



How to Predict Best Pretraining Data with Small Experiments

Ian Magnusson^{*1,2} Nguyen Tai^{*3} Ben Bogin^{*1} David Heineman¹ Jena Hwang¹ Luca Soldaini¹ Akshita Bhagia¹
Jiacheng Liu^{1,2} Dirk Groeneveld¹ Oyvind Tafjord¹ Noah A. Smith^{1,2} Pang Wei Koh^{1,2} Jesse Dodge¹

Abstract

Because large language models are expensive to pretrain on different datasets, using smaller-scale experiments to decide on data is crucial for reducing costs. Which benchmarks and methods of making decisions from observed performance at small scale most accurately predict the datasets that yield the best large models? To empower open exploration of this question, we release models, data, and evaluations in DATADECIDE—the most extensive open suite of models over differences in data and scale. We conduct controlled pretraining experiments across 25 corpora with differing sources, deduplication, and filtering up to 100B tokens, model sizes up to 1B parameters, and 3 random seeds. We find that the ranking of models at a single, small size (e.g., 150M parameters) is a strong baseline for predicting best models at our larger target scale (1B) (~80% of comparisons correct). No scaling law methods among 8 baselines exceed the compute-decision frontier of single-scale predictions, but DATADECIDE can measure improvement in future scaling laws. We also identify that using continuous likelihood metrics as proxies in small experiments makes benchmarks including MMLU, ARC, HellaSwag, MBPP, and HumanEval > 80% predictable at the target 1B scale with just 0.01% of the compute.

1. Introduction

The cost of training large language models (LMs) necessitates methods of trying out options at small scale, but it also makes it expensive to validate the accuracy of development decisions made with such methods. We focus on the question of choosing between pretraining datasets to use—one of the most impactful development decisions. Common practice (e.g., Li et al., 2024) uses a single, small scale of experiments to cheaply test pretraining data intended for larger-scale models, where scale is determined by number of model parameters and training tokens. The other predominant approach is to fit scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022; Choshen et al., 2024) to the trend in performance observed over multiple small scales, with recent work extending this to the prediction of downstream performance instead of language modeling loss (Gadre et al., 2024; Dubey et al., 2024; Bhagia et al., 2024).

So far decision-making approaches have only been validated without observing the counterfactual outcome, either by producing a single large model on the chosen decision with impressive performance or by low error in predicting the magnitude of observed performance of a small number of large models. Knowing what amount of error in predicting performance over scale is a low enough to actually make a correct decision among datasets, requires a suite of comparable models trained on many datasets. Although a wide variety of open-source pretraining corpora are available, the scaling behavior of data is difficult to assess from off-the-shelf models that vary simultaneously in data, optimizer, and modeling decisions.

To make it possible to empirically study what methods make the best decisions over data, we build DATADECIDE¹—a suite of models we pretrain on 25 corpora up to 100B tokens, over 14 different model sizes ranging from 4M parameters up to 1B parameters (more than 30K model checkpoints in total). We evaluate all models across a suite of 10 downstream tasks and calculate how accurately small models predict which pretraining corpora lead to better performance

^{*}Equal contribution ¹Allen Institute for AI ²Paul G. Allen School of Computer Science & Engineering, University of Washington ³University of Pennsylvania. Correspondence to: Ian Magnusson <ianmag@cs.washington.edu>.

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

¹DataDecide collection on HuggingFace

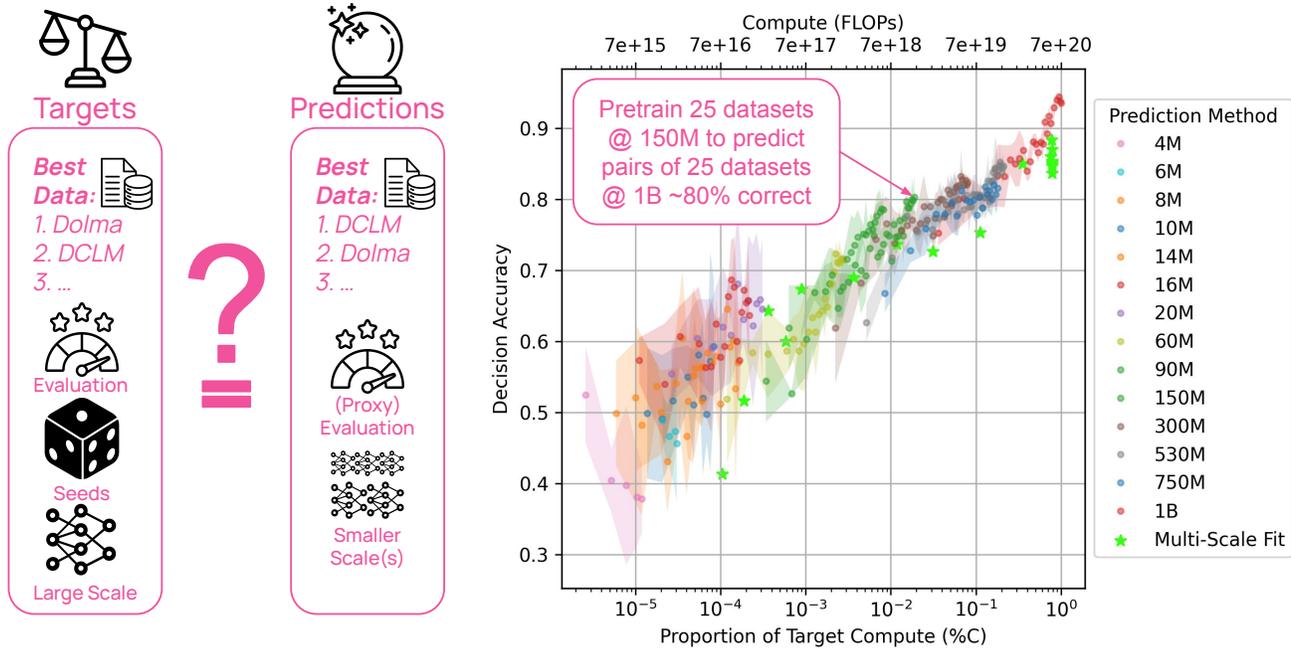


Figure 1. Which pretraining data to use? Ideally, compare performance of large models with fixed configurations averaged over random seeds (left). In practice, cheaper, smaller-scale experiments are used (center). Here DATADECIDE measures accuracy of pairwise decisions between 25 pretraining corpora to find efficient prediction methods (right).

at our largest scale. Our conclusions provide practical recommendations for the best benchmarks, prediction methods, and metrics to use to make decisions.

We call the 25 corpora we train on *data recipes* as they range across popular corpora including Dolma (Soldaini et al., 2024), DCLM (Li et al., 2024), RefinedWeb (Penedo et al., 2023), C4 (Raffel et al., 2019), and FineWeb (Penedo et al., 2024) as well as combinations of interventions on these datasets such as source mixing, deduplication, and filtering. Previous work has considered only 2 (Biderman et al., 2023) or 6 recipes (Magnusson et al., 2024; Brandfonbrener et al., 2024). We also offer a novel affordance by including 3 random seed reruns for even our largest runs, to help quantify whether variation occurs due to random initialization and data order or differences in the distribution of data.

Concretely, DATADECIDE allows analyses such as Figure 1 (right), which shows the relationship between compute used to predict a ranking of datasets and how accurately that ranking reflects mean performance over 3 seed runs (quantified here by OLMES; Gu et al., 2024) for models fully trained on those datasets at the target (1B) scale. We measure the accuracy of decisions as the percent of compared pairs of datasets where the prediction identifies the correct winner. Each point represents the average decision accuracy of a given method over 3 prediction attempts using small models with different random seeds, and shading shows standard

deviation.

