The Science of Data Filtering: Data Curation cannot be Compute Agnostic

Reviewed on OpenReview:

Editor:

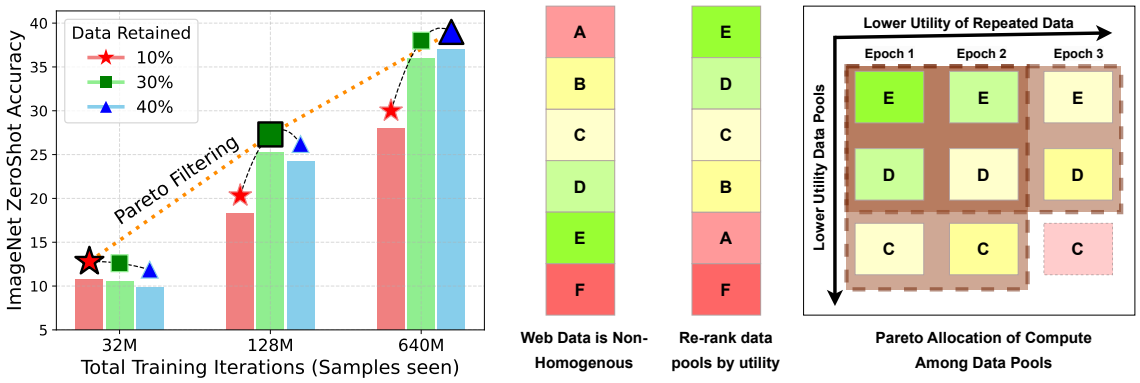
Abstract

Vision-language models (VLMs) are trained on massive web scrapes, requiring careful data curation. For instance, the LAION public dataset retained only about 10% of the total crawled data. In recent times, data curation has gained prominence with several works developing strategies to retain 'high-quality' subsets of 'raw' scraped data. However, these strategies are typically developed agnostic to the available compute for training. In this paper, we demonstrate that making filtering decisions independent of training compute is often suboptimal—well-curated data rapidly loses its utility when repeated, eventually decreasing below the utility of 'unseen' but 'lower-quality' data. In fact, we show that even a model trained on *unfiltered common crawl* obtains higher accuracy than that trained on the LAION dataset post 40 or more repetitions. While past research in neural scaling laws has considered web data to be homogenous, real data is not. Our work bridges this important gap in the literature by developing scaling trends that characterize the 'utility' of various data subsets, accounting for the diminishing utility of a data point at its 'nth' repetition. Our key message is that data curation *can not* be agnostic of the total compute a model will be trained for. Based on our analysis, we propose FADU (Filter by Assessing Diminishing Utility) that curates the best possible pool for achieving top performance on Datacomp at various compute budgets, carving out a pareto-frontier for data curation.

1 Introduction

Large scale visual-language models like CLIP are trained on massive scrapes of the web (Common Crawl), which are noisy and hence require careful curation. Datasets such as LAION datasets (Schuhmann et al., 2021) used a strategy of filtering out image-caption pairs that had 'low' similarity score as assessed by an already pre-trained CLIP model. Later approaches developed more sophisticated filtering methods (Abbas et al., 2023; Radenovic et al., 2023; Maini et al., 2023), often leading to improved performance of the resulting visual language models. To the best of our knowledge, however, all these data filtering methods make a common assumption—data filtering can be carried out *independent* of considering compute budget (i.e., the number of training steps) used to train the resulting VLM.

In this paper, we show that instead, there is a fundamental relationship between the performance of a data filtering mechanism and the ultimate compute budget. Specifically, we show that there exist scenarios where training on 'aggressively filtered' good data (such as the LAION dataset) is actually *worse* than naively training on the unfiltered common crawl. This is because, after repeating for more than 40 epochs, the filtered data has



(a) (b)

Figure 1: **(a)** Retaining top-30% data based on CLIP scoring is common in vision-language model training. Our results indicate that the filtering aggressiveness must adapt to the compute-to-data pool size ratio, addressing the diminishing utility of good data with repetitions. We present outcomes across 18 visual understanding tasks, employing a 128M sample global data pool at different compute scales. **(b)**The Dynamic Problem of Data Filtering: Web data is non-homogenous, and past work has succeeded at ranking various data subsets according to their diminishing quality (y-axis). However, training on 'high-quality' data for multiple epochs leads to diminishing utility (x-axis), an angle ignored in past work. Our work aims to answer—*what is the best allocation of and return for computational resources?*

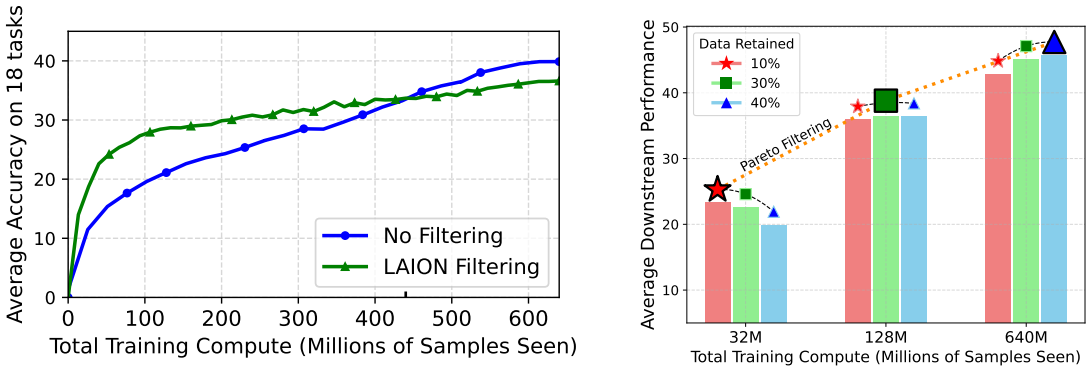
negligible remaining utility. On the other hand, common crawl samples, though lower in initial utility, are seen fewer times and hence have a higher utility than LAION towards the end. In other words, the utility of data diminishes with repetition, and hence filtering metrics must be designed by assessing the tradeoff between the diminishing utility of a small pool of 'high-quality' data, and the lower initial but slower diminishing utility of a larger pool that includes 'lower-quality' data.

