, we explore whether replacing the learner model in term 1 of Equation 3 with a much smaller model can still produce comparable learner efficiency gains to those already observed (see Appendix Algorithm 1). Specifically, we introduce a third "online" model, which has the same architecture and size as the reference model, but is trained in parallel with the learner. In this case, the cost of scoring examples reduces to:

$$F_{\text{act}} = F_{\text{ref}} + F_{\text{online}} = 2F_{\text{ref}} \tag{6}$$

and can be scaled down along with the reference model. We instantiate our method for two canonical pre-training tasks: visual classification and multimodal learning, which we call ClassAct, ActiveCLIP and ActiveSigLIP respectively. In Appendix Section A.6, we describe an asynchronous active learning framework where scoring and learning is performed in parallel on separate devices. This setup does not affect the FLOP calculations but can mitigate the overhead in time even when using non-approximate scoring.

### 3.4 Losses for canonical visual pre-training tasks

For visual classification with ClassACT, we use the standard cross-entropy loss for both actors and learners. For multimodal learning with ActiveCLIP, learners optimize the contrastive loss $\ell_{\text{learn}} = \ell^{\text{im,txt}}_{\text{learn}} + \ell^{\text{txt,im}}_{\text{learn}}$, whereas the actor loss $\ell_{\text{act}}$

**Algorithm 1** ClassAct / ActiveCLIP

**Input:** Randomly initialized learner model $\theta_l$ and small online model $\theta_o$, small pre-trained reference model $\theta_r$. Models use loss $\ell_{\text{act}}$ for scoring data and $\ell_{\text{learn}}$ for computing updates. Dataset $\mathcal{D}$, batch size $B$, sub-batch size $b < B$.
**while** training **do**
  $X \sim \mathcal{D}$, where $|X| = B$ ▷ Sample uniformly
  $S = \ell_{\text{act}}(X|\theta_o) - \ell_{\text{act}}(X|\theta_r)$ ▷ Get scores
  $I \sim \text{SoftMax}(S)$, where $|I| = b$ ▷ Sample indices
  $Y = X[I]$ ▷ Collect sub-batch
  $\theta_l \leftarrow \text{Adam}[\nabla_{\theta_l} \ell_{\text{learn}}(Y|\theta_l)]$ ▷ Update learner model
  $\theta_o \leftarrow \text{Adam}[\nabla_{\theta_o} \ell_{\text{learn}}(Y|\theta_o)]$ ▷ Update online model
**end while**

is simply the dot-product similarity between image and text embeddings:

$$\ell_{\text{act}}(\boldsymbol{x}_i|\theta) = -\boldsymbol{z}_i^{\text{im}} \cdot \boldsymbol{z}_i^{\text{txt}} \tag{7}$$

$$\ell_{\text{learn}}^{\text{im,txt}}(\boldsymbol{x}_i|\theta) = -\log \frac{\exp(\boldsymbol{z}_i^{\text{im}} \cdot \boldsymbol{z}_i^{\text{txt}})}{\sum_j \exp(\boldsymbol{z}_i^{\text{im}} \cdot \boldsymbol{z}_j^{\text{txt}})} \tag{8}$$

where $\boldsymbol{z}_i^{\text{im}} = f^{\text{im}}(\boldsymbol{x}_i;\theta)$ and $\boldsymbol{z}_i^{\text{txt}} = f^{\text{txt}}(\boldsymbol{x}_i;\theta)$ are image and text embeddings respectively, and $\ell_{\text{learn}}^{\text{txt,im}}(\boldsymbol{x}_i;\theta)$ is defined analogously to $\ell_{\text{learn}}^{\text{im,txt}}(\boldsymbol{x}_i;\theta)$. Similarly, ActiveSigLIP instead uses the sigmoid loss [44] for the learner's objective and $\ell_{\text{act}}$ for scoring.

## 4 Experiments

All our experiments were conducted with Vision Transformers [9] for which strong baselines are available across model sizes [43]. Unless specified, we adopt models with patch-size 16 throughout (ViT-S refers to ViT-S/16 and similar). We consider two canonical tasks for large-scale pretraining: classification on JFT-300M [36] and multimodal contrastive learning [30] on large image-text datasets. When pre-training with JFT classification we use held-out classification performance as the evaluation metric. When pre-training on image-text data we evaluate with standard multimodal transfer tasks: ImageNet zero-shot classification and image-to-text / text-to-image retrieval on COCO.