Measuring the tradeoff of compute cost to better decisions lets us make the following recommendations about small experiments for making data decisions:

- §3.1 – The amount of compute you need to allocate for a given decision accuracy depends heavily on task. MMLU and ARC are much cheaper to predict than HelLaSwag and some tasks such as SocialIQA are difficult to predict at all scales.
- §3.2 – 8 baseline scaling law methods do not exceed the compute to decision accuracy frontier set by ranking single scale experiments.
- §3.3 – At small scales, continuous metrics using answer likelihood are better or equivalent predictors of decisions than using the same discrete accuracy target metric.
- §3.4 – Better decisions can be explained in part by low run-to-run variance and a wide spread of benchmark performance values for different data, traits which can be improved by proxy metrics.

Future research can extend DATADECIDE with little extra compute by running new evaluations on our checkpoints, pretraining additional small models to compare against the large target models we provide, or trying new prediction

Source / Recipe	Description
Dolma1.7 <i>Original, No code, No math/code, No Reddit, No Flan</i>	A 2.3T-token corpus (Dolma 1.7 Soldaini et al., 2024) sampling common LM sources for open research. We ablate code, math/code, Reddit, or Flan subsets.
Dolma1.6++ <i>Original</i>	Dolma 1.6 plus additional sources from Dolma 1.7: RedPajama’s arxiv subset, openwebmath, algebraic stack, flan, starcoder, falcon.
C4 <i>Original</i>	The C4 dataset (Raffel et al., 2019) as prepared in Dolma 1.7, heuristically filtered from the April 2019 Common Crawl.
FineWeb-Pro <i>Original</i>	The FineWeb Pro corpus (Zhou et al., 2024), featuring model-driven data cleaning on FineWeb.
FineWeb-Edu <i>Original</i>	The deduplicated FineWeb-Edu subset of SmolLM-Corpus (Ben Allal et al., 2024), focused on educational web pages.
Falcon <i>Original</i>	The Falcon RefinedWeb corpus (Penedo et al., 2023) in Dolma 1.7, derived from Common Crawl through June 2023 and more aggressively filtered/deduplicated than C4.
Falcon+CC <i>Original, QC 10%, QC 20%, QC Orig 10%, QC Tulu 10%</i>	Falcon and Dolma 1.7’s Common Crawl. We quality filter to top 10% or 20% documents with reproduced or original (Li et al., 2024) filter or retrain filter on pre-release version of Tulu-v3 (Lambert et al., 2024).
DCLM-Baseline <i>Original, QC 7% FW2, QC 7% FW3, QC FW 3%, QC FW 10%, QC 10%, QC 20%</i>	A SOTA Common Crawl corpus using best ablated deduplication, cleaning heuristics, and quality filter. We quality filter to top 7% of DCLM classified documents and further take 2+ or 3+ scores with FineWeb-edu classifier; or filter to top 3% or 10% with FineWeb-edu classifier; or take top 10% or 20% with reproduced DCLM classifier.
$\lambda\%$ DCLM-Baseline + $1 - \lambda\%$ Dolma1.7	Fractional combinations of Dolma1.7 and DCLM-Baseline mixing different proportions of the two datasets for $\lambda \in \{25\%, 50\%, 75\%\}$.

Table 1. DATADECIDE enables the study of data differences over scales through controlled pretraining experiments on 25 data recipes. These take different source datasets and apply interventions from ablating domains, deduplication, mixing, to quality filtering with different classifiers and thresholds. We release all pretraining corpora, as well as models trained on each recipe and each of the 14 model configurations in Table 2 with 3 random seeds.

methods with lightweight manipulations such as smoothing and curve fitting on top of our released evaluation results.

2. Methods

Our aim is to empirically test the predictability of downstream performance at a larger, target scale using small experiments. We describe DATADECIDE §2.1, the prediction methods we examine §2.2, the metrics we use to assess predictions §2.3, how we measure downstream performance §2.4, and proxy metrics for our performance evaluations §2.5. We will release all models, checkpoints, pretraining corpora, and evaluations.

2.1. The DATADECIDE Suite

We pretrain a suite of 1,050 models using 25 data recipes \times 14 model scales \times 3 random seeds for initialization and data order. Table 1 describes the 25 data recipes included in DATADECIDE that aim to provide coverage of common data preparation choices such as deduplication, ablating domains, mixes of existing datasets, as well as quality filters with different implementations, training data, and thresholds for quality classifiers.

We select a token to parameter ratio of 100, which at $5 \times$

“Chinchilla” ($5 \times C$) optimal ratio (Hoffmann et al., 2022) captures the typical overtraining favored for inference savings.

All 1B (target size) models have 3 full reruns with different seeds, while other model sizes have second and third seed runs that are terminated early after 25% of the target compute budget. We train the 1B reruns all the way to completion to allow our target “gold” predictions to account for run-to-run variance in evaluations due to weight initialization and data order. For instance, we find that the standard deviation between runs at the 1B $5 \times C$ scale can be as high as 2% points of accuracy for some recipes on most tasks. Meanwhile, at the non-target scales we wish to make predictions with a small fraction of the target compute, so we avoid reruns that would use an impractically large prediction budget.

Whether for extrapolating scaling laws or ranking single scale experiments, it is important to select reasonable hyperparameters for each scale to avoid confounding in performance differences that are simply due to suboptimal hyperparameters. We use OLMo’s *model ladder* (Groeneveld et al., 2024; OLMo et al., 2025; Bhagia et al., 2024) to programmatically create LM pretraining configurations for a specified parameter size and token-parameter ratio to enable

running a grid of model scaling experiments. The model ladder uses heuristics from the literature (Porian et al., 2024) to set global batch size and learning rate based on scaling factors. The hyperparameters that determine parameter count (layers, hidden dimension, number of heads, MLP dimension) were handpicked by OLMo developers for each scale to achieve the desired number of parameters. Appendix Table 2 details the configurations of all our models.

2.2. Prediction Methods

Broadly, there are two approaches in the literature to predicting large-scale performance based on small-scale experiments. We use straightforward implementations of each to assess where they succeed and fail at making decisions about which data recipes to use.

Ranking Single Scale Experiments (Single Scale) This simple approach is employed by work such as Li et al. (2024) and consists of running a set of ablations or experiments over data recipe options while holding constant all other modeling variables including scale. The winning data recipe by downstream accuracy (or proxies) at the small experimental scale is assumed to extrapolate to the target scale.

Extrapolating Scaling Laws (Multi Scale) Another approach to making decisions with predictions across scales used in works such as Dubey et al. (2024) is to fit scaling laws to *multiple* small experiments across a range of scales for each of the data recipes. The winning recipe is decided as the one whose scaling law shows the highest *extrapolated* performance at the target scale. Although scaling laws were first observed for language modeling loss (Kaplan et al., 2020; Hoffmann et al., 2022), they have been extended to predict downstream performance through a two-step approach that also fits a function from loss to downstream performance (Gadre et al., 2024; Bhagia et al., 2024). We follow a method from Bhagia et al. (2024). Their proposed approach incorporates separate parameters for number of model parameters and number of tokens trained to account for over or undertrained models. But as our suite only includes one token-parameter ratio, we use the simplified 3 parameter baseline, $L(C)$, as a first step which we chain with second step, $Acc(L)$, defined as follows where $A, \alpha, E, a, b, k, L_0$ are optimized parameters:

$$L(C) = \frac{A}{C^\alpha} + E \quad (1)$$

$$Acc(L) = \frac{a}{1 + e^{-k(L-L_0)}} + b \quad (2)$$

Following Bhagia et al. (2024) we fit Equation 1 only on observations of final, fully trained checkpoints as accounting for the learning rate schedule’s impact on intermediate checkpoints would require further parameters in the equation increasing the required number of observations and

cost. To account for step-to-step noise in evaluation we average the last 10% of checkpoints as the final observed loss. Equation 2, however, is fit on all observations including intermediate checkpoints. We explore variations for a total of 8 multi scale approaches defined in Appendix C; none of these make for substantially better decisions than the method defined in this section.