In order to characterize this phenomenon, we develop new scaling laws for VLMs that account for the effect of repeatedly training (as afforded by the compute budget) on the same data points. We estimate the scaling curves of test error for models trained from 128M to 34B total samples (i.e. training steps) seen. Across multiple architectures and data scales, our scaling curves reliably fit the final test error of the models. Most importantly, this scaling allows allow us to predict the "pareto optimal" filtering approach: given a compute budget, we can determine a threshold of data filtering that leads to the best-performing model. Our estimated 'optimal' filtering threshold achieves state-of-the-art performance at each of the compute scales from 32M to 640M samples.

2 Data Filtering for a Compute Budget

2.1 Experimental setup

We are given a large initial pool of data to train a VLM (which we use synonymously with CLIP) and want to study the effects of data filtering at different compute budgets. As our base unfiltered pool, we use the "medium" scale (128M samples) of the recently data



(a) (b)

Figure 2: **(a)** Given an initial data pool of 128M samples, we train ViT-B/32 CLIP models for a total of 640M samples. As we increase the training compute, the accuracy gains on the 128M LAION data subset that aggressively filtered the common crawl to 10% of its initial size plateau. Surprisingly, even no-filtering of the common crawl is better than the popular LAION dataset after seeing more than 450M samples. **(b)** We modify the state of the art data curation approach by changing the filtering threshold after ranking the data by their metric. While the original paper proposed retaining 30% of the data, our results show that depending on the ratio of compute to data pool size, we must adaptively make the filtering less (or more) aggressive to account for the diminishing utility of good data with repetitions. Results are presented on an average of 18 visual understanding tasks with a global data pool size of 128M samples, and varying compute scales.

curation benchmark, Datacomp (Gadre et al., 2023a). In Datacomp, the compute budget is fixed to 128M, with the implicit assumption that data filtering methods will continue to obey their respective ordering in performance as we change the compute budget. In this work, we explicitly consider different compute budgets for training steps:$\{32M, 64M, 128M, 640M\}$ and study the performance of data filtering methods. Note that filtering to different amounts (for a fixed compute) changes the number of times each training sample is seen. At a compute budget of 128M, each sample is seen 10 times from a filtered pool of 12.8M samples.

We assess the performance of our models based on their zero-shot accuracies across a diverse set of 18 downstream tasks. This includes both—(a) classification tasks like ImageNet, ImageNetOOD, CIFAR10, etc., and (b) retrieval tasks like Flickr and MSCOCO. More details about the downstream evaluation tasks can be found in Appendix E.

2.2 When "good" data performs worse

We start with the popular LAION filtering strategy used in obtaining the LAION dataset (Schuhmann et al., 2021, 2022). This filters for image-caption pairs with a high similarity score (> 0.28) as measured by OpenAI's CLIP model. When filtering from common crawl, this threshold amounts to retaining just 10% of the original pool.

We first compare training without filtering (i.e. raw common crawl) with training on LAION-filtered subset, at varying compute budgets. Figure 2a shows the average down-

stream accuracy on 18 tasks (Section 2.1), as the total training iterations (compute) is scaled from 32M to 640M. We make the following observations:

1. **Good data is better at low compute budget**: In the regime of low training compute, utilizing high-quality data (for example, via LAION filtering) is beneficial, corroborating the conventional data filtering intuition. For instance, at 128M training iterations, LAION's approach of filtering surpasses the no-filter strategy significantly, achieving an increase of 7.5% zero-shot accuracy averaged over 18 tasks.

2. **Data filtering hurts at high compute**: The advantage offered by data filtering consistently diminishes with increasing compute budget. Remarkably, beyond 450M iterations, training on the unfiltered common crawl dataset outperforms that on LAION.

Why does the same data filtering, which supposedly picks the 'best' data, thereby improving performance at low compute, end up hurting performance at high compute? At a 450M compute budget, LAION-filtered data, retaining 10% of the pool, is seen approximately 32 times. This frequent repetition leads to diminishing utility for each sample. Initially, LAION-filtered data shows high utility at lower compute budgets due to minimal repetition. However, at higher computes, its utility drops significantly due to over-repetition. Conversely, unfiltered samples start with lower utility but experience a lesser decline, surpassing LAION-filtered data in utility over time due to fewer repetitions.

2.3 Data filtering must be compute-aware

In the previous section, we saw that the popular LAION-filtering method offered lower gains and eventually under performing the uncurated pool as we increase our training compute. We study the performance of some recently proposed state-of-the-art data filtering methods as we change our compute budget.

We specifically analyze two methods: (a) CLIP score filtering (b) `T-MARS` , which ranks data based on CLIP scores after masking text (OCR) features in images (Section B). We compare three levels of varying aggressive filtering for each data filtering approach, and vary total compute (training iterations) from 32M to 640M, just like before.

Figure 1a illustrates the comparison of top-10%, top-30%, and top-40% CLIP filtering at compute scales of 32M, 128M, and 640M. At a 32M compute scale, highly aggressive filtering, retaining only the top-10% data as per CLIP scores, yields the best results, while the least aggressive top-40% filtering performs the worst. However, this trend *reverses entirely as the compute is scaled to 640M* . While top-10% filtering excels at low training compute due to fewer repetitions, its utility diminishes rapidly with increased compute due to data repetition. Similar trends are observed with the `T-MARS` scoring metric (Figure 2b).

These observations underscore the need for a compute aware filtering strategy balancing two aspects: the high initial utility of high-quality data, which diminishes quickly due to repeated epochs, versus lower-quality but larger data that offers lower initial utility but a slower rate of decline due to fewer repetitions given a larger filtered subset pool size.

Can we turn this insight into a more performant compute-aware data filtering method? The straightforward strategy is to simply try varying levels of filtering at the compute budget and pick the best. But this is impractical. Now, we attempt at *effectively extrapolating* from smaller compute budgets to larger while accounting for diminishing utility with repetition.

