Scaling Laws for Data Filtering—
Data Curation *cannot* be Compute Agnostic

Sachin Goyal[*†] Pratyush Maini[*†]
Zachary C. Lipton[†] Aditi Raghunathan[†] J. Zico Kolter[†,‡]
Carnegie Mellon University[†] Bosch Center for AI[‡]
{sachingoyal,pratyushmaini,zlipton,raditi,zkolter}@cmu.edu

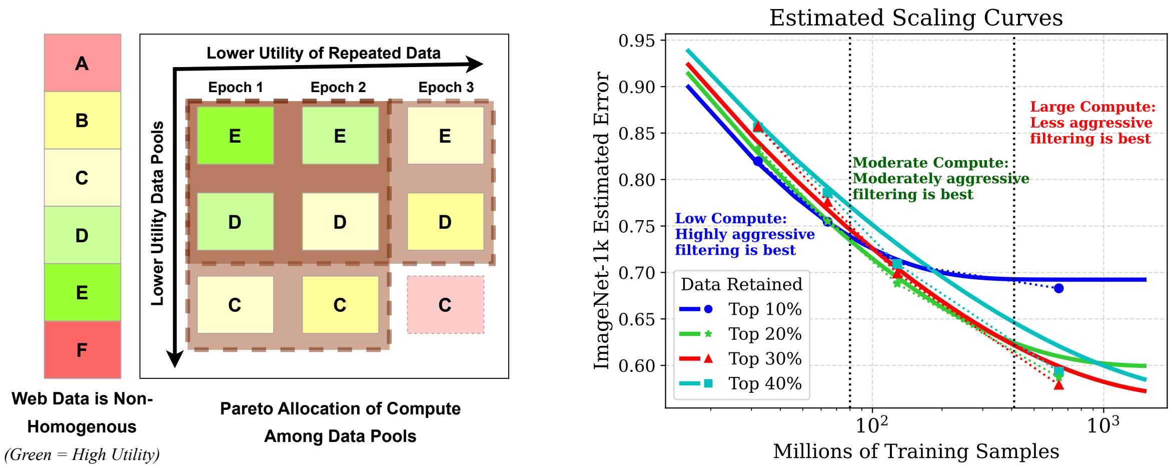

Figure 1. **(a) The Dynamic Problem of Data Filtering**: Web data is non-homogenous, and past work has proposed metrics that ranking various data subsets according to their diminishing quality (y-axis). However, training on 'high-quality' data for multiple epochs leads to diminishing utility (x-axis), an angle ignored in past work. Assume we have compute equivalent to 6 data pools, one could train on the best pool (E) for 6 epochs, or train on the best two pools (E and D) for 2 epochs each, and so on. Our work aims to answer–*what is the best allocation of computational resources in such scenarios?* **(b) Data Filtering Scaling Laws:** Our work proposes scaling laws for predicting the model performance on mixtures of data pools of various quality. Note that we do not train on data mixtures to fit the above scaling curves (scatter points are test points), rather the scaling curves are estimated from the scaling parameters of individual pools.

Abstract

Vision-language models (VLMs) are trained for thousands of GPU hours on carefully selected subsets of massive web scrapes. 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. 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 laws that characterize the* differing *'utility' of various data subsets, and accounting for how this diminishes for 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. Even without ever jointly training on multiple data buckets, our scaling laws enable us to estimate model performance under this dynamic trade-off between quality and repetition. This allows us to curate the best possible pool for achieving top performance on Datacomp at various compute budgets, carving out a pareto-frontier for data curation.*

[*] Equal Contribution.

1. Introduction

Machine learning has evolved from supervised training on small carefully labeled datasets to training on massive scrapes of the web, usually collected from data sources such as Common Crawl [1]. However, since web-scale datasets are noisy, they are generally curated to extract 'high quality' informative data points. For example, LAION dataset [42, 43] is a carefully curated dataset which retains just 10% of the original web-crawled data, by keeping only the image-caption pairs with a high CLIP similarity scores (along with some other rules). Later approaches developed more sophisticated filtering methods like T-MARS [31] which ranks the data based on CLIP score after masking the text or Fang et al. [11], Maini et al. [31] which rank the data based on the drop in CLIP scores after the model if finetuned on some held-out validation data. Note that all these data curation approaches involve ranking the data using some metric and then carefully choosing a threshold score, below which the samples are filtered out. Visual Language Models (VLM's) like CLIP are then generally trained for multiple epochs on these curated datasets [14].

