Does your data spark joy? Performance gains from domain upsampling at the end of training

Cody Blakeney [*1] **Mansheej Paul** [*1] **Brett W. Larsen** [*1] **Sean Owen** [1] **Jonathan Frankle** [1]

Abstract

Pretraining datasets for large language models (LLMs) have grown to trillions of tokens composed of large amounts of CommonCrawl (CC) web scrape along with smaller, domain-specific datasets. It is expensive to understand the impact of these domain-specific datasets on model capabilities as training at large FLOP scales is required to reveal significant changes to difficult and emergent benchmarks. Given the increasing cost of experimenting with pretraining data, how does one determine the optimal balance between the diversity in general web scrapes and the information density of domain specific data? In this work, we show how to leverage the smaller domain specific datasets by upsampling them relative to CC at the end of training to drive performance improvements on difficult benchmarks. This simple technique allows us to improve up to 6.90 pp on MMLU, 8.26 pp on GSM8K, and 6.17 pp on HumanEval relative to the base data mix for a 7B model trained for 1 trillion (T) tokens, thus rivaling Llama-2 (7B)—a model trained for twice as long. We experiment with ablating the duration of domain upsampling from 5% to 30% of training and find that 10% to 20% percent is optimal for navigating the tradeoff between general language modeling capabilities and targeted benchmarks. We also use domain upsampling to characterize at scale the utility of individual datasets for improving various benchmarks by removing them during this final phase of training. This tool opens up the ability to experiment with the impact of different pretraining datasets at scale, but at an order of magnitude lower cost compared to full pretraining runs.

[*]Equal contribution [1]Databricks Mosaic Research. Correspondence to: Cody Blakeney <cody.blakeney@databricks.com>.

ES-FoMo II: Workshop on Efficient Systems for Foundation Models, 41st International Conference on Machine Learning, Vienna, Austria. Copyright 2024 by the author(s).

1. Introduction

Pretraining datasets for large language models (LLMs), such as Dolma (Soldaini et al., 2023), have grown to trillions of tokens. To accommodate such large scales, they are typically composed of two types of data sources. First, they contain large amounts of web scraped data processed from CommonCrawl (CC) dumps. These are typically hundreds of billions to trillions of tokens in size and contain a diverse distribution of information. However, because of their size, they are necessarily less information dense and are not as filtered. Second, LLM pretraining mixes contain datasets that either target certain domains or come from single high quality sources. These are much smaller (often less than a hundred billion tokens tokens). They are also more carefully processed and are dense with information from domains we want LLMs to be good at; however, since their sources are limited, they are often less diverse (Computer, 2023).

Related Works: One of the biggest challenges to pretraining LLMs is determining the optimal strategy for mixing datasets that come from CC and smaller domain specific sources. Some previous works have opted to pretrain entirely on heavily processed CC data (Penedo et al., 2023). Others have used different heuristics to balance between CC and more domain specific datasets (Computer, 2023). At smaller scales, there have been attempts to algorithmically optimize the data mix proportions, but these methods have not been openly validated at the scale most modern language models are trained (Xie et al., 2024). Given the sheer cost of validating data mixing strategies at this scale, there is a paucity of open research on pretraining data for LLMs.

Ideally one would conduct data mix experiments at smaller scales to identify what is a good data mix. However, this is often ineffective because large FLOP scales are required to reveal significant changes in difficult and emergent benchmarks. In fact, most LLMs trained at smaller scales register random accuracy on many important benchmarks such as MMLU (Wei et al., 2022). As a result, experiments at smaller scales can often be misleading; the variation between different data mixes on important benchmarks is often due to noise rather then dataset quality at this scale. On the other hand, it is prohibitively expensive and impractical to exhaustively characterize datasets by doing multiple

training runs at the scale needed to measure above random performance on these metrics.

In this work, our goal is to characterize the utility of an alternative approach to conduct pretraining data experiments at a reasonable scale. Our strategy is to modify the data mixture *at the end of training* after we have already trained for enough FLOPs to measure meaningful signal on difficult benchmarks. We show that this is an effective strategy for improving LLM pretraining data mixes with experiments that are an order of magnitude cheaper than full training runs.

Contributions:

- We begin with a baseline mix of publicly available datasets that achieves the same scaling of performance with FLOPs as the Llama-2 model family for a 7B model trained for 1 trillion tokens (App. A).
- We introduce domain upsampling—a data intervention which upsamples domain specific datasets relative to Common Crawl at the end of training—and demonstrate that it can boost challenging metrics. In particular, we observe improvements of up to 6.90 pp on MMLU, 8.26 pp on GSM8K, and 6.17 pp on HumanEval relative to the base data mix in our training setup. This makes our performance comparable to Llama-2 (7B) but at approximately half the training FLOPs (Sec. 3.1).
- We ablate the percentage of training that utilizes domain upsampling and show 10%-20% is optimal for navigating the tradeoff between general language modeling capabilities and targeted benchmarks (App. B).
- We show how domain upsampling can be used as a FLOP-efficient tool to characterize how datasets impact model capabilities. By removing a subset of math-heavy pretraining data from the datasets we upsampled at the end of training, we quantified the impact these datasets have on specific benchmarks (Sec. 3.2).