3 Scaling Laws: Hypothesis on Utility

In the context of image-language modeling, let x_i denote an image-caption pair (I, T). Further, let $\mathcal{S}_n = \{x_i\}_{i=1}^n$ be the training set and $f(\mathcal{S}^k)$ denote the error of the model f after seeing $\mathcal{S}_n$ for k epochs. Following Cherti et al. (2023) we consider downstream zeroshot error on ImageNet as the empirical estimate for the model's error.

3.1 Defining Utility

First, let us consider the simple cases of assessing the utility of a single sample. Utility refers to the decrease in model error after seeing a sample once during the training. Mathematically, utility of $(n+1)^{\text{th}}$ sample is given by:

$$\mathcal{U}(x_{n+1}) = f(\{x_i\}_{i=1}^n) - f(\{x_i\}_{i=1}^{n+1}). \tag{1}$$

Past works on scaling laws (Kaplan et al., 2020; Jia et al., 2021) estimate the error of a model (at a fixed parameter count) after training for n samples as:

$$f(\{x_i\}_{i=1}^n) = an^b + d; a, d > 0; b < 0, \tag{2}$$

where $a > 0, b < 0$ and $d > 0$ are constants to be determined empirically. Intuitively, b factors in in the diminishing gains as more data is seen and also models the utility of the data pool itself, with a lower value indicating higher utility. Whereas, a is a normalizer and d estimates an irreducible error at the end of training to infinity. For instance, Cherti et al. (2023) noted that the b value for OpenAI's filtered dataset was lower than that of the LAION dataset, indicating it had higher utility. Plugging in equation 1, one can estimate the utility of $(n+1)^{\text{th}}$ sample as: $\mathcal{U}(x_{n+1}) = a[n^b - (n+1)^b]$. Note that the value of the exponent b is negative, and n is very large, hence the utility of any data point stays positive and keeps diminishing as we see more training samples.

3.2 Utility of repeated data

The loss definition above follows prior discourse in the literature that finds that model loss decays as a power law (Kaplan et al., 2020; Hoffmann et al., 2022). However, a key assumption in these works is that each data point is only seen once during training. This assumption while prevalent in the language modeling literature, is far from true in the vision-language literature. For example, the CLIP (Radford et al., 2021a) models were trained for 32 epochs on a dataset of 400M image-text pairs. Intuitively (and as seen empirically in § 2), the gains from repeatedly seeing the same sample should diminish with the epoch, something that the utility estimates in Equation 3.1 do not account for. This raises an important question—how does one model the diminishing utility with epochs?

We propose the following estimate for the utility of seeing a datapoint for the k^{th} time:

$$\mathcal{U}(x_{n+1}, k) = a_k[n^b - (n+1)^b], \quad a_k = a_0 \left(\frac{1}{2}\right)^{k/\tau} \tag{3}$$

where a_0 and τ (half-life) are constants to be estimated empirically. This equation corresponds to a half-life type decay of the marginal utility of an additional sample, if the sample

seen is repeated. Half-life τ is a factor that depends on the size of the data pool. Recently Muennighoff et al. (2023) (in the paradigm of language models) estimate the effective number of samples seen, when the data is repeated. However, in Appendix H we show that our proposed approach of decaying the marginal utility leads to a much better parametric fit.

3.3 Utility of a Data Pool

In practice, the utility of samples doesn't vary much within examples in a close neighborhood when ranked by any given data quality metric. Hence, rather than estimating per sample utility, we are more interested in estimating the accuracy when we train a model from scratch on a data pool for k epochs. Given a data pool $\mathcal{S}_n = \{x_i\}_{i=1}^n$ with n samples, the utility of $\mathcal{S}_n$ at k^{th} epoch follows from equation 3:

$$\mathcal{U}(\mathcal{S}_n, k) = a_k[(k-1)^b - k^b]n^b, \quad \text{where } a_k = a_0 \left(\frac{1}{2}\right)^{k/\tau} \tag{4}$$

Finally, given the training set $\mathcal{S}_n$, the final error of model $f(\mathcal{S}_n, k)$ after training on $\mathcal{S}_n$ for C total training samples can be written as:

$$\begin{aligned} f(\mathcal{S}_n, a, b, \tau, d, C) &= d + U(1)\delta + U(2)\delta^2 + \ldots + U(j)\delta^j + \ldots + a[C^b - (kn)^b]\delta^k, \\ U(j) &= a[(j-1)^b - j^b]n^b, \quad \delta = \frac{1}{2}^{1/\tau} \text{ and } k = \lfloor C/n \rfloor \end{aligned} \tag{5}$$

Again, observe that the utility of the repeated data (the training set $\mathcal{S}_n$) keeps on falling with the half-life decay factor.

3.4 Estimating the Utility of Mixture of Data Pools

A unique challenge posed by our problem formulation is the presence of multiple data subsets with different respective data utilities. In a scenario where we jointly train on multiple data subsets, how can we estimate the effective utility (recall b in equation 5 denotes the utility of a pool) of the combined pool? One naive way to estimate the error on training on multiple data mixtures would be to use the average error on them. However, this does not factor in the interplay of the two different b values in the exponent of the scaling curve. To address this, let us first consider a simpler problem of training on two data subsets $\mathcal{S}_n^P$ and $\mathcal{S}_n^Q$ with utility values b_P and b_Q respectively. To simplify the analysis we remove the terms a, c, d but it directly follows in their presence as well.

Theorem 1 *Given k data pools $\mathcal{S}_n^1 \ldots \mathcal{S}_n^k$, sampled uniformly at random with respective utility values $b^1 \ldots b^k$, the effective utility value b_{eff} for the combined pool is the arithmetic mean of the individual utility values. Formally, $b_{eff} = \frac{\sum_i^k b^i}{k}$.*

This theorem implies that when merging different data buckets of equal size, we can approximate the utility parameter of the aggregate bucket, b_{eff}, simply as the arithmetic mean of the utility parameter of each of the individual buckets.

