LANGUAGE MODELS SCALE RELIABLY WITH OVER-TRAINING AND ON DOWNSTREAM TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling laws are useful guides for derisking expensive training runs, as they predict performance of large models using cheaper, small-scale experiments. However, there remain gaps between current scaling studies and how language models are ultimately trained and evaluated. For instance, scaling is usually studied in the compute-optimal training regime (i.e., "Chinchilla optimal" regime). In contrast, models are often over-trained to reduce inference costs. Moreover, scaling laws mostly predict loss on next-token prediction, but models are usually compared on downstream task performance. To address both shortcomings, we create a testbed of 104 models with 0.011B to 6.9B parameters trained with various numbers of tokens on three data distributions. First, we fit scaling laws that extrapolate in both the amount of over-training and the number of model parameters. This enables us to predict the validation loss of a 1.4B parameter, 900B token run (i.e., 32× over-trained) and a 6.9B parameter, 138B token run (i.e., a compute-optimal run)—each from experiments that take 300× less compute. Second, we relate the perplexity of a language model to its downstream task performance by proposing a power law. We use this law to predict top-1 error averaged over downstream tasks for the two aforementioned models, using experiments that take 20× less compute. To facilitate further research on reliable scaling, we will provide all results of our experiments.

1 INTRODUCTION

Training large language models is expensive. Furthermore, training high-quality models requires a complex recipe of algorithmic techniques and training data. To reduce the cost of finding successful training recipes, researchers first evaluate ideas with small experiments and then extrapolate their efficacy to larger model and data regimes via scaling laws. With reliable extrapolation, it is possible to quickly iterate at small scale and still pick the method that will perform best for the final large training run. Indeed, this workflow has become commonplace for training state-of-the-art language models like Chinchilla 70B (Hoffmann et al., 2022), PaLM 540B (Chowdhery et al., 2022), GPT-4 (OpenAI, 2023), and many others.

Despite their importance for model development, published scaling laws differ from the goals of training state-of-the-art models in important ways. For instance, scaling studies usually focus on the compute-optimal training regime ("Chinchilla optimality" (Hoffmann et al., 2022)), where model and dataset size are set to yield minimum loss for a given compute budget. However, this setting ignores inference costs. As larger models are more expensive at inference, it is now common practice to over-train smaller models (Touvron et al., 2023a). Another potential mismatch is that most scaling laws quantify model performance by perplexity in next-token prediction instead of accuracy on widely used benchmark datasets. However, practitioners usually turn to benchmark performance, not loss, to compare models.

In this paper, we conduct an extensive set of experiments to address both scaling in the over-trained regime and benchmark performance prediction.

Motivated by the practice of training beyond compute-optimality, we first investigate whether scaling follows reliable trends in the over-trained regime. We notice, as implied by Hoffmann et al. (2022), for a set of models of different sizes trained with a constant ratio of tokens to parameters, models' reducible loss L' (Hestness et al., 2017; Hoffmann et al., 2022) follows a power law ($L' = \lambda \cdot C^{-\eta}$)

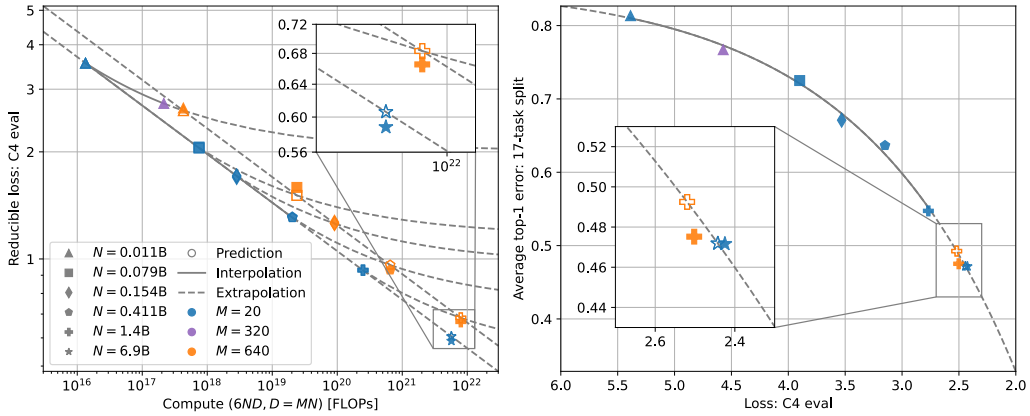

Figure 1: **Reliable scaling with over-training and on downstream error prediction.** *(left)* We fit a scaling law for model validation loss, parameterized by (i) a token multiplier $M = D/N$, which is the ratio of training tokens D to parameters N and (ii) the compute C in FLOPs used to train a model, approximated by $C = 6ND$. Larger values of M specify more over-training. We are able to extrapolate, in both N and M, the validation performance of models requiring more than $300\times$ the training compute used to construct the scaling law. *(right)* We also fit a scaling law to predict average downstream top-1 error as a function of validation loss. We find that fitting scaling laws for downstream error benefits from using more expensive models when compared to fitting for loss prediction. We predict the average error over 17 downstream tasks for models trained with over $20\times$ the compute. For this figure, we train all models on RedPajama (Together Computer, 2023).

in the amount of training compute C. We find that as one increases the ratio of tokens to parameters, corresponding to more over-training, the scaling exponent η remains about the same, while the scalar λ changes. We explain our observations by reparameterizing existing scaling laws in relation to the amount of over-training.

To establish empirically that scaling *extrapolates* in the over-trained regime, we further experiment with a testbed of 104 models, trained from scratch on three different datasets: C4 (Raffel et al., 2019; Dodge et al., 2021), RedPajama (Together Computer, 2023), and RefinedWeb (Penedo et al., 2023). We find that scaling laws fit to small models can accurately predict the performance of larger models that undergo more over-training. Figure 1 *(left)* illustrates our main over-training result, where we invest $2.4e19$ FLOPs to extrapolate the C4 validation performance of a 1.4B parameter model trained on 900B tokens, which requires $300\times$ more compute to train.

In addition to over-training, we also investigate if scaling laws can predict the performance of a model on downstream tasks. We establish a power law relationship between language modeling perplexity and the average top-1 error on a suite of downstream tasks. While it can be difficult to predict the error on individual tasks, we find it possible to predict aggregate performance from a model's perplexity among models trained on the same training data. Figure 1 *(right)* presents our main downstream error prediction result, where we invest $2.7e20$ FLOPs to predict the average top-1 error over a set of downstream tasks to within 1 percentage point for a 6.9B compute-optimal model, which requires $20\times$ more compute to train.