Throughout, we will refer to the large batch of size $B$ sampled uniformly from the training data as the 'super-batch', and the prioritised smaller batch of size $b < B$ as the 'sub-batch'. In all our experiment, we filter 50% of uniformly sampled data such that $\rho = B/b = 2$, although more aggressive filtering regimes warrant investigation [35].

### 4.1 Evaluating loss-based scoring heuristics in the large-data regime

We begin by evaluating loss- and learnability-based heuristics on their ability to accelerate supervised classification on JFT (Fig. 3). Arguably the most intuitive

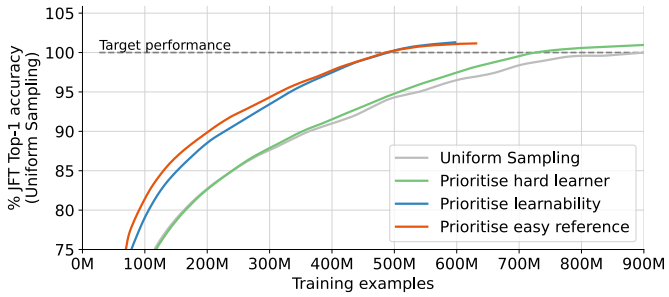


**Fig. 3: Evaluation of loss-based data-selection criteria for large-scale classification**. We train a ViT-B on JFT-300M with different data-selection policies. Prioritising hard data under the learner (green curve) produced marginal gains over the uniform sampling baseline. Prioritizing data using both *learnability* (blue curve, [27]) and *easy reference* prioritization (red curve, [15]) produced significant speedups and performance gains.

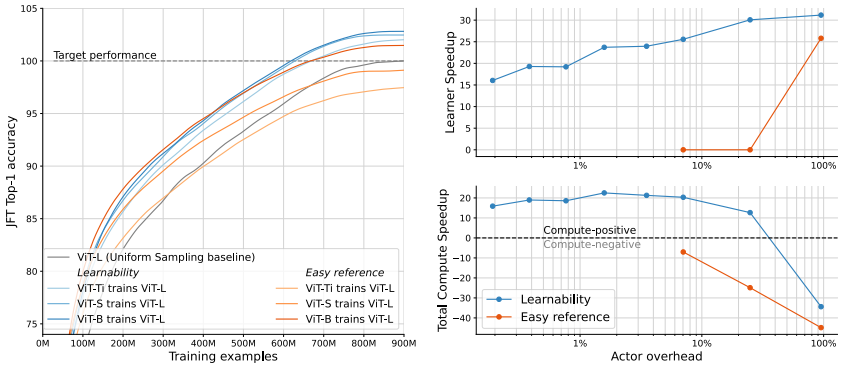


**Fig. 4: Generalization of data-selection policies across models scales. Left:** We train a ViT-L for 3 epochs on JFT using uniform sampling (grey) or prioritized data sampling using example learnability (blue) or low-loss under the reference model (red). Example scores are computed using ViT-B actors (dark), or cheaper ViT-S or ViT-Tiny models (light). While both example learnability and "easy reference" yield good speedups with expensive actors, learnability criteria are much more robust to approximate scoring. **Top right:** Learner (ViT-L) speedup is computed as the fraction of learner iterations saved in order to attain the baseline's top performance. Actor overhead is computed as the additional computation in FLOPs required to score examples with a particular actor architecture (varying from ViT-Mu to ViT-L, see Appendix Table 4). Example learnability yields robust learner speedups across actor scales, "easy reference" scoring does not. **Lower right:** total compute efficiency is calculated as a product of learner efficiency and actor overhead, indicating the amount of computation required to reach baseline performance. Approximate actors (*i.e.* ViT-S or smaller) computing example learnability enable total compute speedups, other schemes do not.

method to score data is to prioritise training on data with high loss under the learner (*hard learner*). In our experiments, this strategy (Eq. 1) only marginally improved performance over the uniform sampling baseline, despite requiring an additional inference pass over the super-batch. This is perhaps not surprising - data points with high loss may also be unlearnable due to *e.g.* label noise, such that training on those data points does not result in the model performing any better on the held-out test set. Large scale datasets are more likely to be noisy.