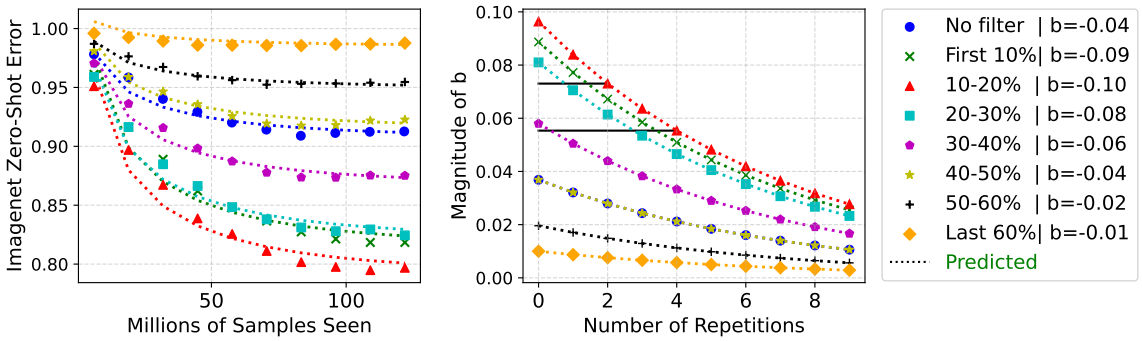


Figure 3: **Scaling curves with repeated data for visual-language models**: We partition the DataComp medium scale pool samples into various buckets, based on the CLIP scores, and train a model on each bucket for 10 epochs. (a) The estimated error curves using the proposed scaling laws (Equation 5). (b) Diminishing utility with epochs of various data subsets. Observe that the utility of the best bucket (red) at it's 4^{th} repetition becomes less than that of worse buckets like top-30%-40% subset.

4 FADU: Filtering by Assessing Diminishing Utility

Recall that we empirically observed in Figures 1a, 2b that the diminishing utility of repeated good data necessitates the need to adapt the aggressiveness of data filtering in accordance with the compute available. In this section, we use our proposed scaling laws to estimate the best thresholding strategy given any data filtering metric and training compute. We will use CLIP-score-based data ranking as a running example to demonstrate the same.

Applying our scaling laws requires us to transition from this metric-based example ranking to example utility. In order to do the same, we divide the dataset into multiple subsets ordered by the example ranking. This is based on the assumption that the utility of all examples in a small neighborhood (based on the metric's ranking) is similar. Now, we need to estimate the scaling curve parameters for various subsets of the data. Consider M equal disjoint data buckets $\mathcal{S}_1, \mathcal{S}_2 \ldots, \mathcal{S}_M$ of the training pool $\mathcal{S}$, ranked by quality.

We propose Filtering by Assessing Diminishing Utility (`FADU`) where we predict the best-filtered subset given a fixed training compute $\mathcal{C}$, which is the one with the highest average utility over the training duration. Specifically, our approach consists of two steps:

1. **Subset utility estimation**: We first train a model on each individual bucket (separately), and fit the test error with Equation 5 to estimate the initial pool parameters $b^1, b^2, \ldots, b^M$ and half-lives τ for the buckets.

2. **Error estimation for training on k buckets**: We sort all the buckets based on their utility, and then estimate the error if we jointly trained on `top-k%` data subsets. We first estimate the effective scaling parameters $a_{\texttt{top-k\%}}, b_{\texttt{top-k\%}}$ of the joint data, which is given by the arithmetic mean of the corresponding parameters of the subset pools (Theorem 7). Plugging in the estimated parameters in Equation 5, we estimate the final error of the pool as $\ell_{\texttt{top-k\%}} = f(\mathcal{S}_{\texttt{top-k\%}}, a_{\texttt{top-k\%}}, b_{\texttt{top-k\%}}, \tau, d, C)$. The goal is to find the value of k at which we should threshold. Therefore, the `top-k%` pool with the lowest $\ell_{\texttt{top-k\%}}$ is predicted as the best-filtered subset.

4.1 Empirical results: Estimating the utility of repeated data

To assess the utillity of data with repetitions, we again use the DataComp medium scale pool. Specifically, we form six distinct data subsets, categorized by their respective CLIP scores: top 10%, top 10%-20%, and so forth, up to the top 50%-60% subset. Each subset, approximately 12.8M in size, is used to train a model over 10 epochs and estimate the scaling law parameters. Figure 3 presents the estimated scaling curves for each data subset, including the scaling parameter b. The calculated half-life for these data subsets is approximately 3 epochs. We observed two significant trends:

- The estimated utility values b for subsets with higher CLIP scores are markedly lower (more negative), thereby supporting traditional data filtering methods. Interestingly, both empirical results and our scaling laws suggest that the 10%-20% CLIP score subset is more effective than the top 10%, a somewhat unexpected finding.
- While the utility of new data (depicted by the blue curve) and repeated data (other subset curves) both diminish over time, the decrease is more pronounced for repeated data. For instance, after four repetitions, the utility of the best subset pool (shown in red) becomes lower than that of the top 30%-40% CLIP score subset during its first repetition.

It's important to note that this observed diminishing utility is not an artifact of creating subset pools based on CLIP scores. This trend is consistently seen even with recent state-of-the-art data filtering methods like `T-MARS` (Maini et al., 2023), as detailed in Appendix I.

4.2 Predicting the Pareto Curve

Recall that the pareto-filtering threshold must be adapted based on the training compute as shown in Figures 1a, 2b. We now use `FADU` to estimate the optimal `top-k%` bucket based on the algorithm outlined in Section 4. First, we estimate the a, b parameters for different data buckets (each with 10% of data). The corresponding b values for each data pool are depicted in Figure 3. We then find the effective scaling parameters for each `top-k%` bucket and calculate the optimal value of k at compute scales of $\{32M, 128M, 640M\}$. `FADU` predicts that the optimal value of $k = \{1, 3, 4\}$ respectively in the case of CLIP-score based filtering. This precisely matches with the pareto-frontier of data filtering carved out in Figure 1a for the CLIP filtering algorithms. Note that, since the magnitude of b for the $10 - 20\%$ data bucket (when ordered according to CLIP score) is higher than that of the top 10% data bucket, `FADU` also correctly indicates that it is more beneficial to train on the second bucket.