Our results suggest that the proposed scaling laws are promising to derisk (i) the effects of over-training models and (ii) the downstream performance of scaling up training recipes. To facilitate further research on reliable scaling, we will provide all results of our experiments.

2 Developing scaling laws for over-training and downstream tasks

In this section, we develop scaling laws to predict over-trained and downstream performance. First, we provide key definitions (Section 2.1). We next present a scaling law for over-training drawing on empirical observation and prior work (Section 2.2). To connect loss scaling and downstream error

prediction, we observe that average top-1 error decreases exponentially as a function of validation loss, which we formalize as a novel scaling law (Section 2.3). In later sections, we build an experimental setup (Section 3) to quantify the extent to which our scaling laws extrapolate reliably (Section 4).

2.1 PRELIMINARIES

Scaling laws for loss. Typically, scaling laws predict model loss L as a function of the compute C in FLOPs used for training. If one increases the number of parameters N in a model or the number of tokens D that a model is trained on, compute requirements naturally increase. Hence, we assume C is a function of N, D. Following Kaplan et al. (2020), we use the approximation $C = 6ND$, which Hoffmann et al. (2022) independently verify. We consider,

$$L(C) = E + L'(C), \tag{1}$$

where E is an *irreducible loss* and L' is the *reducible loss*. E captures the Bayes error or minimum possible loss achievable on the validation domain. The $L'(C)$ term captures what can possibly be learned about the validation domain by training on a source domain. $L'(C)$ should approach zero with increased training data and model capacity. $L'(C)$ is often assumed to follow a power law: $L'(C) = \lambda \cdot C^{-\eta}$ (i.a., Hestness et al. (2017); OpenAI (2023)). It is also often helpful to consider a power law in a log-log plot, where it appears as a line with slope $-\eta$ and y-intercept $\log(\lambda)$.

Token multipliers. We define a token multiplier $M = D/N$ as the ratio of training tokens to model parameters for notational convenience. M allows us to consider fixed relationships between D and N even as a model gets bigger (i.e., as N becomes larger).

Compute-optimal training. Hoffmann et al. (2022) establish compute-optimal training, where, for any compute budget H, the allocation of parameters and tokens is given by,

$$\arg\min_{N,D} L(N, D) \text{ s.t. } C(N, D) = H. \tag{2}$$

To solve for the optimal N^*, D^*, one can sweep N, D for each compute budget, retaining the best configurations. Hoffmann et al. (2022) find that as the compute budget increases, N^* and D^* scale roughly evenly. Assuming equal scaling, there is a fixed compute-optimal token multiplier $M^* = D^*/N^*$ per training distribution.

Over-training. We define over-training as the practice of allocating compute sub-optimally, so smaller models train on a disproportionately large number of tokens (i.e., $M > M^*$). While loss should be higher than in the compute-optimal allocation for a given training budget, the resulting models have fewer parameters and thus incur less inference cost.

2.2 SCALING LAWS FOR OVER-TRAINING

To propose a scaling law for over-trained models, we first turn to empirical observation. We train four model configurations with parameter counts between 0.011B and 0.411B for token multipliers M between 20 and 640, where $M = 20$ points lie roughly on the compute-optimal frontier, and larger M corresponds to more over-training. We defer experimental details to Section 3 to focus on our observations first. In Figure 2, we show loss against compute in a log-log plot for the models trained on three datasets and evaluated on the C4 eval set. We notice parallel lines when fitting power laws to the reducible loss, which suggests a near-constant scaling exponent even with increased over-training. This indicates that scaling behavior should be describable in the amount of over-training.

In search of an analytic expression for the observations in Figure 2, we consider existing scaling literature. A common functional form for the risk of a model, as proposed in prior work (Rosenfeld et al., 2020; Hoffmann et al., 2022) is,

$$L(N, D) = E + AN^{-\alpha} + BD^{-\beta}. \tag{3}$$

Recall from Section 2.1, N is the number of parameters and D the number of training tokens. The constants E, A, α, B, β are fit from data. By fitting this parametric form, Hoffmann et al. (2022) find that scaling exponents α and β are roughly equal, suggesting that one should scale N and D

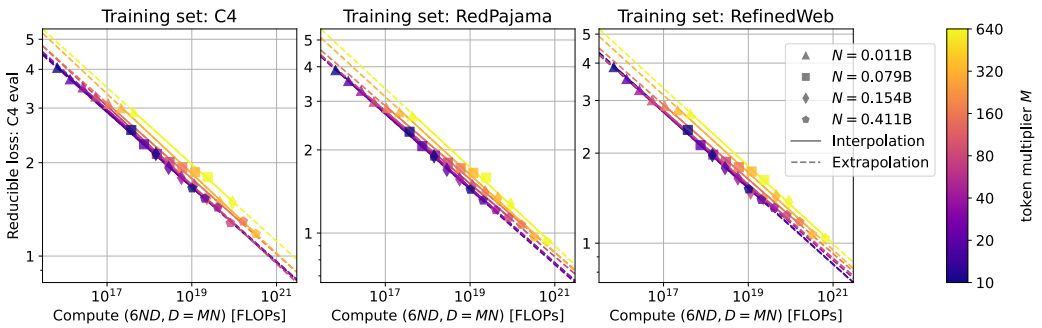

Figure 2: **Scaling in the over-trained regime follows consistent power law exponents.** We notice parallel lines in the log-log plots of reducible loss vs. training compute for a range of token multipliers M, which give the ratio of training tokens to model parameters. Larger M corresponds to more over-training. For a power law giving reducible loss as a function of compute: $L'(C) = \lambda \cdot C^{-\eta}$, the exponent η remains relatively constant resulting in lines with approximately fixed slope (Figure 17). The scalar λ that determines the y-intercept, however, shifts with different token multipliers. This suggests λ is a function of the token multiplier, while η is not.