In this work, we first show that data curation cannot be agnostic to compute. Specifically, **when training for large compute (large epochs) one needs to filter less aggressively as compared to when training for small compute**. For example, we show that there exist settings when training on aggressively filtered LAION dataset [42, 43] is actually worse than naively training on unfiltered raw data from the common crawl. This is because, after multiple repetitions, the high-quality filtered data has negligible remaining utility. On the other hand, the low-quality data samples, though lower in initial utility, are seen fewer times and hence have a higher utility towards the end. In other words, the utility of data diminishes with repetition, and hence filtering metrics must be designed by assessing the trade-off between the diminishing utility of a small pool of 'high-quality' data, and the initially lower but slowly diminishing utility of a larger pool that includes 'lower-quality' data.

Given the large variance in computational budgets one might have, identifying the ideal data filtering threshold poses a challenge. A straightforward, yet computationally prohibitive method would involve training models with datasets curated at different thresholds. To circumvent this, we leverage scaling laws to predict the performance of models trained with optimal filtering strategies.

Web data is inherently heterogeneous, consisting of data pools with varying levels of quality. However, current scaling law research tends to model web data using a unified set of scaling parameters, which is problematic considering these parameters represent critical dataset characteristics such as quality and diversity. In this study, we introduce the **first scaling laws tailored for heterogeneous web data, enabling the prediction of models trained with data mixtures of diverse quality pools under repetition**. Crucially, our approach does not rely on training with all possible data mixtures to determine scaling parameters. Instead, we estimate these parameters for any given data mixture by utilizing those of the individual pools.

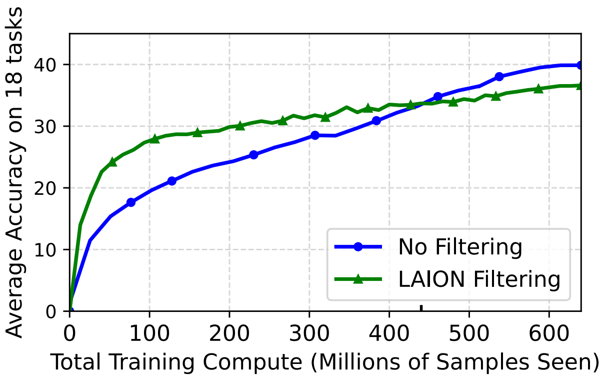

Figure 2. Given an initial data pool of 128M samples, we train ViT-B/32 CLIP models for a total of 640M samples. As we train for longer, the accuracy gains on the LAION data subset that filtered the common crawl to 10% of its initial size plateau. Surprisingly, even no-filtering of the common crawl is better than the popular LAION dataset after seeing more than 450M samples.

Empirically, we show that our scaling laws for heterogeneous webdata allow us to predict the pareto-optimal filtering strategy at various compute budgets ranging from 32M to 640M, using the medium scale pool (128M samples) of DataComp [14]. Finally, we also validate our scaling curves at extremely large scale up to 34B compute using pre-trained publicly available checkpoints.

2. Related Work

Data Filtering Vision-language models are trained on noisy webscale datasets, making data filtering a crucial precursor. OpenCLIP [23] tried to reproduce the performance of OpenAI's CLIP [38] by curating LAION-400M [42] dataset. However, their performance still lagged that of CLIP, suggesting the importance of DataCuration. Recently, Datacomp [14] 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 [42, 43] 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. [30], Nguyen et al. [33] propose to use synthetic-captions generated by an image captioning model [29] to rank the data. Recently, `T-MARS` [31] and CAT [36] 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 [31] and DFN [12] 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. [48] which uses a mixture of rules and Xu et al. [46] 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. [26] 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 Hernandez et al. [20], Hoffmann et al. [21] who delve deeper into training compute-optimal language models and highlight the importance of balancing computation with model and data sizes. Sardana and Frankle [41] propose modifications to incorporate the inference cost as well into the scaling laws. Bahri et al. [3], Hutter [22] theoretically study neural scaling laws.