2.3. Prediction Metrics

Our predictive task is to forecast which of a pair of data recipes will perform better at some target scale based on small-scale experiments. We use the following metrics to measure the quality of these predictions.

Prediction Error Scaling laws literature (Bhagia et al., 2024; Gadre et al., 2024) typically evaluates success from predicted and actual downstream performance, using relative error ($\frac{|\text{predicted}-\text{actual}|}{\text{actual}} \times 100\%$) or absolute error ($|\text{predicted} - \text{actual}| \times 100\%$). We call these absolute or relative “prediction error” to distinguish from the following metric.

Decision Accuracy Unlike previous work, we also measure the impact of predictions on *decisions* about which data recipe is better than another. The metric we use to capture this is decision accuracy, an accuracy over all pairs of data recipes A and B where either A or B is defined as the correct winner based on which achieves higher performance at the target scale. This is nearly equivalent to Kendall’s τ , but ranges from 0 to 1. We define the target-scale winner based on mean downstream performance over 3 random seeds. Thus decision accuracy can be formalized as follows. Let \mathcal{P} be the set of all data recipe pairs (A, B) with observed mean performance y_A, y_B and predicted performance \hat{y}_A, \hat{y}_B , respectively, then decision accuracy is:

$$\frac{1}{|\mathcal{P}|} \sum_{(A,B) \in \mathcal{P}} \mathbb{I}(\text{sign}(\hat{y}_A - \hat{y}_B) = \text{sign}(y_A - y_B)) \quad (3)$$

Percent of Target Compute Budget (%C) We measure compute in terms of theoretical FLOPs following the simplifying assumption made in most scaling literature that the costs associated with training a model are captured well enough by FLOPs = $6ND$, based solely on the number of parameters (N) and tokens trained (D) (Kaplan et al., 2020). We consider the efficiency of a prediction based on the ratio of the experimental budget and the target budget in FLOPs, $\%C = \frac{c}{C} \times 100\%$.

2.4. Performance Evaluation with OLMES

We use the OLMES suite of 10 multiple choice question answering benchmarks (Gu et al., 2024): MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), ARC Challenge (Clark et al., 2018), ARC Easy (Clark et al., 2018),

PIQA (Bisk et al., 2020), CommonsenseQA (Talmor et al., 2019), SocialIQA (Sap et al., 2019), OpenBookQA (Mihaylov et al., 2018), BoolQ (Clark et al., 2019), and WinoGrande (Sakaguchi et al., 2020). These tasks are well suited for the model scales we examine with all but BoolQ receiving non-trivial performance. Unless otherwise noted, we consider the macro average of these ten tasks. The underlying metric for each task is accuracy, for which OLMES specifies a different length normalization scheme per task. Our target “gold” rankings which we aim to predict are always based on the “cloze” formulation (CF) accuracy with curated normalization per task, which we refer to as ACCURACY. We diverge from OLMES only in that we make use of all available items in the specified split of each benchmark rather than subsampling them, to reduce variance over the task distribution.

Note that while we focus just on OLMES multiple choice evaluations in this work, our method of validating decisions made through predictions can be applied to other benchmarks. We chose these tasks based on their appropriateness to our range of model scales, and one would have to select different tasks when targeting a larger scale. Moreover, DATADECIDE could be used to identify new evaluations that are sensitive within our range of scales.

2.5. Proxy Metrics for Performance Evaluation

Previous work has noted how discrete metrics such as accuracy can cause jumps in performance across scale that otherwise see more predictable improvements with scale for continuous metrics (Schaeffer et al., 2023). We experiment with using continuous metrics at small scale as proxies of the accuracies selected by OLMES for each task (ACCURACY) at the target scale to improve decision accuracy. We use the following metrics: CORRECT PROB is the average probabilities of the correct continuations. MARGIN is the average difference between the probability of the correct continuation and the most likely incorrect continuation. NORM CORRECT PROB is the average probability of the correct continuation conditioned on the response being in the set of correct or incorrect continuations. TOTAL PROB is the average of the sum of probabilities of all correct and incorrect continuations. ACCURACY is the fraction of instances where the correct continuation has the highest probability. Each of these can be computed with likelihoods normalized by number of tokens or characters; unless otherwise specified we use character length normalization. Appendix Table 3 shows formal definitions.

3. Results

3.1. What is the best way to spend compute for data decisions?

More compute makes better decisions. Decisions from intermediate checkpoints are as good as compute equivalent final checkpoints. The amount of compute needed to make good predictions varies between tasks. ARC and MMLU are predictable with much less compute than HellaSwag. The rest of OLMES tasks give markedly less reliable predictions across the scales we examine.

First looking at the aggregation of all 10 OLMES tasks (Figure 1 right), we see that there is a positive and roughly log-linear relationship between experimental compute and decision accuracy. Specifically, this figure illustrates the relationship between the compute used for predicting best data recipes and the decision accuracy those predictions achieve against targets ranked by OLMES performance at the 1B scale. Each point represents the average decision accuracy over three runs with different random seeds, with shading indicating standard deviation. Points with the same color show all intermediate checkpoints from a given parameter size. The color shows each model size for predicting using ranking single scale experiments. The stars show predictions from extrapolating scaling laws using our default 3-parameter approach, the details of which are discussed further in §3.2.

The ease of prediction is greatly influenced by which evaluation benchmark we use. In Figure 2, we show the relationship of compute and decision accuracy for each of the tasks in OLMES individually. The predictive sensitivity of tasks at a given compute varies significantly, with ARC Easy being consistently predictable with 5 orders of magnitude less compute and BoolQ only reaching beyond trivial decision accuracy for intermediate checkpoints of the target runs. HellaSwag, SocialIQA, WinoGrande show distinct periods of insensitivity followed by roughly log-linear increase after hitting some compute threshold.

3.2. How does extrapolating scaling laws compare to ranking single scale experiments?

A selection of 8 baseline scaling law methods are no more efficient than ranking single scale experiments. Future scaling law methods can be assessed on DATADECIDE.