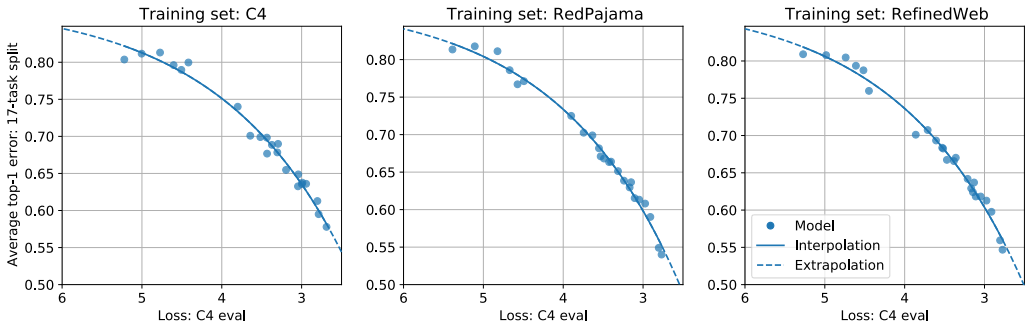

Figure 3: **Average top-1 error scales as a function of loss.** We plot models trained on three datasets and notice an exponential decay of average top-1 error as C4 eval loss, on the x-axis, decreases. We consider on the y-axes average error on 17 evaluations where performance is at least 10 points above random chance for at least one 0.154B scale model. These observations suggest that average top-1 error should be predictable with reliable loss estimates.

equally as compute increases. Hence, we assume $\alpha = \beta$. With this assumption, we reparameterize Equation (3) in terms of compute $C = 6ND$ and a token multiplier $M = D/N$. We get,

$$L(C, M) = E + \left(aM^{\eta} + bM^{-\eta}\right) C^{-\eta}, \tag{4}$$

where $\eta = \alpha/2$, $a = A(1/6)^{-\eta}$, $b = B(1/6)^{-\eta}$ gives the relation to Equation (3). For a complete derivation, see Appendix A.

Equation (4) has the following interpretation: (i) The scaling exponent η is not dependent on M. Thus, we always expect lines with the same slope in the log-log plot—as in Figure 2. (ii) The term $aM^{\eta} + bM^{-\eta}$ determines the offsets between curves with different token multipliers. Hence, we expect non-overlapping, parallel lines in the log-log plot for the range of M we consider—also consistent with Figure 2.

Recall that we make the assumption $\alpha = \beta$, which implies equal scaling of parameters and tokens as more compute is available. However, as explained in Appendix A, even if $\alpha \neq \beta$, we get a parameterization that implies the power-law exponent remains constant with over-training.

2.3 Scaling laws for downstream error

Scaling is typically studied in the context of loss (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023b), which Schaeffer et al. (2023) note is smoother than metrics like accuracy.

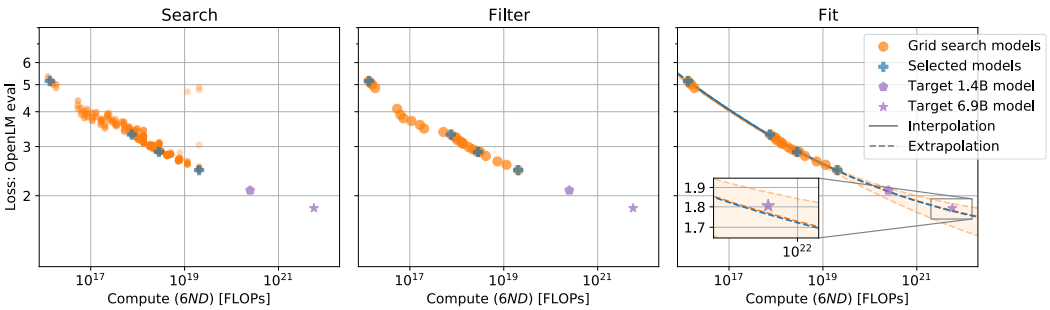

Figure 4: **Search, filter, fit: A recipe for selecting configurations for scaling.** *(left)* To generate the final configurations presented in Table 3, we run a 435 model grid search over model width, hidden dimension, number of attention heads, batch size, and warmup steps. All models are trained near compute-optimally. *(center)* We plot the efficient frontier of models, which appear to follow a trend, excluding models from 5.2×10^{16} to 5.2×10^{17}, which fall below the trend. *(right)* We fit a power law with irreducible error to the remaining configurations, picking four configurations that closely track the full model suite ("Selected models"). These models extrapolate the performance of 1.4B, 6.9B target models. Shaded regions represent bootstrap 95% confidence intervals.

However, practitioners often use downstream benchmark accuracy as a proxy for model quality and not loss on perplexity evaluation sets. To better connect scaling laws and over-training to task prediction, we revisit the suite of models plotted in Figure 2. In Figure 3, we plot average downstream top-1 errors over evaluations sourced from LLM-Foundry (MosaicML, 2023) against the C4 eval loss. We defer details of the setup to Section 3 to focus here on a key observation: average error appears to follow exponential decay as loss decreases.

Based on the exponential decay we observe in Figure 3, we propose the following relationship between downstream average top-1 error Err and loss L,

$$\mathsf{Err}(L) = \epsilon - k \cdot \exp\left(-\gamma L\right), \tag{5}$$

where ϵ, k, γ are fit from data. Equation (5) also has an interpretation in terms of model perplexity $\mathsf{PP}(L) = \exp\left(L\right)$,

$$\mathsf{Err}(\mathsf{PP}) = \epsilon - k \cdot \mathsf{PP}^{-\gamma}. \tag{6}$$

Namely, Err follows a power law in PP that is bounded from above by ϵ signifying arbitrarily high error and from below by $\epsilon - k \cdot \exp(-\gamma E)$, where E is the Bayes error from Equation (4).

Equation (5) in conjunction with Equation (4) suggests a three-step method to predict Err as a function of compute and the amount of over-training. For choices of training and validation distributions, (i) fit a scaling law to Equation (4) using triplets of compute C, token multiplier M, and measured loss L on a validation set to yield $(C, M) \mapsto L$. (ii) Fit a scaling law to Equation (5) using pairs of loss L and downstream error Err for models to get $L \mapsto \mathsf{Err}$. (iii) Chain predictions to get $(C, M) \mapsto \mathsf{Err}$.

3 Constructing a Scaling Testbed

In this section, we discuss our experimental setup to test the predictions suggested by Equations (4) and (5). We first present our general language modeling setup (Section 3.1). Next, we discuss our strategy for determining model configurations for our scaling investigation (Section 3.2) and fitting scaling laws (Section 3.3). We then present metrics to validate how well scaling laws predict loss and downstream performance (Section 3.4).

3.1 Training Setup

We train transformers (Vaswani et al., 2017) for next token prediction, based on architectures like GPT-2 (Radford et al., 2019) and LLaMA (Touvron et al., 2023a). We employ GPT-NeoX (Black et al., 2022) as a standardized tokenizer for all data. See Appendix B for architecture, optimization, and hyperparameter details.

3.2 MODEL CONFIGURATIONS

To get final configurations for the 0.011B to 0.411B parameter models plotted in Figures 2 and 3, we first conduct a wide grid search over a total of 435 models, trained from scratch, from 0.01B to 0.5B parameters (Figure 4 *(left)*). We train on the original OpenLM data mix (Gururangan et al., 2023), which largely consists of RedPajama (Together Computer, 2023) and The Pile (Gao et al., 2020). While we eventually plan to over-train models, at this step we search for *base configurations* near compute-optimality. We train on 20 tokens per parameter ($M = 20$), which, in early experiments, gives models near the compute-optimal frontier. This is similar to findings in Hoffmann et al. (2022)'s Table 3, which suggests that $M = 20$ is near-optimal for the Chinchilla experimental setup.