5 Discussion

Despite recent efforts, the curation and utilization of data remains surprisingly ad-hoc and *hacky*, with very little predictability about the outcomes of a filtering strategy. In particular, all prior filtering approaches (i) propose a metric that ranks examples and filters out data points below a threshold; and (ii) are the thresholds are chosen 'agnostic' of the compute the model is supposed to be trained for. While well-resourced organizations can embark on exhaustive sweeps of 'filtering' parameters, this approach (i) is extremely expensive, especially in the paradigm of web-scale pre-training; and (ii) does not transfer to new training paradigms.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9453–9463, 2019.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, Oct 2017. ISSN 1558-2256. doi: 10.1109/jproc.2017.2675998. URL `http://dx.doi.org/10.1109/JPROC.2017.2675998`.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks, 2023.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop*, 2004.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023a.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023b.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

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

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer, 2021.

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.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL `https://doi.org/10.5281/zenodo.5143773`. If you use this software, please cite it as below.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

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, 2020.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

Anas Mahmoud, Mostafa Elhoushi, Amro Abbas, Yu Yang, Newsha Ardalani, Hugh Leather, and Ari Morcos. Sieve: Multimodal dataset pruning using image captioning models, 2023.

Pratyush Maini, Sachin Goyal, Zachary C Lipton, J Zico Kolter, and Aditi Raghunathan. T-mars: Improving visual representations by circumventing text feature learning. *arXiv preprint arXiv:2307.03132*, 2023.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*, 2023.

Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning, 2023.

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

O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, distillation, and hard negatives for vision-language pre-training. *arXiv:2301.02280*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763, 2021a.

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, 2021b.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, 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.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021.

Hu Xu, Saining Xie, Po-Yao Huang, Licheng Yu, Russell Howes, Luke Zettlemoyer Gargi Ghosh, and Christoph Feichtenhofer. Cit: Curation in training for effective vision-language data. *arXiv preprint arXiv:2301.02241*, 2023.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL `https://aclanthology.org/Q14-1006`.

Haichao Yu, Yu Tian, Sateesh Kumar, Linjie Yang, and Heng Wang. The devil is in the details: A deep dive into the rabbit hole of data filtering, 2023.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2020.

Appendix A. Detailed Discussion

Scaling the scaling curves Past work on scaling laws for CLIP models (Cherti et al., 2023) trained tens of models at varying compute scales ranging from 3B to 34B training samples and models spanning different ViT families. While training models at this compute is extremely expensive, we utilize their pre-trained models. Past works tried to fit scaling laws for this family of models, but the scaling curves showed extremely high errors for models trained on small datasets. We believe this is primarily because they do not account for the impact of diminishing utility of repeated data. We use our proposed scaling laws to estimate errors for the models in question. The revised scaling trends are presented in Figure 4, which are able to predict the error with a much higher accuracy than the past scaling curves without repetitions, as shown in Appendix H. This confirms that our scaling laws hold at massive models trained for 34B data compute, indicating that the diminishing utility of repeated data must indeed be accounted for while predicting model training outcomes.

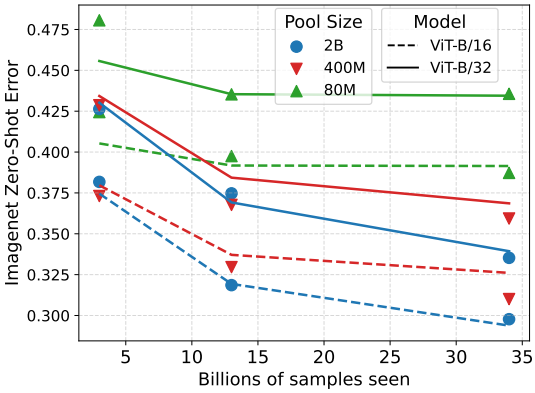


Figure 4: Similar to Figure 5, our scaling law accurately predicts the final error for models trained on 2 different architectures, 3 different pool sizes and 3 different compute budgets.

State of Data Curation Despite recent efforts, the curation and utilization of data remains surprisingly ad-hoc and *hacky*, with very little predictability about the outcomes of a filtering strategy. In particular, all prior filtering approaches (i) propose a metric that ranks examples and filters out data points below a threshold; and (ii) are the thresholds are chosen 'agnostic' of the compute the model is supposed to be trained for. While well-resourced organizations can embark on exhaustive sweeps of 'filtering' parameters, this approach (i) is extremely expensive, especially in the paradigm of web-scale pre-training; and (ii) does not transfer to new training paradigms.

State of Scaling Laws In the paradigm of language modeling, recently Muennighoff et al. (2023) made first attempts at investigating the diminishing utility of data as we repeat over it. Our work builds on these insights, but with one crucial distinction—prior work assumes that web data is homogenous and has uniform utility. However, data curation builds on the fundamental observation that different subsets of web data have different utility. In our work, we highlight a crucial insight regarding the implication of training steps on data utility of non-homogenous datasets. We hope our work lays the foundations for developing data curation as a methodological science where curation decisions can accurately predict model training outcomes.

Appendix B. Related Work

Data Filtering Vision-language models are trained on noisy webscale datasets, making data filtering a crucial precursor. OpenCLIP (Ilharco et al., 2021) tried to reproduce the performance of OpenAI's CLIP (Radford et al., 2021b) by curating LAION-400M (Schuhmann et al., 2021) dataset. However, their performance still lagged that of CLIP, suggesting the importance of DataCuration. Recently, Datacomp (Gadre et al., 2023a) streamlined the efforts in this direction by releasing a well-crafted benchmark challenge for subset selection from common crawl.