Most closely related to our work, Muennighoff et al. [32] show that training on tokens beyond four epochs yields negligible gains compared to training on new language data due to diminishing utility. However, they do not consider the case of different data quality pools. In this work, we how that mixture of data pools cannot be modeled with an effective dataset size formulation of Muennighoff et al. [32]. Crucially, one needs to model a decay in utility factor (the scaling parameter b in $y = an^b$) as well.

Finally, Hashimoto [17] as well study scaling laws for various mixture proporations, but their study is limited to small scale supervised learning tasks. In this work, we focus on scaling laws for large scale contrastive training of visual language models like CLIP.

Scaling laws for downstream performance Although traditionally the scaling laws have focused on modeling the training loss, recent works have started directly modeling the downstream performance [16, 24]. Alabdulmohsin et al. [2], Caballero et al. [6] propose some amendments to estimate downstream performance on image classification and machine transalation tasks respectively. In this work, we model ImageNet zeroshot accuracy and an average performance over 18 tasks from DataComp [14] to fit the scaling curves for data filtering.

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 [15, 43]. Cherti et al. [9] try to fit standard scaling curves similar to Kaplan et al. [26] 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.

3. Data Filtering for a Compute Budget

3.1. 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 of the recently data curation benchmark, Datacomp [14]. The pool contains 128M samples. 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. For example, at a compute budget of 128M, each sample in a filtered pool of 12.8M samples would be seen 10 times.

We assess the performance of our models based on their zero-shot performance 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 B.

3.2. When “good” data performs worse

We start with the popular LAION filtering strategy used in obtaining the LAION dataset [42, 43]. 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 2 shows the average downstream accuracy on 18 tasks (Section 3.1), 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 budget**: The advantage offered by data filtering consistently diminishes as we increase our compute budget. Remarkably, beyond 450M iterations, training on the unfiltered common crawl dataset outperforms that on LAION-filtered data.

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?

LAION-filtering retains around 10% of the data pool, hence at around 450M compute budget, each sample from the LAION-filtered pool is seen around 32 times. The key insight here is that the same sample, as it is seen multiple times over training, offers a diminishing utility each additional time. The LAION-filtered pool has higher initial utility, which does not degrade much at a low compute budget where samples are not repeated too often. However, at a high compute budget, the utility of the LAION-filtered pool diminishes substantially as the samples are repeated multiple times. Eventually, the unfiltered samples, though starting off with a lower utility, end up suffering a smaller drop in utility as they are repeated less often, even outperforming “high-quality” LAION-filtered data at some point.

Remark. In Theorem 1 we will later show that the rate of decay of the utility of a pool is influenced by the size of the pool. In particular, because these models are trained with a contrastive objective offering $O(n^2)$ unique comparisons for a dataset of size n, changing the pool size by a factor of k, actually ends up increasing the total comparisons by k^2. This could potentially mean that the reversal point for LAION v/s no filtering happens much later than 40 epochs when the data pool size is increased by a factor of 10. That said, the insight from this section underscores the need to tailor the filtering approach to the model’s total training compute, challenging existing practices and offering a new direction for optimizing model performance.

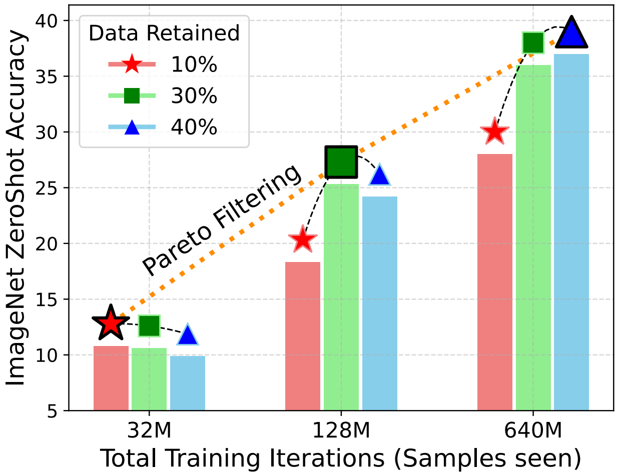

Figure 3. We vary the CLIP filtering threshold after ranking the data by their metric. While the original paper proposed retaining 30% of the data, our results show that depending on the ratio of compute to data pool size, we must adaptively make the filtering less (or more) aggressive to account for 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.