To find maximally performant small-scale models on validation data, we tune model width, number of layers, number of attention heads, warmup steps, and batch size. Our validation set, OpenLM eval, contains tokens from recent arXiv papers, the OpenLM codebase itself, and news articles. We find in early experiments that qk-LayerNorm makes models less sensitive to learning rate, which is a phenomenon Wortsman et al. (2023) report in their Figure 1. Hence, we fix the learning rate ($3e$-3) for our sweeps. We also perform smaller grid searches over 1.4B and 6.9B parameter model configurations at $M = 20$, retaining the best configurations.

At this point, we have many models, several of which give poor performance; following prior work (Kaplan et al., 2020; Hoffmann et al., 2022), we want to keep only models that give best performance. Hence, in Figure 4 *(center)*, we filter out models that do not lie on the Pareto frontier. While there appears to be a general trend, configurations between 5.2×10^{16} and 5.2×10^{17} FLOPs lie below the frontier established by other models. We hypothesize these models over-perform as they are trained for more optimization steps than their neighbors based on our power-of-two batch sizes. We provide support for this hypothesis in Appendix E, but opt to remove these models from our investigation.

To ensure tractable compute requirements for our scaling experiments, we require a subset of models that follows the trend of the entire Pareto frontier. In Figure 4 *(right)*, we fit trends to the Pareto models and to a subset of four models. We notice that the trends closely predict both the performance of the 1.4B and 6.9B models, suggesting that our small-scale configurations reliably extrapolate in the compute-optimal setting.

Moving forward, we do not tune hyperparameters for other token multipliers (i.e., $M \neq 20$), on other training or evaluation distributions, or on validation sets for downstream tasks. For more details including specific hyperparameters, see Appendix C.

To create our scaling testbed, we start with the four small-scale, base configurations from our grid search: $N \in \{0.011\text{B}, 0.079\text{B}, 0.154\text{B}, 0.411\text{B}\}$. To ensure our conclusions are not particular to a single training distribution, we train models on each of C4 (Raffel et al., 2019; Dodge et al., 2021), RedPajama (Together Computer, 2023), and RefinedWeb (Penedo et al., 2023), which have 138B, 1.15T, and 600B tokens, respectively, for different token multipliers $M \in \{5, 10, 20, 40, 80, 160, 320, 640\}$. We omit runs that require more tokens than are present in a dataset (i.e., $N = 0.411\text{B}, M = 640$ for C4). We additionally train $N = 1.4\text{B}$ models at $M = 20$ and at the largest token multiplier possible without repeating tokens (i.e., 80 for C4, 640 for RedPajama, and 320 for RefinedWeb). We train $N = 6.9\text{B}, M = 20$ models on each dataset given the relevance of 7B parameter models (Touvron et al., 2023a; Jiang et al., 2023). In total this results in a testbed of 104 models.

3.3 FITTING SCALING LAWS

We fit Equation (4) to approximate E, a, b, η using curve-fitting in SciPy (Virtanen et al., 2020) (i.e., Levenberg-Marquardt to minimize non-linear least squares). We repeat this process to fit Equation (5) to approximate ϵ, k, γ. We invest $\sim$100 A100 hours to train the models required to fit a scaling law for loss and $\sim$1,000 A100 hours for a corresponding law for downstream error. Unless otherwise specified, we fit to the N, M pairs in Table 1, which are a subset of our full testbed. Our configurations allow us to test for extrapolation to the $N = 1.4\text{B}, M = 640$ (900B token) and the $N = 6.9\text{B}, M = 20$ (138B token) regimes.

Table 1: **Default number of parameters N and token multiplier M to fit our scaling laws.** We invest ∼100 A100 hours to fit Equation (4) and ∼1,000 A100 hours to fit Equation (5).

N	M	Used to fit Equation (4)	Used to fit Equation (5)
0.011B	20	✓	✓
0.079B	20	✓	✓
0.154B	20	✓	✓
0.411B	20	✓	✓
0.011B	320	✓	✓
1.4B	20	✗	✓
Total compute C [FLOPs]		$2.4e19$	$2.7e20$

3.4 Evaluation setup

Evaluation datasets. Unless otherwise stated, our default validation loss dataset is C4 eval. For downstream tasks, we adopt a subset from 46 tasks from LLM-foundry (MosaicML, 2023), which includes standard tasks with both zero-shot and few-shot evaluations. Specifically, we consider a 17-task subset where, for each evaluation, at least one 0.154B scale model—trained with as many as 99B tokens—gets 10 percentage points above chance accuracy: ARC-Easy (Clark et al., 2018), BIG-bench: CS algorithms (bench authors, 2023), BIG-bench: Dyck languages (bench authors, 2023), BIG-bench: Novel Concepts (bench authors, 2023), BIG-bench: Operators (bench authors, 2023), BIG-bench: QA WikiData (bench authors, 2023), BoolQ (Clark et al., 2019), Commonsense QA (Talmor et al., 2019), COPA (Roemmele et al., 2011), CoQA (Reddy et al., 2019), HellaSwag (zero-shot) (Zellers et al., 2019), HellaSwag (10-shot) (Zellers et al., 2019), LAMBADA (Paperno et al., 2016), PIQA (Bisk et al., 2020), PubMed QA Labeled (Jin et al., 2019), SQuAD (Rajpurkar et al., 2016), and WinoGrand (Levesque et al., 2012). For more details on evaluation datasets see Appendix D. We focus on this subset to ensure we are measuring signal, not noise. Including downstream tasks like MMLU (Hendrycks et al., 2021), where performance is close to random chance, however, does not invalidate our results as we show in our evaluation set ablations (Appendix E).

Metrics. We consider three main metrics: *Validation loss*, which is the cross entropy between a model's output and the one-hot ground truth token, averaged over all tokens in a sequence and over all sequences in a dataset. *Average top-1 error*, which is a uniform average over the 17 downstream evaluations, as mentioned in the above paragraph. To measure how good a prediction $\zeta(C, M)$ is, we measure *Relative prediction error*: $|\zeta(C, M) - \zeta_{GT}|/\zeta_{GT}$, where ζ is the predicted loss L or the average top-1 error Err. ζ_{GT} is the ground truth measurement to predict.