Scoring methods based on pre-trained reference models performed much better—both *easy reference* (equation 2) and *learnability* (equation 3) -based prioritization produced significant gains over uniform sampling. Here, we pre-trained an identical ViT-B for the same 3 epochs to use as a reference model for a second training run displayed above. producing speed-ups of 33% (Fig. 3).

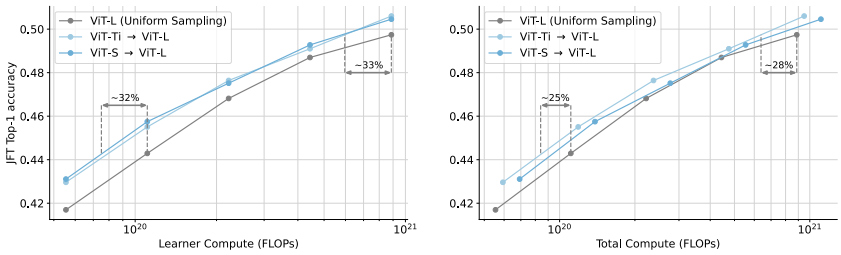


**Fig. 5: Scaling laws for active learning**. We trained a baseline ViT-L over a range of compute budgets (for which ViT-L is compute optimal, see Zhai et al., 2021). We also trained the same ViT-L with both ViT-Ti and ViT-S reference policies, pre-trained for the same number of epochs. **Left**: Small model policies produce robust savings in learner compute. **Right**: When accounting for total compute (learner + actor training and data scoring), small model policies in all compute budgets produce FLOP savings over training with uniform samples. These scaling laws generalize those measured empirically in the uniform sampling setting [43] to the case of non-uniform data selection.

### 4.2 Generalising data-selection policies across scale

The speed-ups afforded by *learnability* prioritization (Fig. 3) come at the cost of the additional inference passes required to score the data during learner training, plus the cost of training the reference model. This makes the overall gains strongly *compute-negative* relative to training with uniform samples. Even if the size of the reference model is scaled down [27], these methods still incur the cost of additional learner inference passes to score the data during training.

**Unlocking compute-positive active learning.** To address this issue, we introduce a set of down-scaled models with the same ViT architecture that we use to score data for training a larger ViT-L model (see Methods Section 3.3. We use ViT-B, ViT-S or ViT-Ti variants (which are 4×, 13× and 47× cheaper than

| Method | ViT model capacity | | | | Speed-up | |
|---|---|---|---|---|---|---|
| | Reference | Online Learner | | Reference Type | Learner speedup | Compute speedup |
| Uniform Sampling (ViT-B) | | | B | | 0% | 0% |
| RHO | Tiny | B | B | Held-out, fixed | 0% | - 79% |
| ClassAct-HO | Tiny | Tiny | B | Held-out, fixed | 18% | 3% |
| ClassAct | Tiny | Tiny | B | In-domain, fixed | 18% | 3% |
| ClassAct-Online | Tiny | Tiny | B | Trained online | 17% | 2% |

**Table 1: Simplifying and accelerating the computation of learnability scores.** Relative to RHO [27], *ClassAct* makes two changes: replacing the reference model with one trained in-domain (removing the need for bespoke held-out sets), and dramatically reducing capacity of the online actor models used for scoring examples. All experiments were conducted on 3 epochs of one-half of JFT to enable the held-out ablations. RHO with a small reference model did not produce a learner speedup in our experiments.

the learner) for both the online and reference model (see Appendix Algorithm 1). In Figure 4 (left) we assess the impact of these cheaper scoring models on learner efficiency. First, we find that easy reference prioritization to be very sensitive to the capacity of the scoring model: while ViT-B scoring models yield reasonable gains over uniform sampling, prioritizing with ViT-S and ViT-Ti scoring models underperforms significantly (Fig. 4, red curves).

In contrast, we find that *learnability* based prioritization yields robust gains, even when the scoring models are significantly scaled down (ClassAct; Fig. 4, blue curves). For example, while ViT-B scoring models yield a 31% learner speedup, the 50× smaller ViT-Ti scoring models still provide a 26% speedup. We pushed this logic by using even smaller scoring models (the ViT-Mu family which we introduce, see Appendix ) which are up to 1000× smaller than the learner. Despite this, prioritizing data based on their scores yields non-negligible speedups (*e.g.* 16% for the smallest actors we consider; Figure 4, top right).