We expand upon modeling and estimating this decay of utility in Section 5, which one can then use to *adaptively* filter the dataset based on the available compute budget.

3.3. 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. Is this something specific to the LAION-filtering method, or does our intuition about diminishing utility of repeated samples hold for other filtering approaches as well?

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, utilizing the CLIP L/14 model, and (b) `T-MARS` , which ranks data based on CLIP scores after masking text (OCR) features in images (refer to Section 2). We compare four levels of varying aggressive filtering for each data filtering approach, and vary total compute (training iterations) from 32M to 640M, just like before.

Figure 3 illustrates the comparison of Top 10-20%, 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 10-20% data as per CLIP scores, yields the best results, while the least aggressive Top 40% filtering approach performs the worst. However, this trend *reverses entirely as the compute is scaled to 640M* . While retaining 10% data 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 10). Here, retaining Top 20% data, as originally proposed, stops being optimal as the compute scale increases, and less aggressive approaches prove to be more effective. These observations underscore the need for a compute aware filtering strategy that balances the high initial utility of high-quality data which quickly diminishes with repetitions, and the lower-quality but larger data that offers lower initial utility but a slower decay due to fewer repetitions.

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. In Section 6, we explore how to *effectively extrapolate* from smaller compute budgets to larger while accounting for diminishing utility with repetition.

4. Scaling Laws for Data Filtering

4.1. Defining Utility

Past works on scaling laws [25, 26] estimate the error of a model (at a given parameter count) after training for n samples as: $y = an^b + d$, where $a, d > 0$ and $b < 0$ are constants to be determined empirically, and y is a performance metric such as the loss of the model on a validation set. 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. For instance, Cherti et al. [9] noted that the b value for OpenAI's filtered dataset was lower than that of the LAION dataset, indicating it had higher utility. Whereas, a is a normalizer and d estimates an irreducible error at the end of training to infinity. Rather than estimating the loss at the end of training for n samples, we can also consider the instantaneous utility of a sample at any point during training. This is given by:

$$\frac{dy}{dn} = a \cdot bn^{b-1} = \frac{y}{n} b. \tag{1}$$

This equation shows that the instantaneous utility of a sample is proportional to the current loss and inversely proportional to the number of samples seen so far. This is intuitive as the utility of a sample decreases as more data is seen.

4.2. Utility under Repetition

Now, let us add one more complexity to this scaling law from past works. In practice, CLIP style pre-training is done by repeating multiple epochs of training on the same data [14]. However, there is no clear understanding of how the utility of a sample changes with repetition. We hypothesize that this utility decays exponentially with the number of times the sample is seen. More formally, the utility parameter (b) of a sample seen $k+1$ times is given by:

$$b_{k+1} = b \cdot \left(\frac{1}{2}\right)^{\frac{k}{\tau}} = b \cdot \delta^k \tag{2}$$

where τ is the half-life of the utility parameter. A higher value of τ indicates that the utility of a sample decays slower with repetition. δ more concisely captures the decay in utility with repetition, and is used for simplicity of notation. Then, a closed form expression of the loss of a model after seeing n samples k times each is given by:

$$y_k = a \cdot n_1^{b_1} \prod_{j=2}^{k} \left(\frac{n_j}{n_{j-1}}\right)^{b_j} + d \tag{3}$$

where n_j is the number of samples seen at the end of j^{th} epoch of training. The equation is derived in Appendix F.1 and forms the basis of our scaling law.

Summary of Parameters Let us concisely summarize the role of each of the parameters in our scaling laws (Eq. 3) in order to develop better intuition about each of them.