4 Results: Reliable extrapolation

In this Section, we quantify the extent to which the scaling laws developed in Section 2 extrapolate larger model performance using the scaling testbed from Section 3. By default, we fit Equations (4) and (5) to the configurations in Table 1, use C4 eval for loss, and the 17-task split from Section 3.4 for average top-1 error.

Over-trained performance is predictable. We highlight our main over-training results in Figure 1 *(left)*. Namely, we are able to extrapolate both in the number of parameters N and the token multiplier M to closely predict the C4 eval performance of a 1.4B parameter model trained on 900B RedPajama tokens ($N = 1.4\text{B}, M = 640$). Our prediction, which takes 300× less compute to construct than the final 1.4B run, is accurate to within 0.7% relative error. Additionally, for the $N = 6.9\text{B}, M = 20$ run, near compute-optimal, the relative error is also 0.7%.

These results support several key takeaways. (i) Scaling can be predictable even when one increases both the model size and the amount of over-training compared to the training runs used to fit a scaling law. (ii) The form presented in Equation (4) is useful in practice for predicting over-trained scaling behavior. (iii) Fitting to Equation (4) gives good prediction accuracy near compute-optimal. More specifically, predictions are accurate both for the 1.4B over-trained model and the 6.7B compute-optimal model using a single scaling fit.

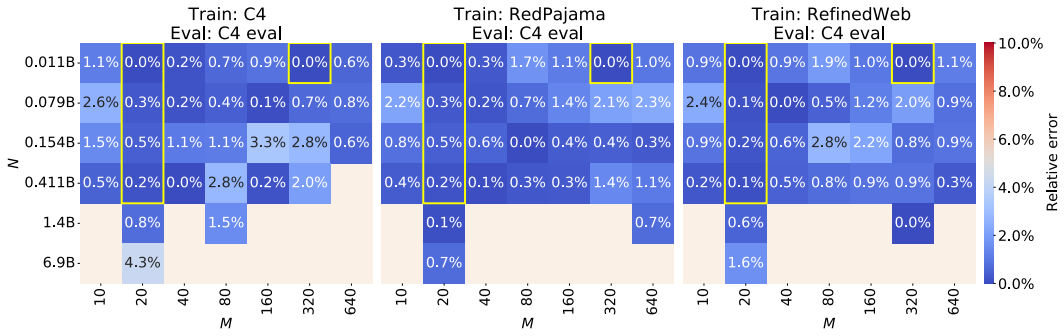

Figure 5: **Relative error on C4 eval for different training distributions.** Boxes highlighted in yellow correspond to pairs—number of parameters N, token multiplier M—used to fit Equation (4). Larger values of M correspond to more over-training. The prediction error is low in both interpolation and extrapolation ranges. Below $N = 1.4$B, empty squares correspond to runs that were not possible due to the limited dataset size for single epoch training. At $N = 1.4$B we run at $M = 20$ and at the largest possible multiplier. At $N = 6.9$B, we run at $M = 20$.

Table 2: **Downstream relative prediction error at 6.9B parameters and 138B tokens.** While predicting accuracy on individual zero-shot downstream evaluations can be challenging ("Individual"), predicting *averages* across downstream datasets is accurate ("Avg.").

	Individual top-1 error				Avg. top-1 error
Train set	ARC-E	LAMBADA	OpenBook QA	HellaSwag	17-task split
C4	28.96%	15.01%	16.80%	79.58%	0.14%
RedPajama	5.21%	14.39%	8.44%	25.73%	0.05%
RefinedWeb	26.06%	16.55%	1.92%	81.96%	2.94%

While Figure 1 explores a specific case of making predictions in the over-trained regime, we aim to understand the error profile of our predictions across training datasets, token multipliers, and number of parameters. Hence, Figure 5 shows the relative error between ground truth loss and predicted loss on C4 eval for models in our testbed. We notice uniformly low prediction error suggesting that predictions are accurate in many settings.

Average top-1 error is predictable. Figure 1 *(right)* presents our main result in estimating scaling laws for downstream error. Concretely, we use the models indicated in Table 1 to fit Equations (4) and (5), chaining the scaling fits to predict the average top-1 error as a function of training compute C and the token multiplier M. Our fits allow us to predict, using $20\times$ less compute, the downstream performance of a 6.9B model trained on 138B RedPajama tokens to within 0.05% relative error and a 1.4B model trained on RedPajama 900B tokens to within 3.6% relative error.

Table 2 additionally shows the relative error of our downstream performance predictions for models trained on C4, RedPajama, and RefinedWeb, indicating that our scaling law functional forms are applicable on many training datasets. We note that while average accuracy is predictable, *individual* downstream task predictions are significantly more noisy. We report relative error for more model predictions in Figures 11 and 12. We also find that if we remove the 1.4B model for the Equation (5) fit, relative error jumps, for instance, from 0.05% to 10.64% on the 17-task split for the 6.9B, 138B token RedPajama prediction. This highlights the importance of investing more compute when constructing scaling laws for downstream task prediction compared to loss prediction.

Under-training, out-of-distribution scaling, compute-reliability trade-offs. In addition to our main results presented above, we include additional results in Appendix E, which we summarize here. First, we notice that when token multipliers become too small (i.e., $M = 5$) scaling becomes unreliable and lies off the trend. Additionally, multipliers other than 20, such as 10, 40, and 80, garner points that are roughly on the compute optimal frontier (Figure 9). This observation suggests that the

compute-optimal multiplier may lie in a range rather than take a single value. To probe the limits of reliable scaling, we attempt to break our scaling laws in out-of-distribution settings. We find that models trained on C4—English filtered—and evaluated on next token prediction on code domains have a high relative error in many cases. Perhaps surprisingly, evaluating the same models on German next token prediction gives reliable loss scaling (Figure 10). We additionally examine the compute necessary to create accurate scaling laws, finding that scaling laws can be constructed more cheaply for loss prediction than for downstream error prediction (Figures 15 and 16).

5 Related work

We review the most closely related work in this section. For additional related work, see Appendix F.

Scaling laws. Early works on scaling artificial neural networks observe predictable power-law scaling in the training set size and number of model parameters (Hestness et al., 2017; 2019; Rosenfeld et al., 2020). Alabdulmohsin et al. (2022) stress the importance of looking at the extrapolation regime of a scaling law. Yang et al. (2021) prescribe architectural and hyperparameter changes when scaling model width to realize performant models; Yang et al. (2024) make analogous recommendations when scaling model depth. Bi et al. (2024) propose hyperparameter aware scaling laws. Unlike the aforementioned work, our investigation focuses on over-training and predicting downstream accuracy.