These experiments demonstrate that, with the appropriate scoring criterion, online and reference models can be significantly downscaled and still produce comparable gains to larger models, with learner efficiency degrading gracefully with the actor overhead (*i.e.* the cost of the reference model and data scoring). As a result, our method ClassAct quickly becomes FLOP positive as the online + reference models are downscaled (Figure 4, bottom right), while at the same time producing speed-ups in wall-clock time for a given learner batch size.

Together, our results expose a pareto front across which to determine an optimal context-specific data selection strategy (Figure 2). Where pre-trained models are available, some of the cost of larger data selection policies can be discounted. If savings in wall-clock time supersede the associated cost of scoring, large models can be tolerated for data selection. Reference model costs can also be amortized across many training runs by appending scores to 'foundation datasets'. However, in the case where no component of the framework can amortized (as in the case of large-scale pretraining), prioritizing data with small ClassAct models can deliver large savings in total computation.

### 4.3 Generalising neural scaling laws to the active-learning setting

We next investigated the scaling behaviour of ClassAct by experimenting over large learner compute budgets (Fig. 5), using both ViT-Ti and ViT-S models as actors for training a ViT-L. Predictably, the ViT-S produced larger, although marginal gains over the ViT-Ti actors when not accounting for scoring FLOPs (Figure 5, left). However, when accounting for total FLOPs, the difference was less pronounced (Figure 5, right). Our results generalize large scale uniform sampling scaling laws such as uncovered by [19] for LLMs and reproduced for large vision transformers by [43] to the case of non-uniform sampling. For the first time, we demonstrate that these scaling laws can be shifted in our favour by selecting data using general model-based scoring heuristics.

### 4.4 Training the reference model in parallel

The reference model needs to be trained if none is already available. This two-step process adds complexity to active model training, especially if using large scale infrastructure. However, an interesting consequence of down-scaling the reference model is that both inference passes and gradients can be computed over a much larger batch than can be computed on the learner. In theory, this would mean that the small reference model could instead be trained *online*, in parallel with the large learner and small online model.

We confirmed our hypothesis by running an experiment in which we trained our reference model on a super-batch of size $B = 10b$ and trained the online and learner model in sequence with the sub-batch of size $b$. To make sure the reference model quickly converges, we additionally set the learning rate to double that of the online and learner models (this would cause instability for the learner and online models because of the additional variance from the smaller batch). We also verified that training the reference model on a held-out set of data performed equally in our experiments to reference models trained on the same data as the learner model [27]. Our 'one-pass' setup, *Online-ClassAct*, produces the same performance as the pre-trained ClassAct pipeline in our Ti-trains-B experiments (Table 1). Pseudocode is shown in appendix Algorithm 1.

We have shown that by *decoupling the scoring models from the learner model entirely*, it is possible to significantly downscale the scoring models with minor degradation to performance (see Table 1). Unlike RHO, which can train a large 'learner' model with a small 'reference' model, we introduce a third 'online' model, with the same architecture and parameter count as the reference model, enabling the reduction of actor computation (see Table 1).

### 4.5 ActiveCLIP: active multimodal learning

We have so far demonstrated that large scale image classifiers can be trained with lower total compute by actively selecting the data used for training. However, classification has largely been superseded as a large scale pre-training method

| | ImageNet 0-Shot | | COCO (im2txtR1) | | COCO (txt2imR1) | | *(Learner)* |
|---|---|---|---|---|---|---|---|
| Speed-up % | 23.0 | 48.1 | 28.6 | 48.0 | 26.8 | 27.0 | ALIGN |
| (vs. Uniform) | 0.0 | 16.3 | 18.8 | 22.1 | 28.2 | 17.5 | LTIP |
| Performance | 47.8 (2.2) | 53.2 (7.4) | 40.9 (2.4) | 44.8 (6.3) | 27.3 (1.8) | 30.5 (5.0) | ALIGN |
| (vs. Uniform) | 46.5 (-0.1) | 47.2 (0.6) | 45.6 (1.2) | 45.4 (1.0) | 31.8 (1.7) | 31.0 (0.9) | LTIP |
| *(Reference)* | ALIGN | LTIP | ALIGN | LTIP | ALIGN | LTIP | |

**Table 2: Generalizing data-selection policies across datasets and tasks.** We pretrain reference models on the large but noisy ALIGN dataset, or the smaller and more curated LTIP dataset [2]. Consistent with [2], we find training with uniform sampling on LTIP to yield stronger transfer learning results than pre-training on ALIGN. These reference models can be used very effectively for data-selection on both LTIP and ALIGN, whereas ALIGN-pretrained reference models yield more modest speedups. All models are provided with 800M training images at resolution 128×128, speedups are shown relative to the time at which the uniform sampling baseline was reached for that evaluation metric. Colour indicates performance relative to uniform (brackets).