Figure 3 contrasts different approaches to fitting scaling laws over multiple scales of small experiments. Each of the 8 approaches is shown in a different color. Multi-scale predictions have a compute budget equal to the training

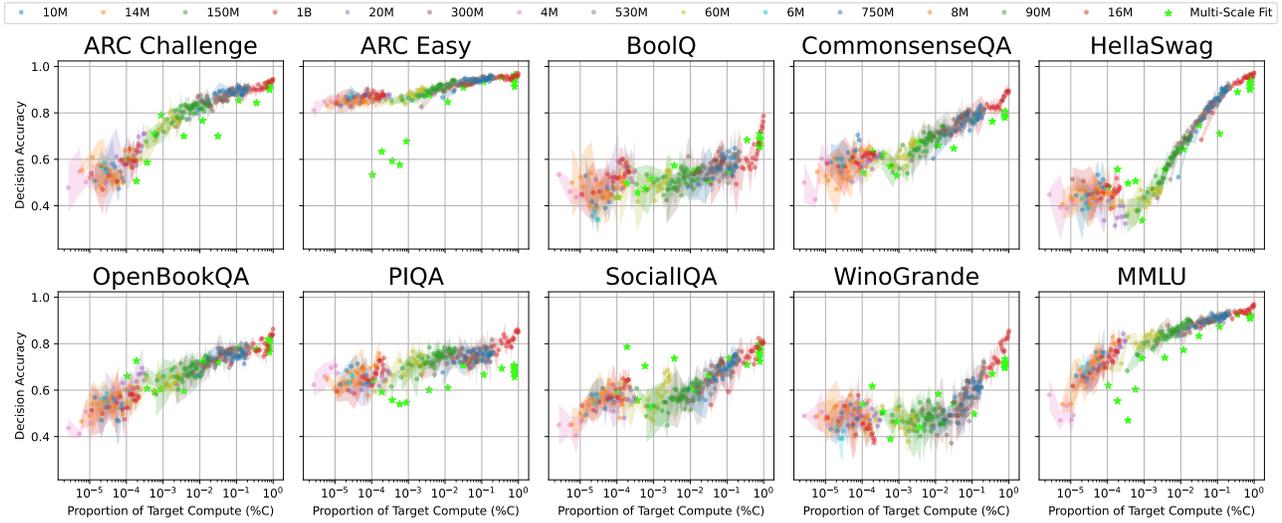


Figure 2. Accuracy in pairwise decisions on best data when evaluating on the 10 OLMES tasks with ACCURACY (shown aggregated in Figure 1). Specific tasks have very distinct ranges of sensitivity, with some like ARC Easy being predictable at small scales and others like HellaSwag requiring substantially more compute to predict.

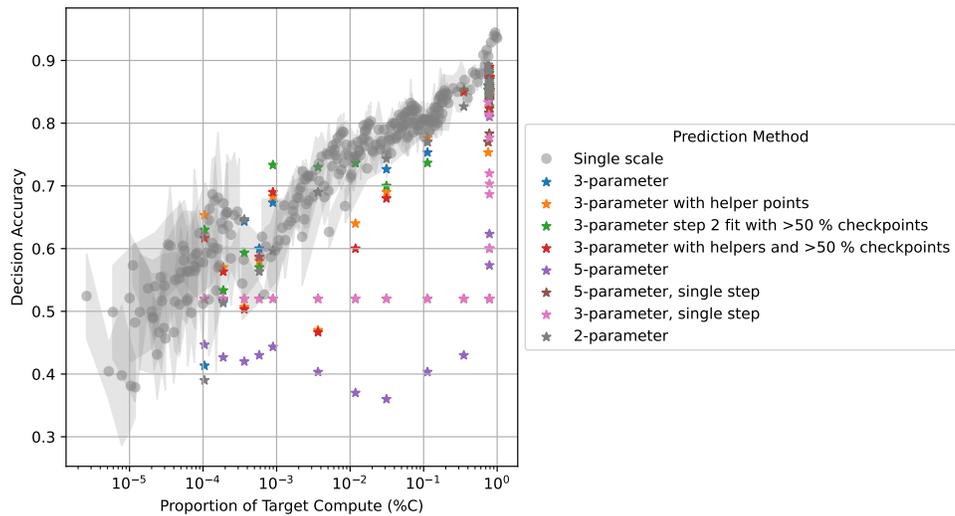


Figure 3. Decision accuracy over 8 baseline scaling law variants. At best, these approaches reach only the same compute to decision accuracy frontier as ranking single scale experiments. DATADECIDE can be used to iterate on future scaling law prediction methods.

cost of the model sizes used to make the prediction. We try the following combinations of models sizes: We use $\{\{s_1, \dots, s_k\} \mid 3 \leq k \leq 14\}$, where s is the ordered set of sizes, to explore the improvements of progressively adding larger model sizes beyond the minimum 3 required for fitting. We also use $\{\{s_k, \dots, s_{14}\} \mid 2 \leq k \leq 11\}$ to try removing potentially noisy information from small models. Unlike single scale results, we make only one prediction attempt with the default fully trained random seed, as final checkpoints are required for fitting the first step of these scaling law variants but are not available for all seeds.

Our scaling law approaches vary in the number of parameters fit, using hard coded points to define the minimum and maximum performance, using only the second half of intermediate checkpoints for fitting the second step, or fitting a function directly from compute to accuracy in a single step. Each of the scaling law variants are defined formally in Appendix C. The 2 and 3 parameter variants all achieve among the top decision accuracy.

A priori we know that ranking single scale experiments cannot correctly predict when the scaling trend of one data recipe overtakes another at scales between our small experiments and target scale. Such crossovers bound the decision accuracy of this constant approximation of performance. Nevertheless ranking single scale experiments sets a high baseline decision accuracy, implying relatively little crossover occurs. It is difficult to distinguish evaluation variance from true crossovers, but the scaling trends we empirically observe cross over frequently. Improved future scaling laws may be able to advance the Pareto frontier on DATADECIDE as they are not bound by crossovers.

3.3. What proxy metrics give better signal for predictions at small scale?

At small scales, continuous metrics using the character normalized likelihood of correct or all answer options serve as better or equivalent predictors of decisions than using the same ACCURACY as used at the target scale.

Figure 4 shows the decision accuracy over different proxy metrics. Here we chose a single length normalization, *_PER_CHAR. Metrics follow similar trends regardless of length normalization and this one is empirically optimal for most of the tasks that we observe.

Using CORRECT PROB or TOTAL PROB leads to decision accuracy at least as good as any other metric for most small scales. These continuous metrics are simple likelihoods over answer strings. In particular, TOTAL PROB may be interpretable as signal of a model having exposure to the domain of a given task in the form of higher likelihoods on

incorrect but presumably relevant additional answers.

We notice two very distinct types of trends over the different tasks. Either the different proxy metrics are nearly indistinguishable and increase in decision accuracy with compute or CORRECT PROB and TOTAL PROB are flat with respect to scale and the other metrics only rise up to that level of decision accuracy towards the full target compute budget. In the last order of magnitude below the target compute ACCURACY and the other metrics tend to overtake CORRECT PROB and TOTAL PROB, while these two metrics sometimes even decrease in decision accuracy. Notably these other metrics that trend with ACCURACY include continuous metrics that penalize probability assigned to incorrect answers, NORM CORRECT PROB and MARGIN.

3.4. How can we make evaluation benchmarks more predictable?

The decision accuracy on a task is driven in part by low run-to-run variance and a wide spread of performance values for different data recipes. Using CORRECT PROB sees wider spreads or reduced noise for many tasks. Using this metric enables predicting rankings for code tasks that are too hard for accuracy metrics at small scales.

What underlies differences in decision accuracy when benchmarks and metrics change? The evaluation must separate pairs of data recipes by an amount greater than combined noise from run-to-run variance of each of the pair’s runs. In Figure 5, we plot tasks with a given metric using fully trained 150M models over these two characteristics: 1) noise—the standard deviation over 3 random seed runs averaged over all recipes, and 2) spread—the standard deviation among the mean performance of the different data recipes. Each point also shows the decision accuracy. We see that some highly predictable tasks (e.g., MMLU) are characterized by having low run-to-run noise, while others (e.g., ARC Easy) widely spread the different data recipes. We also see that improvements from using CORRECT PROB often align with improvements in one of these two characteristics.