Hoffmann et al. (2022) investigate how the number of model parameters N and training tokens D should be chosen to minimize loss L given a compute budget C. Hoffmann et al. (2022) find that when scaling up C, both N and D should be scaled equally up to a multiplicative constant (i.e., $N \propto C^{\sim 0.5}$ and $D \propto C^{\sim 0.5}$) to realize compute-optimality. Appendix C of the Chinchilla paper additionally suggests that these findings hold across three datasets. However, Hoffmann et al. (2022) do not verify their scaling laws for training beyond compute-optimality, or for downstream error prediction—both of which are central to our work.

Sardana & Frankle (2023) propose modifications to the Chinchilla formulation to incorporate inference costs into the definition of compute-optimality and solve for various fixed inference budgets. Their key finding, which is critical for our work, is that when taking into account a large enough inference budget, it is optimal to train smaller models for longer than the original Chinchilla recommendations. Our work presupposes that over-training can be beneficial. Instead of solving for inference-optimal schemes, we support empirically a predictive theory of scaling in the over-trained regime. Additionally, we provide experiments across many validation and training sets.

For predicting downstream scaling beyond loss, Isik et al. (2024) relate the number of pre-training tokens to downstream cross-entropy and machine translation BLEU score (Papineni et al., 2002) after fine-tuning. In contrast, we take a holistic approach to evaluation by looking at top-1 error over many natural language tasks. Schaeffer et al. (2023) argue that emergent abilities (Wei et al., 2022b) are a product of non-linear metrics and propose smoother alternatives. As a warmup for why non-linear metrics may be hard to predict, Schaeffer et al. (2023) consider predicting an ℓ length sequence exactly: $\mathsf{Err}(N, \ell) \approx 1 - \mathsf{PP}(N)^{-\ell}$, where N is the number of parameters in a model and PP is its perplexity. This is a special case of our Equations (5) and (6), where the number of training tokens does not appear, $\epsilon = 1, k = 1$, and $\gamma = \ell$. In contrast, we treat ϵ, k, γ as free parameters for a scaling law fit, finding that average error over downstream tasks can make for a predictable metric. Owen (2024) observe the scaling behavior of many open source models on downstream tasks. However, the study does not control for different architectures, training codebases, optimization schemes, and training datasets, etc. In contrast, we create a standardized, open-source setting, which controls these factors.

Over-training in popular models. There has been a rise in over-trained models (Touvron et al., 2023a;b; Llama Team, 2024) and accompanying massive datasets (Together Computer, 2023; Penedo et al., 2023; Soldaini et al., 2024; Albalak et al., 2024). For example, Chinchilla 70B (Hoffmann et al., 2022) is trained with a token multiplier of 20, while LLaMA-2 7B (Touvron et al., 2023b) uses a token multiplier of 290. In our investigation, we look at token multipliers from 5 to 640 for coverage of popular models. The recent Llama3 8B model is a notable outlier, with token multipliers of $\sim$1900. However, it is unclear if, at 15T tokens, Llama3 8B was trained in the single epoch regime we considr in this paper. Practically, training a 1.4B parameter model for worth of tokens is prohibitive in our

setting as 1) due to compute limitations and 2) the 2.8T training token requirement for a single-epoch run, which is larger than public datasets at the time of our training runs.

6 LIMITATIONS, FUTURE WORK, AND CONCLUSION

Limitations and future work. We identify limitations, which provide motivation for future work.

- **Hyperparameters.** While our configurations are surprisingly amenable to reliable scaling across many training and testing distributions without further tuning, there is a need to develop scaling laws that do not require extensive hyperparameter sweeps.
- **Scaling up.** Validating the trends in this paper for even larger runs is a valuable direction. Additionally, repeating our setup for models that achieve non-trivial performance on harder evaluations like MMLU is left to future work.
- **Scaling down.** Actualizing predictable scaling with even cheaper runs is important to make this area of research more accessible, especially for downstream error prediction.
- **Failure cases.** While we present a preliminary analysis of when scaling is unreliable, future work should investigate conditions under which scaling breaks down.
- **Post-training.** It is common to employ fine-tuning interventions after pre-training, which we do not consider. Quantifying to what degree over-training the base model provides benefits *after* post-training is an open area of research.
- **Individual downstream task prediction.** While we find that averaging over many task error metrics can make for a predictable metric, per-task predictions are left to future work.
- **In-the-wild performance.** Downstream task performance is a proxy for the in-the-wild user experience. Analyzing scaling trends in the context of this experience is timely.
- **Dataset curation.** Our work only deals with existing training datasets. Exploring dataset curation for improved model scaling is another promising direction.

Conclusion. We show that the loss of over-trained models, trained past compute-optimality, is predictable. Furthermore, we propose and validate a scaling law relating loss to average downstream task performance. We hope our work will inspire others to further examine the relationship between model training and downstream generalization. Our testbed will be made publicly available, and we hope it will make scaling research more accessible to researchers and practitioners alike.

REFERENCES

Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. In *International Conference on Learning Representations (ICLR)*, 2022. https://arxiv.org/abs/2110.02095.

Ibrahim Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. In *Advances in Neural Information Processing Systems (NeuIPS)*, 2022. https://arxiv.org/abs/2209.06640.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint*, 2024. https://arxiv.org/abs/2402.16827.

Loubna Ben Allal, Raymond Li, Denis Kocetkov, Chenghao Mou, Christopher Akiki, Carlos Munoz Ferrandis, Niklas Muennighoff, Mayank Mishra, Alex Gu, Manan Dey, et al. Santacoder: don't reach for the stars! *arXiv preprint*, 2023. https://arxiv.org/abs/2301.03988.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Conference of the North American Chapter of the Association for Computational Linguistics (NACCL)*, 2019. https://aclanthology.org/N19-1245.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, David Berard, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Laurent Kirsch, Michael Lazos, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2024. https://pytorch.org/blog/pytorch-2-paper-tutorial.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeffrey Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Veselin Stoyanov. Efficient large scale language modeling with mixtures of experts. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. https://aclanthology.org/2022.emnlp-main.804.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint*, 2016. https://arxiv.org/abs/1607.06450.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint*, 2021. https://arxiv.org/abs/2102.06701.

Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Maxim Krikun, Colin Cherry, Behnam Neyshabur, and Orhan Firat. Data scaling laws in nmt: The effect of noise and architecture. In *International Conference on Machine Learning (ICML)*, 2022. https://proceedings.mlr.press/v162/bansal22b.html.

BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. In *Transactions on Machine Learning Research (TMLR)*, 2023. https://openreview.net/forum?id=uyTL5Bvosj.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings ACM conference on fairness, accountability, and transparency (FAccT)*, 2021. https://dl.acm.org/doi/10.1145/3442188.3445922.