2. Training Details

We studied domain upsampling on 7B parameter models trained for 1T tokens. This FLOP scale was chosen so the model performed above the noise floor on key metrics like MMLU, enabling us to see the effects of data interventions.

The 7B models trained for this work are decoder-only transformers using the MPT architecture in LLM Foundry (MosaicML et al., 2023). To evaluate our models we use the latest version of the Eval Gauntlet v0.3 (MosaicML et al., 2023), an evaluation framework consisting of 35 popular in context learning evaluation tasks used to evaluate LLM base models. The Gauntlet v0.3 aggregates scores on benchmarks across 6 categories (Appendix C). We use an inverse square root learning schedule similar to (Zhai et al., 2022).

Parameter	Value
Optimizer	LionW (Chen et al., 2024)
Learning Rate	0.00012
Betas	0.9, 0.95
Weight Decay	0.00012
Max Sequence Length	4096
Batch Size	960
Tokenizer	Tiktoken (GPT-4)
Positional Embedding	ALiBi (Press et al., 2022)

Table 1. Training Hyperparameters.

3. Results

Here we present the experiments demonstrating the performance boost achieved by domain upsampling as well as its utility in characterizing how datasets affect challenging, emergent metrics.

3.1. Domain upsampling significantly boosts performance on challenging metrics

For our baseline data mix, we grouped a set of publicly-available datasets into 4 broad categories (App. A):

- **Large-Scale Common Crawl:** Datasets derived from Common Crawl that emphasize scale. These datasets trade off thorough quality filtering in favor of curating a large and diverse set of tokens.
- **Small-Scale Common Crawl:** Datasets derived from Common Crawl with more extensive filtering but are smaller than large-scale Common Crawl.
- **Domain Specific data:** Small datasets that target certain domains or are from individual sources and are of high quality (e.g. Wikipedia).
- **Code:** Code data across a variety of programming languages.

Next, we introduce domain upsampling during the last 20% of training for our 1T token training run. For this, we start with a checkpoint at 0.8T tokens of training, change the mixing proportions of our pretraining data mix, and continue training for the remaining 0.2T tokens. The exact mixing proportions of our domain upsampled pretraining mix are in Table 2. These percentages were chosen based on the following heuristic: we hypothesize that though the Large-Scale CC adds a lot of diversity to the pretraining data mix, it is advantageous to emphasize Domain Specific data at the end of training to bias our model towards token distributions that have high information density in domains we care about. Thus, we remove Large-Scale Common Crawl from our data mix while upsampling both Domain Specific and Code subsets. We also maintain Small-Scale Common Crawl at high percentage to prevent a large distribution shift in our pretraining data.

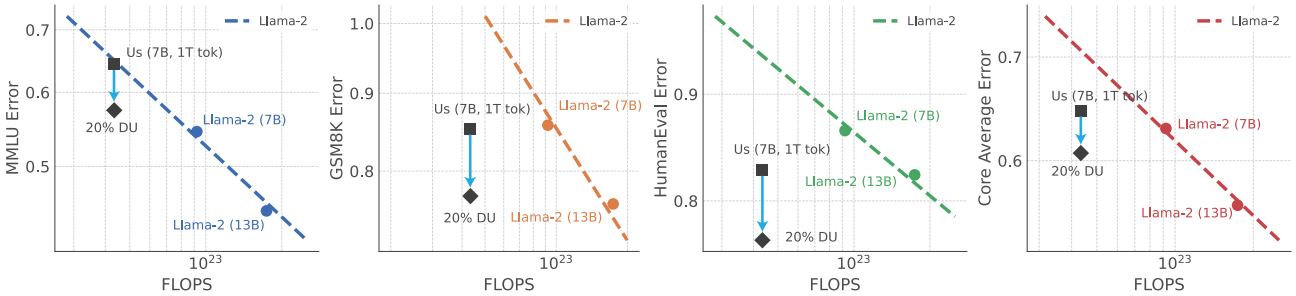

Figure 1. Domain upsampling (DU) using the proportions presented in Table 2 provides a significant performance boost on challenging metrics for no additional FLOP cost. The dashed lines represent the same scaling for the Llama-2 family of models as described in Figure 2. The square markers are the performance of our 7B model trained for 1T tokens with the data mix described in Appendix A; the diamond markers are the resulting models when domain upsampling is performed with the proportions specified in Table 2 for the final 20% or 200B tokens of training. The light blue arrow emphasizes the improvement we observe from DU: **6.90 pp** on MMLU, **8.26 pp** on GSM8K, **6.17 pp** on HumanEval, and **3.95 pp** on the Gauntlet v0.3