As a practical application of these insights, we demonstrate that a change of proxy metric makes predictable two code tasks (Austin et al., 2021; Chen et al., 2021) that are otherwise too challenging for our small models. Figure 6 shows how decision accuracy goes from trivial to 80% when using CORRECT PROB. The switch of metric allows small models to get above the noise floor for these tasks, while still predicting large-scale accuracy metrics. Notably, two math benchmarks (Lewkowycz et al., 2022; Cobbe et al., 2021) do not see such a benefit. They do however give decision accuracy above 80% if we switch the *target metric* to CORRECT PROB, raising a question for future work to

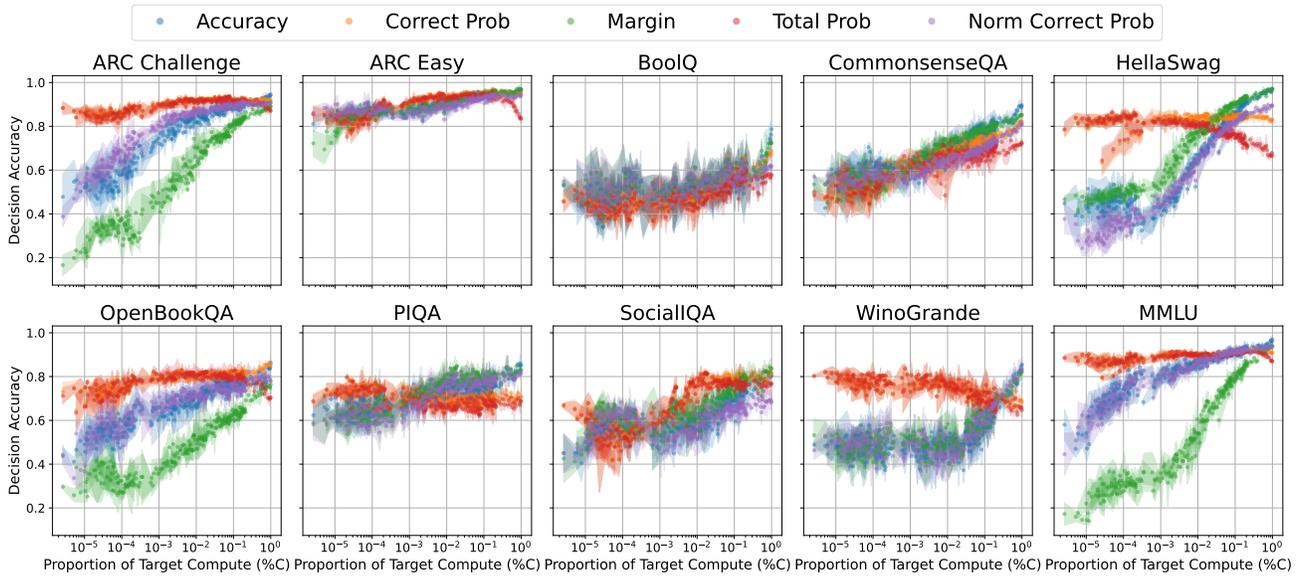


Figure 4. Per-task decision accuracy using character normalized proxy metrics for ACCURACY targets. 5 tasks benefit at smaller scales from using raw likelihood of answers (CORRECT PROB and TOTAL PROB), as opposed to discrete ACCURACY or continuous metrics that penalize probability on incorrect answers (NORM CORRECT PROB, MARGIN).

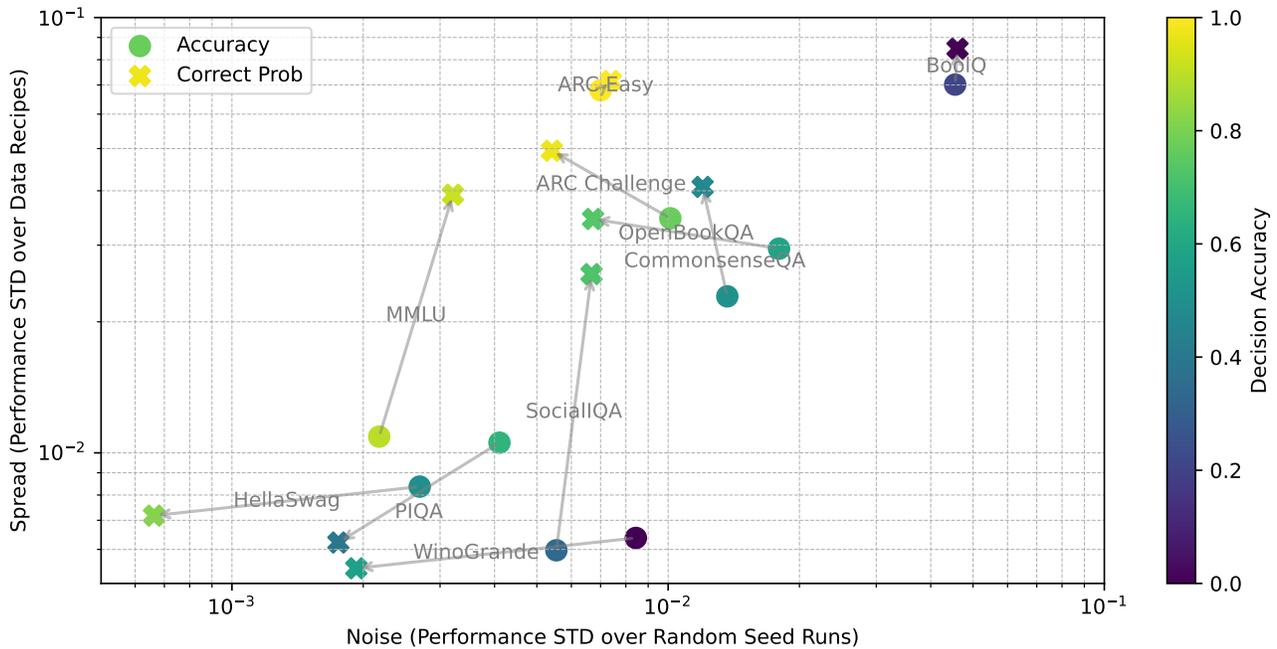


Figure 5. Why do some tasks or metrics get better or worse decision accuracy? At 150M with CORRECT PROB tasks like HellaSwag succeed with low run-to-run variance and tasks like SocialQA widely spread the performance assigned to different pretraining data.

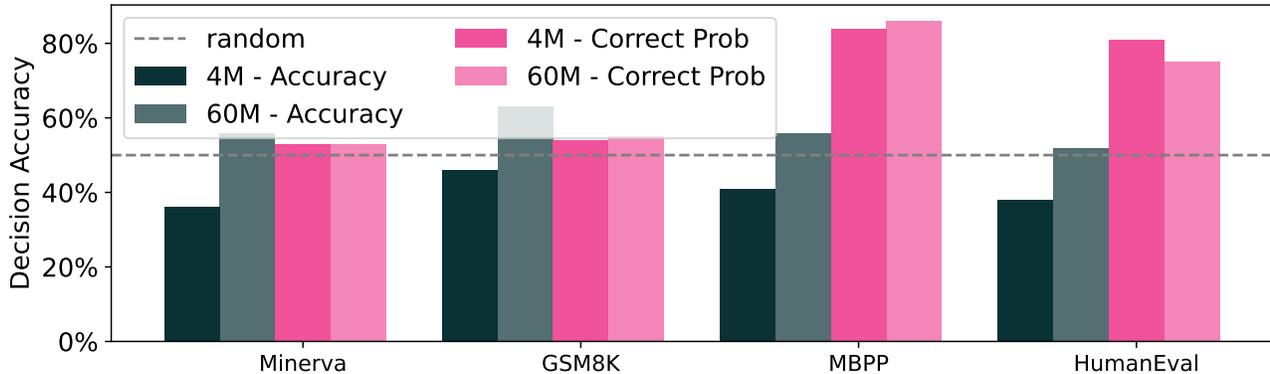


Figure 6. Code tasks such as humaneval and MBPP go from trivial decision accuracy to largely predictable when using using continuous CORRECT PROB instead of discrete ACCURACY. Meanwhile common math tasks remain near trivial decision accuracy regardless of metric.

explore whether changing the target metric can be justified.