DeepSeek-AI Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wen-Hui Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Jun-Mei Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Min Tang, Bing-Li Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Yu Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yi Xiong, Hanwei Xu, Ronald X Xu, Yanhong Xu, Dejian Yang, Yu mei You, Shuiping Yu, Xin yuan Yu, Bo Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghu Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint*, 2024. https://arxiv.org/abs/2401.02954.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint*, 2022. https://arxiv.org/abs/2211.05100.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020. https://arxiv.org/abs/1911.11641.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model. *BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022. https://aclanthology.org/2022.bigscience-1.9.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2005.14165.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. In *International Conference on Learning Representations (ICLR)*, 2023. https://openreview.net/forum?id=sckjveqlCZ.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. https://arxiv.org/abs/2212.07143.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. In *Journal of Machine Learning Research (JMLR)*, 2022. https://arxiv.org/abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint*, 2022. https://arxiv.org/abs/2210.11416.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. https://aclanthology.org/N19-1300.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*, 2020. https://openreview.net/pdf?id=r1xMH1BtvB.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint*, 2018. https://arxiv.org/abs/1803.05457.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://arxiv.org/abs/2205.14135.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023. https://proceedings.mlr.press/v202/dehghani23a.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. https://aclanthology.org/N19-1423.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. https://aclanthology.org/2021.emnlp-main.98.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning (ICML)*, 2022. https://arxiv.org/abs/2112.06905.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint*, 2024. https://arxiv.org/abs/2402.01306.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Mitchell Wortsman Ryan Marten, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Mehdi Cherti Richard Vencu, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. https://arxiv.org/abs/2304.14108.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint*, 2020. https://arxiv.org/abs/2101.00027.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. *arXiv preprint*, 2021. https://arxiv.org/abs/2109.07740.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. https://aclanthology.org/2021.emnlp-main.478.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint*, 2024. https://arxiv.org/abs/2402.00838.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint*, 2023. https://arxiv.org/abs/2312.00752.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. https://openreview.net/forum?id=yWd42CWN3c.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022. https://arxiv.org/abs/2111.00396.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio Cesar, Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *Preprint*, 2023. https://www.microsoft.com/en-us/research/publication/textbooks-are-all-you-need.

Suchin Gururangan, Mitchell Wortsman, Samir Yitzhak Gadre, Achal Dave, Maciej Kilian, Weijia Shi, Jean Mercat, Georgios Smyrnis, Gabriel Ilharco, Matt Jordan, Reinhard Heckel, Alex Dimakis, Ali Farhadi, Vaishaal Shankar, and Ludwig Schmidt. OpenLM: a minimal but performative language modeling (lm) repository, 2023. https://github.com/mlfoundations/open_lm.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. https://arxiv.org/abs/2009.03300.

T. J. Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *arXiv preprint*, 2020. https://arxiv.org/abs/2010.14701.

Danny Hernandez, Jared Kaplan, T. J. Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint*, 2021. https://arxiv.org/abs/2102.01293.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Frederick Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint*, 2017. https://arxiv.org/abs/1712.00409.

Joel Hestness, Newsha Ardalani, and Gregory Diamos. Beyond human-level accuracy: Computational challenges in deep learning. In *Principles and Practice of Parallel Programming (PPoPP)*, 2019. https://arxiv.org/abs/1909.01736.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://arxiv.org/abs/2203.15556.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. In *International Conference on Learning Representations (ICLR)*, 2017. https://arxiv.org/abs/1611.01462.

Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and Sanmi Koyejo. Scaling laws for downstream task performance of large language models. *arXiv*, 2024. https://arxiv.org/abs/2402.04177.

Maor Ivgi, Yair Carmon, and Jonathan Berant. Scaling laws under the microscope: Predicting transformer performance from small scale experiments. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. https://aclanthology.org/2022.findings-emnlp.544.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Florian Bressand Diego de las Casas, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint*, 2023. https://arxiv.org/abs/2310.06825.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. https://aclanthology.org/D19-1259.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint*, 2020. https://arxiv.org/abs/2001.08361.

Tobit Klug, Dogukan Atik, and Reinhard Heckel. Analyzing the sample complexity of self-supervised image reconstruction methods. *arXiv preprint*, 2023. https://arxiv.org/abs/2305.19079.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint*, 2019. http://arxiv.org/abs/1909.11942.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library, 2022. https://github.com/facebookresearch/xformers.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *International conference on the principles of knowledge representation and reasoning*, 2012. https://aaai.org/papers/59-4492-the-winograd-schema-challenge.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020. https://aclanthology.org/2020.acl-main.703.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint*, 2023. https://arxiv.org/abs/2305.06161.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *International Joint Conference on Artificial Intelligence*, 2020. https://arxiv.org/abs/2007.08124.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint*, 2019. http://arxiv.org/abs/1907.11692.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. https://arxiv.org/abs/2201.03545.

AI @ Meta Llama Team. The llama 3 herd of models. *arXiv preprint*, 2024. https://arxiv.org/abs/2407.21783.

Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, et al. The data provenance initiative: A large scale audit of dataset licensing & attribution in ai. *arXiv preprint*, 2023. https://arxiv.org/abs/2310.16787.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*, 2017. https://arxiv.org/abs/1711.05101.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder 2 and the stack v2: The next generation. *arXiv preprint*, 2024. https://arxiv.org/abs/2402.19173.

Risto Luukkonen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, et al. Fingpt: Large generative models for a small language. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. https://aclanthology.org/2023.emnlp-main.164.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groenveld, Iz Beltagy, Hanneneh Hajishirz, Noah A. Smith, Kyle Richardson, and Jesse Dodge. Paloma: A benchmark for evaluating language model fit. *arXiv preprint*, 2023. https://paloma.allen.ai.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. In *Computational Linguistics*, 1993. https://aclanthology.org/J93-2004.

William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. Effects of parameter norm growth during transformer training: Inductive bias from gradient descent. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. https://aclanthology.org/2021.emnlp-main.133.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. https://arxiv.org/abs/1809.02789.

MosaicML. Llm evaluation scores, 2023. https://www.mosaicml.com/llm-evaluation.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. Crosslingual generalization through multitask finetuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022. https://aclanthology.org/2023.acl-long.891.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro Von Werra, and Shayne Longpre. Octopack: Instruction tuning code large language models. *arXiv preprint*, 2023a. https://arxiv.org/abs/2308.07124.

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. In *Advances in Neural Information Processing Systems (NeuIPS)*, 2023b. https://arxiv.org/abs/2305.16264.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. *arXiv preprint*, 2024. https://arxiv.org/abs/2402.09906.