1. **Utility Parameter** (b): The change in loss scales with the number of samples seen exponentially based on the value of b. A high quality data bucket will have a lower b value compared to a worse data bucket.
2. **Half life** (τ): The repetition parameter captures the decay in the utility of repeated data. Intuitively, the half life τ captures the diversity of the data bucket. Data buckets with high diversity will have a higher value of τ, allowing more repetitions of the bucket, as one would desire.
3. **Decay Parameter** (δ): The decay parameter is a parameter directly derived from τ, and not a unique parameter. We use this for simplicity of notation. δ captures the fractional decay in the utility parameter with one epoch of training on that data.
4. **Normalizer** (a): The normalizer aims to capture an intrinsic property of the task allowing us to relate the change in loss with the number of samples seen. This does not change with the bucket. We learn a common value of a that minimizes the loss for all buckets, and treat it as a fixed constant across all buckets.
5. **Irreducible loss** (d): This is a constant parameter added to the loss that can not be reduced further.

4.3. The case of heterogeneous web data

Now we are ready to add the final layer of complexity to our scaling laws, that of heterogeneous data. A unique challenge in the paradigm of webdata, and critically missed in the existing works on scaling laws, is the presence of data pools of different quality. As discussed in Section 3, web-data can generally be partitioned into multiple subsets (like using clip score), each with it's own respective scaling parameters (like respective data utility parameter b).

Training large scale models then involves jointly training on a mixture of multiple data buckets. This brings us to us central question—*how can we estimate the loss and thus the scaling curves for a mixture of pools effectively*? This ultimately allows to curate the data conditional to any compute, rather than a static curation. 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, and how does the repetition parameter (τ) change with increasing data mixtures.

Theorem 1. *Given p data pools $\mathcal{S}_n^1 \ldots \mathcal{S}_n^p$, sampled uniformly at random with respective utility, repetition parameters $(b_1, \tau_1) \ldots (b_p, \tau_p)$, then the new repetition half-life of each of the buckets $\hat{\tau} = p \cdot \tau$. Additionally, the effective utility value for the combined pool b_{eff} for the combined pool at the k^{th} repetition is the weighted mean of the individual utility values. Formally,*

$$b_{eff}^{(k)} = \frac{\sum_i^p b_i \hat{\delta}_i^k}{p}, \tag{4}$$

where $\hat{\delta}_i = \left(\frac{1}{2}\right)^{1/\hat{\tau}}$, the new decay parameter per bucket.

We refer the reader to Appendix F.1 for the derivation of the formulae for b_{eff}. We assume that the utility of a sample decays exponentially on being seen multiple times. However, there is one major challenge of contrastive training paradigm, where the effective number of samples in a data pool of size N is N^2. This is because each sample is paired with every other sample in the data pool. In Appendix G we show that $\hat{\tau} = \frac{\hat{N}}{N}\tau$ where $\hat{N}$ is the total number of samples in the data pool and τ is the half life.

4.4. Various other formulations

While deciding the scaling laws for data filtering, we considered various other formulations. This included scaling laws that modeled the decay in 'effective samples' [32] rather than effective utility. We describe various design considerations and why they were not chosen in Appendix F. Further, we also study various choices such as the need for allowing different data buckets to have different 'half lives' (τ), but a unified normalizer (a), and the way we optimized various scaling parameters in Appendix E.

5. Results: Fitting scaling curves for various data utility pools

Experiment Setup: We experiment on the DataComp medium scale pool which consists of 128M image-caption pairs. In this work, we use T-MARS [31] score and CLIP score as the two data utility estimates and rank the web-data based on them. Specifically, we form four distinct data subsets, categorized by their respective T-MARS (or CLIP) scores: top 10% (10% datapoints with the highest scores), top 10%-20%, and so forth, up to the top 30%-40% subset. Each subset, approximately 12.8M in size, is then used to train a model for over 10 epochs. Finally, we estimate the parameters for the scaling curve with repeated data (Eq, 3), by fitting over the obtained downstream zeroshot error on ImageNet or an average performance over 18 visual classification and retrieval tasks (Appendix B).