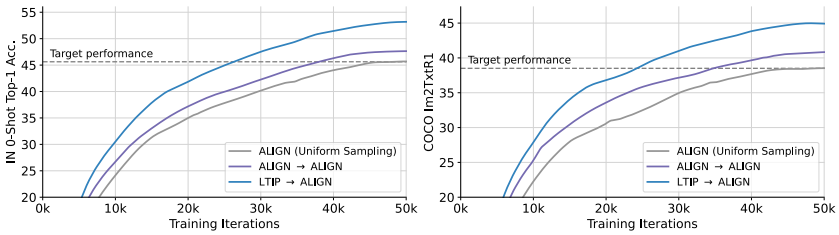


**Fig. 6: Reference policies generalize across tasks**. We train a ViT-B reference model on either ALIGN or LTIP, then use it to train a second ViT-B learner on ALIGN with ActiveCLIP. The biggest gains were found with an LTIP reference model, despite it needing to perform out-of-domain generalization. In 50k iterations, ActiveCLIP selects the 800M "cleanest" examples from the ALIGN dataset, whose size is 1.6B in total.

by CLIP-style multimodal training [30]. Figure 6 demonstrates that our CLIP-adapted active learning method *ActiveCLIP* (see Methods) produces similar speedups in terms of learner computation as observed in JFT classification. Specifically, we find that prioritized sampling with learnability scores accelerates multimodal pre-training by 18-48%, depending on the evaluation metric (ImageNet zero-shot accuracy or COCO retrieval) and reference model configuration, which we explore below.

### 4.6 Policy generalization across tasks

To fully leverage off-the-shelf pre-trained image models for training (Figure 2), our results up to now suggest that the reference model should be trained on the same task that the learner model is being trained for. In Figure 6 (see Table 2 for full results), we show that we can in fact pre-train our reference models on related but distinct datasets. Moreover results suggest that there may even be a benefit

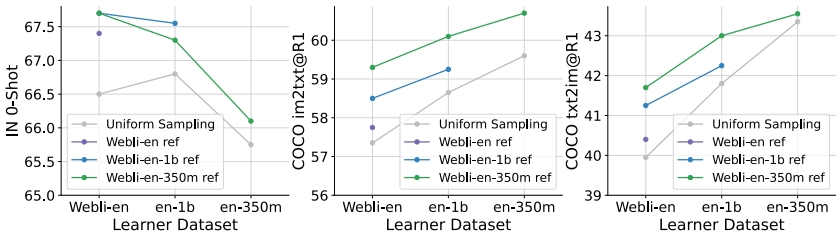


**Fig. 7: Reference models trained on curated datasets are powerful data selectors**. Active data selection using *ActiveSigLIP* with reference models trained on increasingly curated subsets of Webli. Webli-350M reference models are consistently more effective than those trained on raw or 1B subsets.

to cross-training in some-cases; an ALIGN→ALIGN reference→learner model combination ('in-domain' ActiveCLIP) produced similar speedups to ClassAct on JFT. However, these gains were greatly surpassed by an LTIP→ALIGN combination. One possibility is that LTIP is less noisily labeled such that the scoring policies it produces are 'cleaner' - i.e. more able to filter for 'clean' data. The corollary may also be true; active data selection appears to greatly improve the utility of ALIGN, suggesting that it contains a large proportion of data that is of low quality for training.

We observed the same effect when training with ActiveSigLIP on the large scale Webli dataset [6] (results summarised in Table 3). We trained reference models on both the raw 4B set as well as two extensively curated 1B / 350M subsets, which were then used to train subsequent learner models (Fig. 7). In all-but-one cases, filtering data with the 350M-trained referenced model produced the best results when transferring to 350M, 1B and 4B learner datasets. Notably, the significant gains observed over training with uniform sampling on the 350M subset suggest that our method can still improve over methods that pre-filter datasets once but then train with uniform samples [12].