Most of the state-of-the-art data curation approaches involve ranking the data using some metric. For example, LAION (Schuhmann et al., 2021, 2022) uses a CLIP score based filtering (amongst many other rules), where samples with a image-caption similarity score lower than 0.28 (as assesed by a pretrained CLIP) are filtered out. Mahmoud et al. (2023); Nguyen et al. (2023) propose to use synthetic-captions generated by an image captioning model (Li et al., 2023) to rank the data. Recently, `T-MARS` (Maini et al., 2023) and CAT (Radenovic et al., 2023) highlighted that a large fraction of images in these webscale datasets lack any learnable "visual" features, and have high similarity with the caption only due to text in the images (OCR) matching the caption. They propose to filter out 50% of the data based on the CLIP similarity scores after masking the text using an OCR detection algorithm. Similarly, C-SSFT (Maini et al., 2023) and DFN (Fang et al., 2023) propose filtering out mislabeled samples by assessing the drop in CLIP scores when finetuning a pretrained CLIP on a held-out validation set. Some other works include Yu et al. (2023) which uses a mixture of rules and Xu et al. (2023) which uses similarity with downstream metadata.

In this work, we highlight why data filtering cannot be agnostic to training compute and how the ordering varies as one changes the training paradigm. Infact, we showcase LAION filtering (used to train state-of-the-art OpenCLIP models) can even be sub-optimal to no-filtering or training on the raw common crawl under certain settings.

Scaling Laws in Language Modeling One of the most salient trends in recent deep learning research is the observation that neural network performance often improves predictably with an increase in model size, data size, and computation. In the domain of language modeling, such observations have been systematized into a set of principles known as *scaling laws*. Kaplan et al. (2020) conducted a comprehensive study on scaling laws for neural language models. They observed that, given fixed computational budgets, there exists an optimal model size, training data size, and training time. Interestingly, the triple (model size, data size, batch size) tends to scale in a roughly lock-step manner, reinforcing the notion that larger models require more data and more computation to be trained effectively. This observation is corroborated by Hoffmann et al. (2022); Hernandez et al. (2021) who delve deeper into training compute-optimal language models and highlight the importance of balancing computation with model and data sizes.

Most closely related to our work, recently Muennighoff et al. (2023) show that training on tokens beyond 4 epochs yields negligible gains compared to training on new language data. They model this by proposing an "effective data size" which decreasing with repetitions. Our work (in the vision-language domain) highlights why such a characterization is

Table 1: Scaling curves give us the ability to estimate the utility of data subsets at various stages of training. We compare various curriculum based training strategies with the baseline of approach of uniform training from the best bucket. Our observations indicate that the baseline of approach of uniform sampling from the best top-k% bucket works the best, opening up interesting directions for future work.

Curriculum Methods	128M Compute		640M Compute	
	Imagenet	Avg.	Imagenet	Avg.
Baseline (Best CLIP)	27.3%	24.3%	39.0%	46.1%
Greedy	26.3%	23.4%	36.9%	44.9%
Smooth L→R	26.8%	23.9%	38.6%	45.7%
Smooth R→L	27.2%	24.1%	38.9%	45.9%

not optimal, as the webscale data is not homogeneous and does not have a uniform utility distribution.

Scaling Laws in CLIP Application of scaling laws to models like CLIP is still an area of active research. As with the scaling laws observed in pure language models, there's an indication that as the model and data sizes for CLIP grow, its performance on downstream vision tasks improves, albeit with diminishing returns (Schuhmann et al., 2022; Gadre et al., 2023b). Cherti et al. (2023) try to fit standard scaling curves similar to Kaplan et al. (2020) on CLIP models of varying size and architecture. However, note that contrary to language models which are rarely trained with more than 3-4 epochs, CLIP training invovles upto 30-40 epochs even at the largest data scale. As we highlight in this work, one needs to model the diminishing gains of data with repeated epochs, in order to accurately estimate scaling curves for visual-language model training.

Appendix C. Curriculum learning

One main implication of our findings is that we need the data filtering strategy to be compute aware. Our proposed algorithm FADU is one simple way to do this.

Going one step beyond, the ability to model precise utility of data points depending on number of repetitions should confer the ability to perform curriculum learning. FADU treats all filtered samples equally. In principle, we should recognize the heterogeneity in the quality of samples and try to train more steps on higher quality samples and fewer steps on lower quality samples.

We perform an initial experiment to test this where we discretize our initial unfiltered pool into several buckets where each bucket has data of roughly the same "quality" (for e.g. as measured by the CLIP score). We compute the (diminished) utilities for each bucket using our scaling law (Equation 5) accounting for the number of times each bucket was seen so far. We now consider two curriculum learning approaches:

- **Greedy Curriculum**: We pick the bucket with the maximum (diminished) utility and make a pass over the entire bucket. We then recompute the new diminished utilities and repeat this process, until the compute budget is exhausted. This is a

simple greedy version where at any given point, training is performed on the bucket with the highest utility at each point.

- **Smooth Curriculum**: We also consider two other variants of curriculum training, which we call smooth curriculum learning. We first identify all the top k buckets in a greedy way in the data pool and calculate the number of repetitions for each of them (i.e. the number of times they occur in top-k). We then simply train over all the buckets, removing the buckets as their number of repetitions gets exhausted. We call this approach Smooth L→R curriculum. We also explore a reverse version of the same, where we train on the best bucket with the highest number of iterations first, and then keep on adding lower-quality data buckets (which are still in top-k) to the training pool.

Table 1 compares curriculum learning based on the utility values with the baseline. We see that greedy curriculum learning approach actually does worse than the less sophisticated approach where we treated all samples equally. This does not directly contradict our model of utility, but exposes a nuance that future work on curriculum learning should handle. Our model of utility assumes there is no distribution shift as we train the model for different epochs. However, curriculum learning changes this! For example, switching from a higher quality data pool to a lower quality pool after a few epochs on the higher quality pool exposes the model to a distribution shift which makes the training unstable. Furthermore, if there is a continuous distribution shift while training, the models might "forget" what it learnt from the initial distribution of high quality data and retain mode from the low quality data it sees at the end of training.