The results of this end-of-training data intervention are shown in Figure 1 and Table 3. Domain upsampling was incredibly effective in boosting model performance relative to the initial pretraining data mix on all challenging benchmarks. Given the large amount of code and math related data in the domain upsampled data mix, it is perhaps unsurprising that this intervention led to GSM8K and HumanEval scores that are approximately 10pp higher than Llama-2 (7B) despite the model being trained for half the total number of tokens. Additionally, this did not come at a cost to general language modeling capabilities; it led to an overall model performance improvement as measured by Gauntlet v0.3 Core Average. In fact, it improved world knowledge—as measured by MMLU and the Gauntlet v0.3 subset—relative to the base data mix, bringing us closer to Llama-2 (7B) performance on these metrics. There was only a small 1pp tradeoff in the Language Understanding subset.

Dataset Category	No DU	DU
Large-Scale Common Crawl	34.35%	0%
Small-Scale Common Crawl	36.70%	30%
Domain Specific	7.17%	35%
Code	21.78%	35%

Table 2. Domain upsampling (DU) is a data intervention in which datasets are removed from the data mix at the end of training in order to scale up or upsample the reamining data. We consider results for removing Large-Scale Common Crawl and scaling up the remaining datasets as specified in this table.

Overall, this across the board improvement on challenging benchmarks establishes the efficacy of domain upsampling as a pretraining data intervention for improving model performance. Importantly, even using simple heuristics for choosing the new data mix proportions has strong positive effects, leaving opportunity for further improvement with better tuned mixing proportions.

3.2. Domain upsampling is a FLOP-efficient tool to characterize how datasets impact model capabilities

Having observed that upsampling code and our domain specific datasets for a small percentage of training leads to significant improvements on difficult and emergent tasks, we explore the question: how does one attribute improvements to specific subsets of these data? Notably, as can be seen in Figure 3 and Table 7, GSM8K scores—a task measuring math and reasoning abilities—improves monotonically as duration of domain upsampling is increased. We hypothesize, given the quantity of math related data in our high-quality datasets that these may be responsible for some or all of this improvement. To quantify the impact of these datasets we repeat our experiment, applying domain upsampling for the last 10% of the training duration. We keep our dataset proportions identical to those in Table 2, but remove the math related subsets. We present the results in Table 4.

We observe that not only do the the mathematical knowledge and reasoning skills, as measured by MMLU (which contains STEM subsets) & GSM8k, *not* reach the same level of performance as the model trained using domain upsampling that included them, but in fact performance is worse then the baseline model with no domain upsampling. Moreover, every Gauntlet v0.3 subcategory score for the domain upsampling sans-math with the exception of programming is lower than the baseline model. From this we can draw the conclusion that these specific datasets are responsible for the majority of the mathematical knowledge and reasoning capabilities in both the base model and the domain upsampled variant.

With this observation we have successfully done something which generally would be considerably more expensive. That is, we have measured the impact of pretraining datasets at a scale where difficult and emergent tasks can be reli-

	Us (7B, 1T tok)		**Llama-2**	**OpenLlama**
Benchmark	No DU	20% DU	7B (2T tok)	7Bv2 (1T tok)
MMLU (5-shot)	35.69	42.59	**45.51**	40.38
GSM8K (8-shot)	14.71	**22.97**	14.25	7.05
HumanEval (pass@1)	17.23	**23.40**	13.55	15.20
Gauntlet v0.3				
Core Average	35.37	**39.32**	37.05	32.96
World Knowledge	41.77	44.19	**50.94**	43.79
Commonsense Reasoning	38.38	**42.59**	35.48	34.91
Language Understanding	61.52	60.08	**65.02**	61.00
Symbolic Problem Solving	16.28	20.23	**22.23**	19.09
Reading Comprehension	37.02	**45.45**	35.05	23.82
Programming (HE)	17.23	**23.40**	13.55	15.20

Table 3. Full evaluation reults for the models presented in Figure 1 along with a comparison to OpenLlama 7Bv2. Overall, our model with 20% domain upsampling outperforms Llama2 (7B) on the Gauntlet v0.3 despite being trained for 1T fewer tokens. Our model particularly excels at GSM8K and HumanEval but still trails Llama-2 (7B) on MMLU.

ably measured, but at an order of magnitude fewer training FLOPS. We believe application of domain upsampling opens up the ability for researchers to experiment with their pretraining datasets in a tractable way as compared to full pretraining runs.