4. Related Work

Prediction Much work studies scaling behavior in language models. Initially this focused on predicting LM loss from scale as determined by parameter count and tokens trained (Kaplan et al., 2020; Hoffmann et al., 2022). Special consideration is also given to the case of data constrained scaling (Muennighoff et al., 2023; Goyal et al., 2024). Unlike predicting loss, predicting downstream performance from scale is generally harder (Schaeffer et al., 2024). However, recent work has demonstrated it can be done based on a two step prediction that chains together predictions from scale to loss and loss to downstream performance (Gadre et al., 2024; Bhagia et al., 2024; Dubey et al., 2024), sometimes using training loss (Du et al., 2024) or transferring losses from different data recipes (Brandfonbrener et al., 2024; Ruan et al., 2024). The one line of work targeting pretraining data considers the special case of deciding mixing proportions of several data sources optimized through scaling laws (Kang et al., 2024; Ye et al., 2024). Most relevant to our work, Choshen et al. (2024) consider practical methods for better scaling prediction error such as how much compute to use or whether to include intermediate checkpoints. Orthogonally to these findings, we propose a way to assess the accuracy of decisions made with such predictions.

Suites over Data Differences DATADECIDE follows in the footsteps of the Pythia Suite (Biderman et al., 2023) which was the first to offer a controlled comparison of 2 data recipes, using compute scales up to 2×10^{22} FLOPs. Subsequent suites have offered 6 data recipes at 9×10^{20} scale (Magnusson et al., 2024) and 6 data recipes over a

range of scales up to 10^{21} (Brandfonbrener et al., 2024). Our DATADECIDE offers a range of 14 scales up to 7×10^{20} FLOPs, while including an order of magnitude more fine-grained data differences. Meanwhile, DCLM also makes extensive use of ranking single scale experiments to drive improvement in data recipes (Li et al., 2024). They release their best data and a model trained on it, but do not release models from their decision making experiments and do not search over multiple recipes at their largest scale. Where their goal is creating a proposed best recipe, our DATADECIDE enables the assessment of whether a method for decision making really does find the best among proposed recipes.

5. Limitations

The scope of our work is limited to just one ratio of tokens to parameters, 100 or $5 \times$ “Chinchilla” optimal ratio (Hoffmann et al., 2022). We believe this captures the typical case, as most models now favor overtraining for inference savings. Due to compute limitations and the need for a standardized set of model configurations over a long period of time in which compute became available for pretraining, we opt for 14 specific configurations from 4M–1B parameter scale. While observations across more configurations would always be better, this must be traded off with exploring the other dimensions of data recipes and random seed reruns. Likewise, while our 25 data recipes is an order of magnitude more than previous suites, there is always the possibility that findings across these will not be representative of future data recipes. In our evaluations we focus on multiple choice tasks with a “cloze” formulation as we find these to be a good fit for our range of scales. Using DATADECIDE, new evaluations can be assessed easily by others without any additional pretraining.

Acknowledgments

We would like to thank Dave Wadden, Kyle Lo, Valentin Hofmann, and Hannaneh Hajishirzi for fruitful conversations. This material is based upon work supported by the U.S. National Science Foundation under Grant No. 2313998. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. National Science Foundation. IM is supported by the NSF CSGrad4US Fellowship. PWK is supported by the Singapore National Research Foundation and the National AI Group in the Singapore Ministry of Digital Development and Information under the AI Visiting Professorship Programme (award number AIVP-2024-001) and by the AI2050 program at Schmidt Sciences.

Impact Statement

Training large language models is computationally expensive, especially when investigating thoroughly over dimensions of pretraining data composition, model scale, random initialization, and data order. The pretraining experiments in our DATADECIDE required approximately 820K H100 GPU hours. We share the benefit of this cost through releasing all of our models, data, and evaluations so that others will not have to repeat this expenditure. Moreover, our findings can guide efficient and cost-effective model development through the application of decision making with small-scale experiments. While DATADECIDE does not present direct ethical concerns beyond opportunity cost, we acknowledge that decisions about pretraining data heavily impact downstream model behavior. We encourage future research to explore potential biases in data selection methods and their implications for models deployed in the real world.

References

- Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.
- Ben Allal, L., Lozhkov, A., Penedo, G., Wolf, T., and von Werra, L. Smollm-corpus, July 2024. URL <https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus>.
- Bhagia, A., Liu, J., Wettig, A., Heineman, D., Tafjord, O., Jha, A. H., Soldaini, L., Smith, N. A., Groeneveld, D., Koh, P. W., Dodge, J., and Hajishirzi, H. Establishing task scaling laws via compute-efficient model ladders, 2024. URL <https://arxiv.org/abs/2412.04403>.
- Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O’Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and van der Wal, O. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL <https://arxiv.org/abs/2304.01373>.
- Bisk, Y., Zellers, R., Le bras, R., Gao, J., and Choi, Y. PIQA: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020. doi: 10.1609/aaai.v34i05.6239. URL <https://ojs.aaai.org/index.php/AAAI/article/view/6239>.
- Brandfonbrener, D., Anand, N., Vyas, N., Malach, E., and Kakade, S. Loss-to-loss prediction: Scaling laws for all datasets, 2024. URL <https://arxiv.org/abs/2411.12925>.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021. URL <https://arxiv.org/abs/2107.03374>.
- Choshen, L., Zhang, Y., and Andreas, J. A hitchhiker’s guide to scaling law estimation, 2024. URL <https://arxiv.org/abs/2410.11840>.
- Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T. (eds.), *NAACL*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. doi: 10.18653/v1/N19-1300.
- Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, 2018. URL <http://arxiv.org/abs/1803.05457>.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

- Common Crawl. Common crawl. URL <https://commoncrawl.org>. Accessed: 2025-05-21.
- Du, Z., Zeng, A., Dong, Y., and Tang, J. Understanding emergent abilities of language models from the loss perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024*. URL <https://openreview.net/forum?id=35DAviqMFO>.
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A. S., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Rozière, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E. A., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I. M., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J. L., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J.-Q., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., neth Heafield, K.-., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M. B., Singh, M., Paluri, M., Kardas, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M. H. M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N. S., Duchenne, O., cCelebi, O., Alrassy, P., Zhang, P., Li, P., Vasić, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S. C., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., ney Meers, W., Martinet, X., Wang, X., Tan, X. E., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A. K., Grattafiori, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Vaughan, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Franco, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, P.-Y. B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Civin, D., Beaty, D., Kreymer, D., Li, S.-W., Wyatt, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Ozgenel, F., Caggioni, F., Guzm'an, F., Kanayet, F. J., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Thattai, G., Herman, G., Sizov, G. G., Zhang, G., Lakshminarayanan, G., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Molybog, I., Tufanov, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., KamHou, U., Saxena, K., Prasad, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Huang, K., Chawla, K., Lakhotia, K., Huang, K., Chen, L., Garg, L., Lavender, A., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Tsimpoukelli, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Laptev, N. P., Dong, N., Zhang, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollár, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Li, R., Hogan, R., Battey, R., Wang, R., Maheswari, R., Howes, R., Rinott, R., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Shankar, S., Zhang, S., Wang, S., Agarwal,