Appendix D. Proof of Theorem 1

We restate the Theorem 1 again here.

Theorem 1 *Given k data pools $\mathcal{S}_n^1 \ldots \mathcal{S}_n^k$, sampled uniformly at random with respective utility values $b^1 \ldots b^k$, the effective utility value b_{eff} for the combined pool is the arithmetic mean of the individual utility values. Formally,*

$$b_{\mathit{eff}} = \frac{\sum_i^k b^i}{k} \tag{6}$$

Proof

Consider the case when $k = 2$. Let y denote the error of model after seeing n samples from the two pools. For simplicity, we assume $y = n^b$, ignoring the constant a, τ, d, but the proofs follows otherwise as well. From equation 1, we have:

$$y = n^b; \quad \frac{dy}{dn} = bn^{b-1} = b\frac{n^b}{n} = y\frac{b}{n} \tag{7}$$

Now, consider that we sample two times from $\mathcal{S}_n^P$ and $\mathcal{S}_n^Q$ respectively. Let $n_1 = n + 1$ and $n_2 = n + 2$ denote the total samples seen after the model is trained on the two random

draws. From equation 7, we have:

$$\frac{dy_1}{dn_1} = y_1 \frac{b_P}{n_1}, \quad y_1 = y + y_1 \frac{b_P}{n+1} \tag{8}$$

$$\frac{dy_2}{dn_2} = y_2 \frac{b_Q}{n_2}, \quad y_2 = y_1 + y_2 \frac{b_Q}{n+2} \tag{9}$$

Given that $n_1, n_2 \gg 1$, and $y_1 \sim y_2 \sim y$, we have:

$$y_2 \approx y + y \frac{b_P}{n+1} + y \frac{b_Q}{n+2} \tag{10}$$

Simplifying further, we obtain:

$$\frac{(y_2 - y)}{2} \approx y \frac{b_P + b_Q}{2n}, \quad \frac{dy}{dn} \approx y \frac{b_P + b_Q}{2n} = y \frac{b_{\text{eff}}}{n} \tag{11}$$

Thus, this analysis demonstrates the linearity of the combined utility values b_P and b_Q when two different data pools are sampled uniformly at random. Therefore, we can conclude that for a set of k data pools with b as the exponent and weight values, each governed by $b_i \forall i \in [1 \dots k]$, $b_{\text{eff}} = \frac{\sum_i^k b_i}{k}$. In addition to observing the linearity of b values, we empirically also find that a values also follow a similar linearity, as further discussed in Appendix F. ■

Appendix E. Downstream Evaluation Datasets

Following prior work (Radford et al., 2021a; Wortsman et al., 2021), we evaluate our models on a variety of image classification and retrieval datasets to assess their zero-shot capabilities. While the Datacomp (Gadre et al., 2023a) benchmark averages performance across 38 different datasets, we use a subset of 18 such datasets where medium-scale models give better than random performance in order to be able to develop reliable scaling laws. More specifically, we select the following datasets:

1. ImageNet: a 1000-class image classification challenge (Russakovsky et al., 2015).
2. ImageNet-OOD: Six associated Imagenet distribution shifts—ImageNet-V2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2020), ImageNet-A (Hendrycks et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-O (Hendrycks et al., 2019), and ObjectNet (Barbu et al., 2019).
3. VTAB: 6 out of 12 datasets from the Visual Task Adaptation Benchmark (Zhai et al., 2020), including Caltech-101 (Fei-Fei et al., 2004), CIFAR10 (Krizhevsky, 2009), CIFAR100 (Krizhevsky, 2009), Oxford Flowers-102 (Nilsback and Zisserman, 2008), Oxford-IIIT Pets (Parkhi et al., 2012), and RESISC45 (Cheng et al., 2017).
4. Additional classification datasets: Food-101 Bossard et al. (2014), Pascal VOC 2007 (Everingham et al.), and Stanford Cars (Krause et al., 2013).
5. Retrieval: 2 retrieval tasks of MSCOCO (Chen et al., 2015) and Flickr (Young et al., 2014).

Buckets (by CLIP score)	'a'	'b'
Top 10%	1.27	-0.09
Top 10%-20%	1.29	-0.10
Top 20%-30%	1.22	-0.08
Top 30%-40%	1.12	-0.06
Top 40%-50%	1.04	-0.04
top 50%-60%	0.96	-0.02
Last 40%	0.94	-0.01
Mean (Estimated)	1.07	-0.04
No filter (Actual)	1.03	-0.04

Table 2: **Linearity of scaling curve parameters:** The scaling curve parameters show a linear interpolation while mixing buckets, empirically as well. For example, the (weighted) mean of parameter 'a' over the various buckets is 1.07, which closely approximates the parameter 'a' for the whole data i.e. no filtering.

Appendix F. Linearity of Scaling Curve Parameters

In Section 3, we proved that the scaling curve parameter 'b' can be linearly interpolated when working with a mixture of distributions. In this section, we empirically show that (i) the linearity of 'b' indeed holds, and (ii) the normalization parameter 'a' also respects a similar linearity property.

Recall that in § 4.1 and Figure 3, we estimated the utilities of various data buckets based on the CLIP score (Table 2). Now, if the scaling curve parameters 'a' and 'b' follow a linear interpolation when mixing the various buckets, the mean of these scaling parameters (weighted mean to be precise since one of the bucket is last-40%, which has 4x more data) over the individual top-k% score based buckets should be same as the scaling parameters estimated for no filter training. Empirically, we indeed observe the same (Table 2). For example, the (weighted) mean 'a' over the various clip score buckets is 1.07, whereas the actual 'a' for no-filter data pool was 1.03, which is an error of less than 4%.