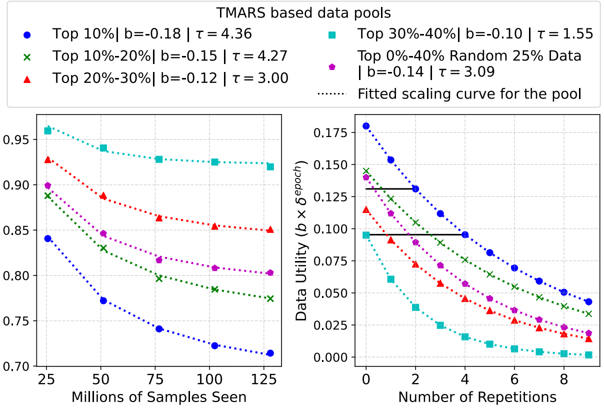

Figure 4. **Scaling curves with repeated data for visual-language models**: We partition the DataComp medium scale pool(128M) samples into various buckets, based on the CLIP scores, and train a model on each bucket for 10 epochs. (a) The estimated error curves using the proposed scaling laws (Equation 3). (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 4th repetition becomes lesser than that of worse buckets like top-20%-30% (red curve) subset at it's 4st epoch. This highlights why one needs to adapt the filtering aggressiveness with compute.

Stable optimization of scaling parameters is a crucial step in estimating the scaling laws. This is especially challenging due to the sensitive loss landscape given the complex equations. In this work, we converged at using grid search to estimate the scaling constants a, b, d and τ. We detail in Appendix E on why we made this choice and share the detailed grid used for each of the scaling parameters.

Fitting the scaling laws for individual pools: Figure 4 shows the fitted scaling curves (along with the respective parameters) for various data utility pools using T-MARS score as a data utility metric (See Appendix C for CLIP score based data pools). The central column in Figure 4 shows the diminishing utility with epochs of the various data pools. We note some key observations next.

Web data is heterogeneous and cannot be modeled by one set of scaling parameters: The heterogeneity of web

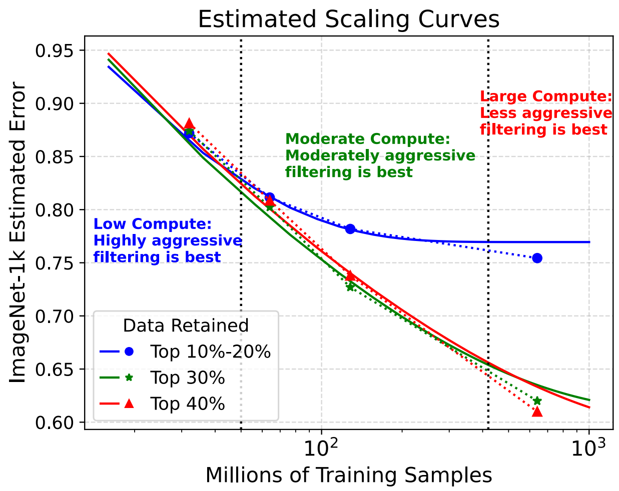

Figure 5. **Estimating scaling curve for combination of various data quality pools based on CLIP Score**: Our work proposes scaling laws for predicting the model performance on combinations of data pools of various quality. Note that we do not train on data combinations to fit the above scaling curves (scatter points are test points), rather the scaling curves are estimated from the scaling parameters of individual pools.

data is pronounced, as evidenced by the variability of the data utility parameter (b), which not only varies significantly but also demonstrates a monotonic decrease in magnitude from the highest quality pool (Top 10%) to the lower quality pool (Top 30%-Top 40%). This variation validates the use of b as a metric for data utility. Notably, the overall utility parameter for web data ($b = -0.14$), depicted by the pink curve, spans the broad spectrum between the highest ($b = -0.18$) and lowest ($b = -0.10$) utility parameters. This underscores the inadequacy of a singular scaling law framework in capturing the diverse nature of web data.

Data diversity varies across pools: Figure 4 elucidate the variation in the repetition parameter (τ) across the pools, signaling that data diversity is also not uniform. Pools of lower data quality exhibit the smallest values of half-life, indicative of lesser diversity within those pools.