Erik Nijkamp, Tian Xie, Hiroaki Hayashi, Bo Pang, Congying Xia, Chen Xing, Jesse Vig, Semih Yavuz, Philippe Laban, Ben Krause, Senthil Purushwalkam, Tong Niu, Wojciech Kryscinski, Lidiya Murakhovs'ka, Prafulla Kumar Choubey, Alex Fabbri, Ye Liu, Rui Meng, Lifu Tu, Meghana Bhat, Chien-Sheng Wu, Silvio Savarese, Yingbo Zhou, Shafiq Rayhan Joty, and Caiming Xiong. Long sequence modeling with xgen: A 7b llm trained on 8k input sequence length. *arXiv preprint*, 2023. https://arxiv.org/abs/2309.03450.

OpenAI. Triton, 2021. https://github.com/openai/triton.

OpenAI. Gpt-4 technical report, 2023. https://arxiv.org/abs/2303.08774.

David Owen. How predictable is language model benchmark performance? *arXiv preprint*, 2024. https://arxiv.org/abs/2401.04757.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. http://www.aclweb.org/anthology/P16-1144.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. https://aclanthology.org/P02-1040.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. BBQ: A hand-built bias benchmark for question answering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022. https://aclanthology.org/2022.findings-acl.165.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1912.01703.

Patronus AI. EnterprisePII dataset, 2023. https://tinyurl.com/2r5x9bst.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint*, 2023. https://arxiv.org/abs/2306.01116.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. https://aclanthology.org/2023.findings-emnlp.936.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017. https://aclanthology.org/E17-2025.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Preprint*, 2019. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John F. J. Mellor, Irina Higgins, Antonia Creswell, Nathan McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, L. Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, N. K. Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Tobias Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew G. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem W. Ayoub, Jeff Stanway, L. L. Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint*, 2021. https://arxiv.org/abs/2112.11446.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. https://arxiv.org/abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint*, 2019. https://arxiv.org/abs/1910.10683.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. In *The Journal of Machine Learning Research (JMLR)*, 2020. https://arxiv.org/abs/1910.10683.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. https://aclanthology.org/D16-1264.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. In *Transactions of the Association for Computational Linguistics (TACL)*, 2019. https://aclanthology.org/Q19-1016.

Melissa Roemmele, Cosmin Adrian Bejan, , and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium*, 2011. https://people.ict.usc.edu/~gordon/copa.html.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/1909.12673.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. Gender bias in coreference resolution. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. https://aclanthology.org/N18-2002.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint*, 2019. https://arxiv.org/abs/1907.10641.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint*, 2019. http://arxiv.org/abs/1910.01108.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019. https://aclanthology.org/D19-1454.

Nikhil Sardana and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. In *NeurIPS Workshop on Efficient Natural Language and Speech Processing (ENLSP)*, 2023. https://arxiv.org/abs/2401.00448.

Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, et al. What language model to train if you have one million gpu hours? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. https://aclanthology.org/2022.findings-emnlp.54.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. https://arxiv.org/abs/2304.15004.

Utkarsh Sharma and Jared Kaplan. A neural scaling law from the dimension of the data manifold. In *Journal of Machine Learning Research (JMLR)*, 2022. https://arxiv.org/abs/2004.10802.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint*, 2020. https://arxiv.org/abs/2002.05202.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, et al. Aya dataset: An open-access collection for multilingual instruction tuning. *arXiv preprint arXiv:2402.06619*, 2024. https://arxiv.org/abs/2402.06619.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint*, 2024. https://arxiv.org/abs/2402.00159.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. https://openreview.net/forum?id=UmvSlP-PyV.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint*, 2021. https://arxiv.org/abs/2104.09864.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. https://aclanthology.org/N19-1421.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. In *International Conference on Learning Representations (ICLR)*, 2022. https://openreview.net/forum?id=f2OYVDyfIB.

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Tran, Dani Yogatama, and Donald Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. https://aclanthology.org/2023.findings-emnlp.825.

MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. www.mosaicml.com/blog/mpt-7b.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. Lamda: Language models for dialog applications. *arXiv preprint*, 2022. https://arxiv.org/abs/2201.08239.

Together Computer. Redpajama: an open dataset for training large language models, 2023. https://github.com/togethercomputer/RedPajama-Data.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*, 2023a. https://arxiv.org/abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint*, 2023b. https://arxiv.org/abs/2307.09288.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint*, 2024. https://arxiv.org/abs/2402.07827.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. https://arxiv.org/abs/1706.03762.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020. https://rdcu.be/b08Wh.

Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. From lsat: The progress and challenges of complex reasoning. *Transactions on Audio, Speech, and Language Processing*, 2021. https://arxiv.org/abs/2108.00648.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*, 2022a. https://openreview.net/forum?id=gEZrGCozdqR.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. In *Transactions on Machine Learning Research (TMLR)*, 2022b. https://openreview.net/forum?id=yzkSU5zdwD.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint*, 2021. https://arxiv.org/abs/2112.04359.

Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, et al. Small-scale proxies for large-scale transformer training instabilities. *arXiv preprint*, 2023. https://arxiv.org/abs/2309.14322.

Yizhe Xiong, Xiansheng Chen, Xin Ye, Hui Chen, Zijia Lin, Haoran Lian, Zhenpeng Su, Jianwei Niu, and Guiguang Ding. Temporal scaling law for large language models. *arXiv preprint*, 2024. https://arxiv.org/abs/2404.17785.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems (NeuIPS)*, 2021. https://arxiv.org/abs/2203.03466.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Feature learning in infinite depth neural networks. In *International Conference on Learning Representations (ICLR)*, 2024. https://openreview.net/forum?id=17pVDnpwwl.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. https://aclanthology.org/P19-1472.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. https://arxiv.org/abs/2106.04560.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeuIPS)*, 2019. https://arxiv.org/abs/1910.07467.

Biao Zhang, Ivan Titov, and Rico Sennrich. Improving deep transformer with depth-scaled initialization and merged attention. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019. https://aclanthology.org/D19-1083.

Yanli Zhao, Andrew Gu, Rohan Varma, Liangchen Luo, Chien chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel. In *Very Large Data Bases Conference (VLDB)*, 2023. https://dl.acm.org/doi/10.14778/3611540.3611569.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. Jec-qa: A legal-domain question answering dataset. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020. https://arxiv.org/abs/1911.12011.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint*, 2023. https://arxiv.org/abs/2304.06364.

Terry Yue Zhuo, Armel Zebaze, Nitchakarn Suppattarachai, Leandro von Werra, Harm de Vries, Qian Liu, and Niklas Muennighoff. Astraios: Parameter-efficient instruction tuning code large language models. *arXiv preprint*, 2024. https://arxiv.org/abs/2401.00788.