Appendix G. Downstream Evaluation Metrics

As detailed in Appendix E, most of the evaluation datasets constitute image-classification tasks. We use the 'Accuracy metric' to evaluate the zero-shot performance of the model on these datasets. The only exceptions include:

1. VTAB: We report 'Mean per Class Recall' for Caltech-101 (Fei-Fei et al., 2004), Oxford Flowers-102 (Nilsback and Zisserman, 2008), Oxford-IIIT Pets (Parkhi et al., 2012) datasets. This follows the standard evaluation protocol in past benchmarks (Gadre et al., 2023a) and is done because of the large number of classes in these datasets.

2. Retrieval: For all the retrieval datasets we report the 'Mean Recall @ 1' which tells how probable is it for the top-recall entry to be relevant.

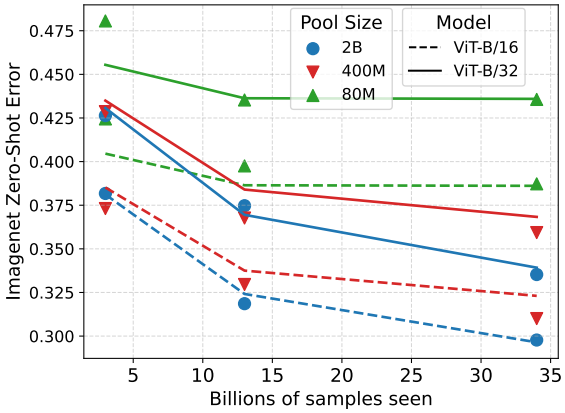


Figure 5: Similar to Figure 6, we use the scaling laws in Muennighoff et al. (2023) to predict the final error for models trained on 2 different architectures, 3 different pool sizes and 3 different compute budgets.

Appendix H. Comparing with Effective Dataset Size based Scaling Laws

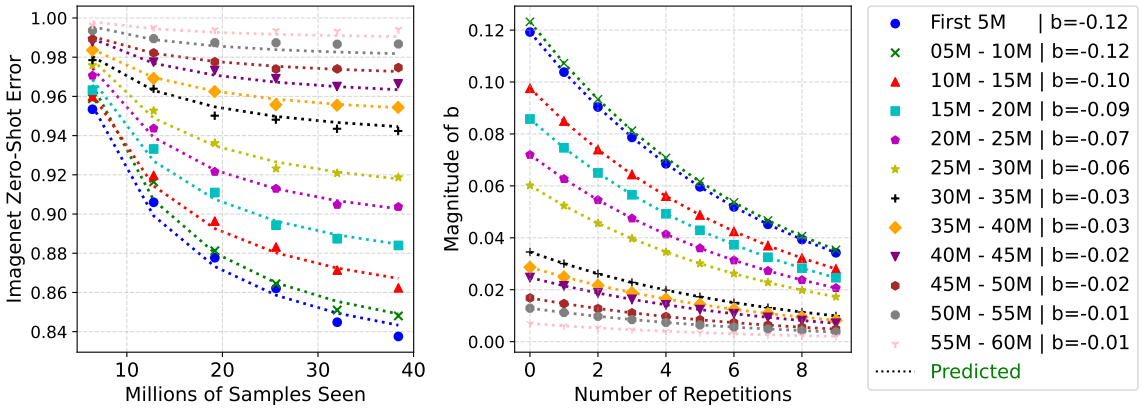


Figure 6: **Scaling curves with repeated data for visual-language models**: We partition the DataComp medium scale pool(128M) samples into various buckets, based on the `T-MARS` scores, and train a model on each bucket for 10 epochs. (a) The estimated error curves using the proposed scaling laws (Equation 5). (b) Diminishing utilities with epochs of various data subsets. Observe that due to repetitions, even the utility of the best bucket (blue curve) at it's 2^{nd} repetition becomes lesser than that of worse buckets like top 10-15M subset at it's 0^{th} epoch. This once again highlights why one needs to adapt the filtering aggressiveness with compute.

Recall the scaling law based on diminishing utility formulated in our work given by Equation 5. While we consider that the utility of each subsequent diminishes with a given half-life, recent work by Muennighoff et al. (2023) considered that the effective data size decays with an empirically estimated half-life. While similar in spirit, the former formulation provides a natural way of understanding how mixtures of data pools should interact with each

other. For completeness, we describe their scaling law below, and then compare their law written in the context of language modeling on the task of image-language CLIP training.

$$f(\mathcal{S}_n, a, b, \delta, d, C) = d + a\ C_{\text{eff}}^b \tag{12}$$

$$C_{\text{eff}} = d + n\delta + n\delta^2 + \ldots + n\delta^j \tag{13}$$

$$+ \ldots + (C - k \cdot n)\delta^{(k-1)} \tag{14}$$

$$k = \lfloor C/n \rfloor,$$

where C is the total number of training samples seen, n is the number of samples in the dataset, k is the number of repetitions of data. δ denotes the fractional decay of the effective data size at each subsequent epoch.

Now, we compare the error in the estimates by the formulation derived in our work as opposed to that in Muennighoff et al. (2023). We depict the estimated values based on Equation 12 in Figure 5. In the case of ViT-B-16 model, the ℓ_2 error between the true and the estimated Imagenet zero-shot accuracies is $8.15e^{-4}$ v/s $9.31e^{-4}$ resulting in a 14% error reduction.

Appendix I. Additional Scaling Curve Results

We presented the scaling curves along with their parametric estimates in Figure 3 for various data buckets based on CLIP score. Here, in Figure 6 we show similar curves for various data buckets based on the `T-MARS` scores. Again, we observe that while the initial scalaing parameters like 'b' for the best data buckets are high, they diminish quite rapidly, even becoming lower than that of worse buckets' parameters at the first repetition.

$$U(j) = \frac{a[(j-1)^{-b} - j^{-b}]}{n_u^b} \tag{15}$$

$$L(N) = d + \frac{a}{n^b} \tag{16}$$

$$L(N) = d + U(1)\delta + U(2)\delta^2 + \tag{17}$$

$$...U(j)\delta^j + ... + U(k)\delta^k \tag{18}$$

$$\tag{19}$$