Utility of high quality data with repetitions is worse than that of low quality data: High quality data, despite having a greater initial utility as depicted in the data-quality versus repetitions plot (Figure 4, center column), experiences a rapid decline in utility with successive epochs. Notably, the utility of the highest quality data pool (Top 10%) drops below that of the lowest quality pool (Top 30%-Top 40%) after the fourth epoch. This emphasizes that *data filtering must be contingent to comput*. While training for more compute, a less stringent filtering approach is advisable, as a small pool of high-quality data may underperform due to frequent repetitions, in contrast to a more sizable pool of data with modestly lower quality.

Finally, it's important to note that this observed diminishing utility is not an artifact of creating subset pools based on T-MARS scores. Similar trends can be seen even with CLIP score based data curation (Appendix C).

6. Estimating the Scaling Laws for Data Mixtures

In Section 5, we derived scaling laws to extract the scaling parameters $a, b, d,$ and τ for data pools of varying quality. The objective is to determine the most effective data curation strategy relative to the available training compute. By employing Theorem 1 alongside the scaling parameters determined for each data pool, we can *estimate* the scaling laws for different combinations of these pools. For instance, the Top-20% pool is considered a combination of the Top-10% and Top 10%-20% data quality pools. The trends from scaling curves can then allow us to predict the pareto optimal data filtering strategy at any given compute.

Figure 1 and Figure 5 present the scaling curves for different data mixtures, evaluating performance on ImageNet. Notably, these curves are derived from the estimated parameters of individual pools, **not from direct training on mixed pools**. The scatter points illustrate actual test performance, serving to validate our estimations.

Aggressive filtering is best for low compute/less repetitions regime Aggressive data filtering proves most advantageous in low compute environments when repetitions are minimal. This is exemplified by the superior performance of the highest quality data pool (Top 10% T-MARS score), as illustrated by the blue curve in Figure 1, when the model is trained for any compute of upto 100M samples seen. The low compute leads to fewer repetitions, thereby preserving the initial high utility of top-quality data. This trend holds true across both ImageNet zeroshot performance and average performance over 18 tasks.

Data curation cannot be agnostic to compute As compute scales beyond 100M samples seen, the optimal data curation strategy shifts. For example, our estimated scaling curves for Imagenet performance for various data quality mixtures indicate Top 20% as the best curation approach when training for 100M to 350M compute, rather than the more aggressive filtering of Top 10% which works the best under 100M training budgets. As the compute scales, the small but high quality subset of Top10% suffers from diminishing utility due to lot of repetitions. On further scaling up the compute beyond 350M samples, even less aggressive filtering strategy of Top 30% works better. These

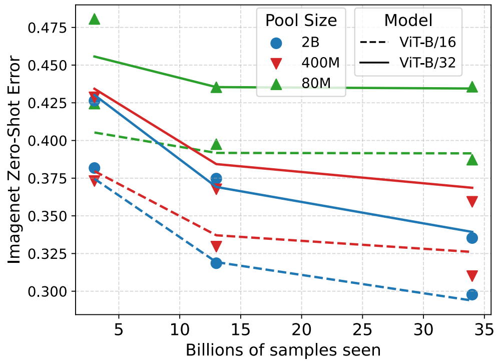

Figure 6. Similar to Figure 5, our scaling law accurately predicts the final error for models trained on 2 different architectures, 3 different pool sizes and 3 different compute budgets.

trends match the pareto optimal strategy as observed empirically as well in Figure 10, where at a compute budget of 32M, Top 10% data retained works the best while at 640M compute, Top30% works the best.

6.1. Scaling the scaling curves

Past work on scaling laws for CLIP models [9] 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 pretrained models. Cherti et al. [9] 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 6, which are able to predict the error with a high accuracy. 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.

7. Discussion

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 where one changes the training samples to pool size ratios.