### 4.7 Comparison to prior multimodal art

We leverage the insight that pre-training a reference model on clean data can facilitate learning on larger, noisy data for training our final model. Here, we train a reference model on LTIP, then use it to train a new model on the much larger mixture of LTIP and ALIGN, following [2], for a total of 3 and 8 billion training examples. Table 3 shows that in this training regime, *ActiveCLIP* surpasses models trained with significantly more data on ImageNet 0-Shot classification and COCO retrieval. Finally, we find that our active learning method is complementary to recent advances in multimodal learning: ActiveSigLIP significantly improves uniformly-trained SigLIP [44] in both COCO retrieval metrics.

| | | IN-1K | COCO | |
|---|---|---|---|---|
| Method | Train ex. | ZS Top-1 | im2txt | txt2im |
| CLIP | 13B | 68.3 | 52.4 | 33.1 |
| EVA-CLIP | 3B+2B | 69.7† | | |
| ActiveCLIP | 3B | **71.3** | **57.7** | **43.0** |
| OpenCLIP | 34B | 70.2 | 59.4 | 42.3 |
| EVA-CLIP | 8B+2B | 74.7† | 58.7 | 42.2 |
| ActiveCLIP | 8B | **72.2** | **60.7** | **44.9** |
| SigLIP | 3B | 72.1 | 60.7 | 42.7 |
| ActiveSigLIP | 3B | 72.0 | **63.5** | **45.3** |

**Table 3: Comparison of ActiveCLIP to public multimodal pre-taining methods**. ActiveCLIP outperforms models trained with the same or more data (CLIP, [30]; EVA-CLIP, [37]; and OpenCLIP, [17]). ActiveSigLIP produced significant gains over the baseline SOTA performance of SigLIP [44]. †benefits from additional ImageNet21K pretraining (+2B training examples). All ActiveCLIP/SigLIP models use a reference model trained on LTIP to guide learning on a mixture of ALIGN and LTIP.

## 5 Discussion

In this work, we have presented a new method for active data selection that builds upon and simplifies the concept of 'learnability'. Our experiments demonstrate that this approach can significantly reduce the computation required for large-scale pretraining, compared to training with uniform samples. To our knowledge, this is the first active learning method that is more efficient than training with uniform samples when accounting for total FLOPs, and that does not rely on hand-designed features, allowing broad application across training setups. We have validated this by showing results on classification and contrastive pretraining, and found that our data selection policies continue to produce efficiency gains in the large-scale regime and can generalize effectively across task modalities. Collectively, our experiments also illustrate a Pareto frontier that allows trading off actor/data-selection computation against savings in training iterations, suggesting an alternative path to improved performance beyond scaling training batch sizes.

This work focused on supervised pre-training for images, but further work could involve extending our method to other modalities and training schemes such as language, video, and generative modeling. An important note is that all our experiments present results from filtering only 50% of the data; further gains may be possible by filtering more aggressively, at the cost of further overheads. In particular, aggressive data-selection coupled with efficient scoring schemes such as the ones proposed here could test the hypothesis that large-scale pretraining can benefit from exponential, rather than power-law, scaling behavior.

## References


1. Abbas, A., Tirumala, K., Simig, D., Ganguli, S., Morcos, A.S.: Semdedup: Data-efficient learning at web-scale through semantic deduplication. arXiv preprint arXiv:2303.09540 (2023)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: A visual language model for few-shot learning. Advances in Neural Information Processing Systems (2022)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems (2020)
4. Campbell, T., Broderick, T.: Bayesian coreset construction via greedy iterative geodesic ascent. In: International Conference on Machine Learning. pp. 698–706. PMLR (2018)
5. Cassirer, A., Barth-Maron, G., Brevdo, E., Ramos, S., Boyd, T., Sottiaux, T., Kroiss, M.: Reverb: A framework for experience replay (2021)
6. Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al.: Pali: A jointly-scaled multilingual language-image model. arXiv preprint arXiv:2209.06794 (2022)
7. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. Journal of Machine Learning Research **24**(240), 1–113 (2023)
8. Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., Zaharia, M.: Selection via proxy: Efficient data selection for deep learning. arXiv preprint arXiv:1906.11829 (2019)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
10. Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al.: Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In: International conference on machine learning. pp. 1407–1416. PMLR (2018)
11. Feldman, V.: Does learning require memorization? a short tale about a long tail. In: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing. pp. 954–959 (2020)
12. Gadre, S.Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., et al.: Datacomp: In search of the next generation of multimodal datasets. arXiv preprint arXiv:2304.14108 (2023)
13. Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C.C.T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al.: Textbooks are all you need. arXiv preprint arXiv:2306.11644 (2023)
14. Har-Peled, S., Kushal, A.: Smaller coresets for k-median and k-means clustering. In: Proceedings of the twenty-first annual symposium on Computational geometry. pp. 126–134 (2005)
15. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. arXiv preprint arXiv:2104.08718 (2021)

16. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D.d.L., Hendricks, L.A., Welbl, J., Clark, A., et al.: Training compute-optimal large language models. In: Advances in Neural Information Processing Systems (2022)
17. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: Openclip (Jul 2021). https://doi.org/10.5281/zenodo.5143773, https://doi.org/10.5281/zenodo.5143773, if you use this software, please cite it as below.
18. Jouppi, N.P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al.: In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th annual international symposium on computer architecture. pp. 1–12 (2017)
19. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Eur. Conf. Comput. Vis. pp. 740–755. Springer (2014)
21. Lindley, D.V.: On a measure of the information provided by an experiment. The Annals of Mathematical Statistics **27**(4), 986–1005 (1956)
22. Loshchilov, I., Hutter, F.: Online batch selection for faster training of neural networks. arXiv preprint arXiv:1511.06343 (2015)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
24. MacKay, D.J.: Information-based objective functions for active data selection. Neural computation **4**(4), 590–604 (1992)
25. Mahmoud, A., Elhoushi, M., Abbas, A., Yang, Y., Ardalani, N., Leather, H., Morcos, A.: Sieve: Multimodal dataset pruning using image captioning models. arXiv preprint arXiv:2310.02110 (2023)
26. Marion, M., Üstün, A., Pozzobon, L., Wang, A., Fadaee, M., Hooker, S.: When less is more: Investigating data pruning for pretraining llms at scale. arXiv preprint arXiv:2309.04564 (2023)
27. Mindermann, S., Brauner, J.M., Razzak, M.T., Sharma, M., Kirsch, A., Xu, W., Höltgen, B., Gomez, A.N., Morisot, A., Farquhar, S., et al.: Prioritized training on points that are learnable, worth learning, and not yet learnt. In: International Conference on Machine Learning. pp. 15630–15649. PMLR (2022)
28. Paul, M., Ganguli, S., Dziugaite, G.K.: Deep learning on a data diet: Finding important examples early in training. Advances in Neural Information Processing Systems **34**, 20596–20607 (2021)
29. Prabhu, A., Dognin, C., Singh, M.: Sampling bias in deep active classification: An empirical study. arXiv preprint arXiv:1909.09389 (2019)
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)
31. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv preprint arXiv:1511.05952 (2015)
32. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022)

33. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
34. Settles, B.: Active learning literature survey (2009)
35. Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., Morcos, A.: Beyond neural scaling laws: beating power law scaling via data pruning. Advances in Neural Information Processing Systems **35**, 19523–19536 (2022)
36. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 843–852 (2017)
37. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389 (2023)
38. Toneva, M., Sordoni, A., Combes, R.T.d., Trischler, A., Bengio, Y., Gordon, G.J.: An empirical study of example forgetting during deep neural network learning. arXiv preprint arXiv:1812.05159 (2018)
39. Xie, S.M., Pham, H., Dong, X., Du, N., Liu, H., Lu, Y., Liang, P., Le, Q.V., Ma, T., Yu, A.W.: Doremi: Optimizing data mixtures speeds up language model pretraining (2023)
40. Xie, S.M., Santurkar, S., Ma, T., Liang, P.: Data selection for language models via importance resampling. arXiv preprint arXiv:2302.03169 (2023)
41. Yang, F., Barth-Maron, G., Stańczyk, P., Hoffman, M., Liu, S., Kroiss, M., Pope, A., Rrustemi, A.: Launchpad: A programming model for distributed machine learning research. arXiv preprint arXiv:2106.04516 (2021), https://arxiv.org/abs/2106.04516
42. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: CoCa: Contrastive captioners are image-text foundation models. In: Transactions on Machine Learning Research (2022)
43. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12104–12113 (2022)
44. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. arXiv preprint arXiv:2303.15343 (2023)