- S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Cho, S.-B., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Kohler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V. A., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wang, X., Wu, X., Wang, X., Xia, X., Wu, X., Gao, X., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Wang, Y., Hao, Y., Qian, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., and Zhao, Z. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. URL <https://api.semanticscholar.org/CorpusID:271571434>.
- Gadre, S. Y., Smyrnis, G., Shankar, V., Gururangan, S., Wortsman, M., Shao, R., Mercat, J., Fang, A., Li, J., Keh, S., Xin, R., Nezhurina, M., Vasiljevic, I., Jitsev, J., Soldaini, L., Dimakis, A. G., Ilharco, G., Koh, P. W., Song, S., Kollar, T., Carmon, Y., Dave, A., Heckel, R., Muennighoff, N., and Schmidt, L. Language models scale reliably with over-training and on downstream tasks, 2024. URL <https://arxiv.org/abs/2403.08540>.
- Goyal, S., Maini, P., Lipton, Z. C., Raghunathan, A., and Kolter, J. Z. Scaling laws for data filtering - data curation cannot be compute agnostic. *CoRR*, abs/2404.07177, 2024. doi: 10.48550/ARXIV.2404.07177. URL <https://doi.org/10.48550/arXiv.2404.07177>.
- Groeneveld, D., Beltagy, I., Walsh, P., Bhagia, A., Kinney, R., Tafjord, O., Jha, A. H., Ivison, H., Magnusson, I., Wang, Y., Arora, S., Atkinson, D., Authur, R., Chandu, K. R., Cohan, A., Dumas, J., Elazar, Y., Gu, Y., Hessel, J., Khot, T., Merrill, W., Morrison, J., Muennighoff, N., Naik, A., Nam, C., Peters, M. E., Pyatkin, V., Ravichander, A., Schwenk, D., Shah, S., Smith, W., Strubell, E., Subramani, N., Wortsman, M., Dasigi, P., Lambert, N., Richardson, K., Zettlemoyer, L., Dodge, J., Lo, K., Soldaini, L., Smith, N. A., and Hajishirzi, H. Olmo: Accelerating the science of language models, 2024. URL <https://arxiv.org/abs/2402.00838>.
- Gu, Y., Tafjord, O., Kuehl, B., Haddad, D., Dodge, J., and Hajishirzi, H. Olmes: A standard for language model evaluations, 2024. URL <https://arxiv.org/abs/2406.08446>.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022. URL <https://arxiv.org/abs/2203.15556>.
- Kang, F., Sun, Y., Wen, B., Chen, S., Song, D., Mahmood, R., and Jia, R. Autoscale: Automatic prediction of compute-optimal data composition for training llms. *ArXiv*, abs/2407.20177, 2024. URL <https://api.semanticscholar.org/CorpusID:271533897>.
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>.
- Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tülu 3: Pushing frontiers in open language model post-training, 2024.
- Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving quantitative reasoning problems with language models, 2022. URL <https://arxiv.org/abs/2206.14858>.
- Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., Gadre, S., Bansal, H., Guha, E., Keh, S., Arora, K., Garg, S., Xin, R., Muennighoff, N., Heckel, R., Mercat, J., Chen, M., Gururangan, S., Wortsman, M., Albalak, A., Bitton, Y., Nezhurina, M., Abbas, A., Hsieh, C.-Y., Ghosh, D., Gardner, J., Kilian, M., Zhang, H., Shao, R., Pratt, S., Sanyal, S., Ilharco, G., Daras, G., Marathe, K., Gokaslan, A., Zhang, J., Chandu, K., Nguyen, T., Vasiljevic, I., Kakade, S., Song, S., Sanghavi, S., Faghri, F., Oh, S., Zettlemoyer, L., Lo, K., El-Nouby, A., Pouransari, H., Toshev, A., Wang, S., Groeneveld, D., Soldaini, L., Koh, P. W., Jitsev, J., Kollar, T., Dimakis, A. G., Carmon, Y., Dave, A., Schmidt, L., and Shankar, V. Datacomp-llm: In search of the next generation of training sets for language models, 2024. URL <https://arxiv.org/abs/2406.11794>.
- Magnusson, I., Bhagia, A., Hofmann, V., Soldaini, L., Jha, A. H., Tafjord, O., Schwenk, D., Walsh, E. P., Elazar, Y., Lo, K., Groeneveld, D., Beltagy, I., Hajishirzi, H., Smith, N. A., Richardson, K., and Dodge, J. Paloma: A

- benchmark for evaluating language model fit, 2024. URL <https://arxiv.org/abs/2312.10523>.
- Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *EMNLP*, pp. 2381–2391, Brussels, Belgium, October–November 2018. doi: 10.18653/v1/D18-1260.
- Muennighoff, N., Rush, A., Barak, B., Le Scao, T., Tazi, N., Piktus, A., Pyysalo, S., Wolf, T., and Raffel, C. A. Scaling data-constrained language models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 50358–50376. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9d89448b63ce1e2e8dc7af72c984c196-Paper-Conference-EMNLP.pdf.
- OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., Guerquin, M., Ivison, H., Koh, P. W., Liu, J., Malik, S., Merrill, W., Miranda, L. J. V., Morrison, J., Murray, T., Nam, C., Pyatkin, V., Rangapur, A., Schmitz, M., Skjongsberg, S., Wadden, D., Wilhelm, C., Wilson, M., Zettlemoyer, L., Farhadi, A., Smith, N. A., and Hajishirzi, H. 2 olmo 2 furious, 2025. URL <https://arxiv.org/abs/2501.00656>.
- Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R.-A., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., and Launay, J. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *ArXiv*, abs/2306.01116, 2023. URL <https://api.semanticscholar.org/CorpusID:259063761>.
- Penedo, G., Kydlíček, H., allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. V., and Wolf, T. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL <https://arxiv.org/abs/2406.17557>.
- Porian, T., Wortsman, M., Jitsev, J., Schmidt, L., and Carmon, Y. Resolving discrepancies in compute-optimal scaling of language models. *ArXiv*, abs/2406.19146, 2024. URL <https://api.semanticscholar.org/CorpusID:270764838>.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.
- Ruan, Y., Maddison, C. J., and Hashimoto, T. Observational scaling laws and the predictability of language model performance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=On5WIN7xyD>.
- Sakaguchi, K., Le Bras, R., Bhagavatula, C., and Choi, Y. WinoGrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, Apr. 2020. doi: 10.1609/aaai.v34i05.6399. URL <https://ojs.aaai.org/index.php/AAAI/article/view/6399>.
- Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social IQa: Commonsense reasoning about social interactions. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *EMNLP*, pp. 4463–4473, Hong Kong, China, November 2019. doi: 10.18653/v1/D19-1454.
- Schaeffer, R., Miranda, B., and Koyejo, S. Are emergent abilities of large language models a mirage?, 2023. URL <https://arxiv.org/abs/2304.15004>.
- Schaeffer, R., Schoelkopf, H., Miranda, B., Mukobi, G., Madan, V., Ibrahim, A., Bradley, H., Biderman, S., and Koyejo, S. Why has predicting downstream capabilities of frontier AI models with scale remained elusive? In *Trustworthy Multi-modal Foundation Models and AI Agents (TiFA)*, 2024. URL <https://openreview.net/forum?id=AbHHrj9afB>.
- Soldaini, L., Kinney, R., Bhagia, A., Schwenk, D., Atkinson, D., Authur, R., Bogin, B., Chandu, K., Dumas, J., Elazar, Y., Hofmann, V., Jha, A. H., Kumar, S., Lucy, L., Lyu, X., Lambert, N., Magnusson, I., Morrison, J., Muennighoff, N., Naik, A., Nam, C., Peters, M. E., Ravichander, A., Richardson, K., Shen, Z., Strubell, E., Subramani, N., Tafjord, O., Walsh, P., Zettlemoyer, L., Smith, N. A., Hajishirzi, H., Beltagy, I., Groeneveld, D., Dodge, J., and Lo, K. Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2024.
- Talmor, A., Herzig, J., Lourie, N., and Berant, J. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Burstein, J., Doran, C., and Solorio, T. (eds.), *NAACL*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. doi: 10.18653/v1/N19-1421.
- Ye, J., Liu, P., Sun, T., Zhou, Y., Zhan, J., and Qiu, X. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *ArXiv*, abs/2403.16952, 2024. URL <https://api.semanticscholar.org/CorpusID:268681464>.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *ACL*, pp. 4791–4800, Florence, Italy, July 2019. doi: 10.18653/v1/P19-1472.