Our scaling laws enable practitioners to precisely assess and quantify the utility of different web data subsets, which is critical given that webdata is heterogeneous. Finally, we show how one can estimate scaling law for a mixture of pool (Theorem 1). This enables a *compute aware* data curation, where one can decide the filtering threshold (which pools to use for training) based on the estimated accuracies using the scaling law for the mixture of pools.

State of Scaling Laws To the best of our knowledge, all scaling laws to date have modeled web data with a singular set of scaling parameters, irrespective of the specific formulation of the scaling law. As we venture into the era of large-scale foundation model training, where data curation is a critical step, our work takes significant steps towards estimating the performance of models over various possible choices of combinations of different data quality pools.

However, several questions remains open. For example, how does the data diversity τ of each pool varies as one mixes them? In this work, we considered τ to remain same as pools as mixed (upto a scaling up by a factor of increase in pool size). Similar question holds for the data quality parameter. In this work, we estimate the effective data quality b_{eff} assuming that $\frac{dy}{dn} = \frac{by}{n}$ is an axiom of scaling laws and holds true always.

8. Limitations

Effect of batch-size: Performance of visual language models trained using contrastive loss, varies considerably with the batch size employed during training. Our scaling laws, however, do not account for this variation. We perform all our experiments with a fixed batch size of 4096 on the medium scale pool of DataComp.

Consistency of scaling parameters as the pool size is scaled by orders of magnitude: While we estimate the scaling parameters of different data quality buckets on a given pool size, it is not clear whether the scaling parameters remain same for a similar quality pool of say 100x the size. Crucially, this can allow us to estimate the optimal training subset for a very large scale training by first optimizing the data pools using scaling laws on a smaller scale.

Variation in data diversity i.e. repetition parameter with mixing of pools: In our work, we operate under the assumption that the repetition parameter, influenced by data diversity, remains consistent (up to a factor proportional to the number of mixed pools). Nonetheless, the alteration in diversity across different pools, especially as we blend pools with varying levels of individual diversity, could be more complex or even challenging to predict accurately.

Acknowledgements: AR thanks Google for providing GCP credits that supported part of this work.

References

[1] Common crawl. https://commoncrawl.org/. 2

[2] Ibrahim Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision, 2022. 3

[3] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws, 2021. 3

[4] 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. 11

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 11

[6] Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws, 2023. 3

[7] 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. 11

[8] 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, 2017. 11

[9] 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. 3, 5, 8, 17

[10] 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. 11

[11] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *International Conference on Machine Learning*, pages 6216–6234. PMLR, 2022. 2

[12] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks, 2023. 3

[13] 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. 11

[14] 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, 2023. 2, 3, 5, 11

[15] 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, 2023. 3

[16] Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Jenia Jitsev, Alexandros G. Dimakis, Gabriel Ilharco, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks, 2024. 3

[17] Tatsunori Hashimoto. Model performance scaling with multiple data sources. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4107–4116. PMLR, 2021. 3

[18] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 11

[19] 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. 11

[20] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer, 2021. 3

[21] 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. 3, 13, 17

[22] Marcus Hutter. Learning curve theory, 2021. 3

[23] 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, 2021. If you use this software, please cite it as below. 2

[24] Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and Sanmi Koyejo. Scaling laws for downstream task performance of large language models, 2024. 3

[25] 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. 5

[26] 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. 3, 5, 13, 17

[27] 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. 11

[28] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 11

[29] 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. 2

[30] Anas Mahmoud, Mostafa Elhoushi, Amro Abbas, Yu Yang, Newsha Ardalani, Hugh Leather, and Ari Morcos. Sieve: Multimodal dataset pruning using image captioning models, 2023. 2

[31] 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. 2, 3, 6

[32] 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. 3, 6, 14, 15, 16, 17

[33] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning, 2023. 2

[34] 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*, 2008. 11

[35] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 11

[36] 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. 2

[37] 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)*, pages 8748–8763, 2021. 11

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[39] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019. 11

[40] 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. 11

[41] Nikhil Sardana and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws, 2023. 3

[42] 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. 2, 4

[43] 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. 2, 3, 4

[44] 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. 11

[45] 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. 11

[46] 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. 3

[47] 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. 11

[48] 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. 3

[49] 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. 11