Zhou, F., Wang, Z., Liu, Q., Li, J., and Liu, P. Programming every example: Lifting pre-training data quality like experts at scale. *arXiv preprint arXiv:2409.17115*, 2024.

A. Hyperparameters

Table 2 provides OLMo model ladder configurations for all models in DATADECIDE.

B. Proxy Metric Definitions

Table 3 provides formal definitions for our proxy metrics (§2.5).

C. Scaling Law Variants

Baseline 3-parameter fit. Our default setup (described in §2.2) follows the two-step fit from (Bhagia et al., 2024) and uses Equation 1 to map compute C to task loss L , and Equation 2 to map task loss to metric score. This variant fits three parameters (A, α, E) in the first step.

2-parameter fit. This is a restricted version of the baseline where the irreducible loss term E is removed from Equation 1, leaving only two parameters:

$$L(C) = \frac{A}{C^\alpha} \tag{4}$$

5-parameter (N, D) fit. Instead of modeling loss as a function of compute C , this variant uses both number of tokens N and number of parameters D directly in the loss function:

$$L(N, D) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + E \tag{5}$$

This introduces five parameters: A, α, B, β , and E .

Single-step prediction. In this variant, the two-stage fitting procedure is replaced with a single step that directly maps compute C to accuracy:

$$Acc(C) = \frac{a}{1 + \exp(-k(\frac{A}{C^\alpha} + E - L_0))} + b \tag{6}$$

This combines the loss and accuracy mapping into one function.

5-parameter, single step. We also test a single-step variant that directly maps from (N, D) to accuracy using a logistic function over the predicted loss. This merges Equations 5 and 2 into:

$$Acc(N, D) = \frac{a}{1 + \exp(-(\frac{A}{N^\alpha} + \frac{B}{D^\beta} + E))} + b \tag{7}$$

This formulation retains the same five parameters from the two-step (N, D) loss function. Following Bhagia et al. (2024), we merge the parameters k and L_0 from the second-stage sigmoid into the loss-side parameters (A, B, E) , yielding a simplified single-stage fit with 7 total free parameters: $\{A, \alpha, B, \beta, E, a, b\}$.

DataDecide: How to Predict Best Pretraining Data with Small Experiments

Model name	Batch size	Hidden dim.	LR	Model size	Heads	Layers	Training steps	Tokens trained
4M	32	64	1.4e-02	3.7M	8	8	5,725	0.4B
6M	32	96	1.2e-02	6.0M	8	8	9,182	0.6B
8M	32	128	1.1e-02	8.5M	8	8	13,039	0.9B
10M	32	144	1.0e-02	9.9M	8	8	15,117	1.0B
14M	32	192	9.2e-03	14.4M	8	8	21,953	1.4B
16M	32	208	8.9e-03	16.0M	8	8	24,432	1.6B
20M	64	192	8.4e-03	19.1M	8	16	14,584	1.9B
60M	96	384	5.8e-03	57.1M	12	16	29,042	5.7B
90M	160	528	4.9e-03	97.9M	12	16	29,901	9.8B
150M	192	768	4.2e-03	151.9M	12	12	38,157	15.0B
300M	320	1,024	3.3e-03	320.0M	16	16	45,787	30.0B
530M	448	1,344	2.8e-03	530.1M	16	16	57,786	53.0B
750M	576	1,536	2.5e-03	681.3M	16	16	63,589	75.0B
1B	704	2,048	2.1e-03	1176.8M	16	16	69,369	100.0B

Table 2. DATADECIDE uses OLMo’s *model ladder* (Groeneveld et al., 2024; OLMo et al., 2025; Bhagia et al., 2024) to programmatically create configurations for 14 model sizes with hyperparameters determined by heuristics in Porian et al. (2024). All models have sequence length of 2024 and MLP ratio of 8. Each configuration is pretrained over 25 data recipes (Table 1). Each recipe and configuration is also trained for 3 random seeds where model sizes < 1B are stopped early at 25% of the compute used to train the 1B model for all but the default seed. Model size is number of non-embedding parameters. Batch size is the number of sequences per batch.

Metric Name	Equation
CORRECT PROB	$\frac{1}{N} \sum_{i=1}^N P(c_{\text{correct}}^{(i)} \text{context}_i)$
MARGIN	$\frac{1}{N} \sum_{i=1}^N (P(c_{\text{correct}}^{(i)} \text{context}_i) - \max_{c' \neq c_{\text{correct}}^{(i)} \in C^{(i)}} P(c' \text{context}_i))$
NORM CORRECT PROB	$\frac{1}{N} \sum_{i=1}^N \frac{P(c_{\text{correct}}^{(i)} \text{context}_i)}{\sum_{c \in C^{(i)}} P(c \text{context}_i)}$
TOTAL PROB	$\frac{1}{N} \sum_{i=1}^N \sum_{c \in C^{(i)}} P(c \text{context}_i)$
ACCURACY	$\frac{1}{N} \sum_{i=1}^N \mathbb{I}(\arg \max_{c \in C^{(i)}} P(c \text{context}_i) = c_{\text{correct}}^{(i)})$
*_per_token	$\log(P(c \text{context})) / \text{tokens}(c)$
*_per_char	$\log(P(c \text{context})) / \text{chars}(c)$

Table 3. Proxy metrics used as alternative inputs to our prediction methods, $C^{(i)}$ is the set of possible continuations for item i and N is the number of items in a benchmark. Each each of the first 5 metrics have *_per_token and *_per_char variants in which likelihoods are normalized as defined in the bottom two rows.

Scaling Law Variant	Relative Error	Absolute Error
3-parameter with helpers and >50% checkpoints	5.6	2.6
3-parameter with helper points	6.0	2.8
3-parameter step 2 fit with >50% checkpoints	5.9	2.9
3-parameter	6.5	3.1
2-parameter	6.5	3.2
5-parameter, single step	42.8	17.4
3-parameter, single step	42.9	42.3
5-parameter	230.8	65.4

Table 4. Average prediction error for 1B targets for the different scaling law setups across tasks and recipes on ACCURACY fit to all models but 1B. We see that other than the single step and 5-parameter variants errors are comparable, and these variants also roughly follow the compute-decision frontier in Figure 3.

Use of helper points. Following Bhagia et al. (2024), we optionally include an extra point ($L = 0.0$, $Acc = 1.0$) in the second-stage fit. This “helper” point anchors the upper asymptote of the accuracy prediction.

Filtering early checkpoints. We experiment with excluding the first 50% of intermediate checkpoints when fitting the second-stage sigmoid. This reduces noise from high-loss early training points and often improves the fit for extrapolation.

Helpers and > 50% checkpoints. Lastly we experiment with combining the previous two techniques on the baseline 3-parameter fit.

Prediction Error. We report prediction errors in Table 4 for each setup. As the best scaling laws variants are all roughly comparable to the simple 3-parameter set up, we use this one as our baseline.