		10% DU	
Benchmark	No DU	With Math	Sans Math
MMLU (5-shot)	35.69	**43.19**	29.71
GSM8K (8-shot)	14.71	20.47	11.37
HumanEval (pass@1)	17.23	20.39	**21.15**
Gauntlet v0.3			
Core Average	35.37	**38.46**	32.54
WK	41.77	**44.72**	39.08
CR	38.38	**42.33**	31.76
LU	**61.52**	60.41	59.97
SPS	16.28	**19.55**	16.80
RC	37.02	**43.35**	26.48
Programming	17.23	20.39	**21.15**

Table 4. Removing the math-specific datasets during domain upsampling results in significantly worse performance on all metrics except programming vs. performing domain upsampling with these datasets. Experiments such as this provide a significantly cheaper method to characterize datasets on challenging benchmarks like MMLU, GSM8K and HumanEval compared to full pretraining runs with different data mixes. The Gauntlet categories are abbreviated: World Knowledge (WK), Commonsense Reasoning (CR), Language Understanding (LU), Symbolic Problem Solving (SPS), and Reading Comprehension (RC).

4. Discussion

Pretraining LLMs has become an increasingly costly and clandestine endeavor given the scale of compute required for each experiment. This problem is exacerbated by the multi-faceted decision space presented to practitioners, especially in the selection of pretraining data. Since many important model capabilities emerge with scale, trying to explore this design space at small compute budgets is often ineffective: observations made about the effects of the pretraining data mix typically do not transfer to larger models or training budgets.

In this work, we take a crucial first step towards making experimentation with pretraining datasets cheaper. We introduce domain upsampling, a method that can strongly impact the performance of the model by making targeted changes to the data mix at the end of training. This enables us to achieve the performance of Llama-2 (7B) but with half the training budget. By varying the duration of domain upsampling, we demonstrate how to navigate the tradeoff between targeting specific domains and making general purpose language models.

Finally, we show how making changes to the data mix only during the domain upsampling period enabled us to cheaply characterize the impact of several math-focused datasets, and we see many opportunities to use this method as a general tool for studying pretraining data in a FLOP-efficient manner. It also creates a platform to test data interventions at scale: instead of testing possible dataset optimization algorithms at small scales and hoping they will generalize, we can test them at the end of training to effectively measure their impact at scale. Overall this brings down the cost of experimentation, making pretraining data experiments more accessible.

References

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Pham, H., Dong, X., Luong, T., Hsieh, C.-J., Lu, Y., et al. Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems*, 36, 2024.

Clark, C., Lee, K., Chang, M., 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.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1300. URL https://doi.org/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 preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Geng, X. and Liu, H. Openllama: An open reproduction of llama, May 2023. URL https://github.com/openlm-research/open_llama.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Barzilay, R. and Kan, M. (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1147. URL https://doi.org/10.18653/v1/P17-1147.

Levesque, H., Davis, E., and Morgenstern, L. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

MosaicML et al. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. *URL www.mosaicml. com/blog/mpt-7b. Accessed*, pp. 05–05, 2023.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1144. URL https://doi.org/10.18653/v1/p16-1144.

Patel, A., Bhattamishra, S., and Goyal, N. Are NLP models really able to solve simple math word problems? In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 2080–2094. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.168. URL https://doi.org/10.18653/v1/2021.naacl-main.168.

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., 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 preprint arXiv:2306.01116*, 2023.

Press, O., Smith, N., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100, 000+ questions for machine comprehension of text. In Su, J., Carreras, X., and Duh, K. (eds.), *Proceedings*

of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pp. 2383–2392. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/D16-1264. URL https://doi.org/10.18653/v1/d16-1264.

Reddy, S., Chen, D., and Manning, C. D. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266, 2019. doi: 10.1162/TACL_A_00266. URL https://doi.org/10.1162/tacl_a_00266.

Roemmele, M., Bejan, C. A., and Gordon, A. S. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sap, M., Rashkin, H., Chen, D., LeBras, R., and Choi, Y. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

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., 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, E. P., Hajishirzi, H., Smith, N. A., Zettlemoyer, L., Beltagy, I., Groeneveld, D., Dodge, J., and Lo, K. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2023.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Wolfe, T., Tunstall, L., and von Platen, P. Jeopardy dataset on hugging face hub. https://huggingface.co/datasets/jeopardy, 2022.

Xie, S. M., Pham, H., Dong, X., Du, N., Liu, H., Lu, Y., Liang, P. S., Le, Q. V., Ma, T., and Yu, A. W. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2024.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12104–12113, 2022.

Zhong, W., Cui, R., Guo, Y., Liang, Y., Lu, S., Wang, Y., Saied, A., Chen, W., and Duan, N. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

A. Baseline data mix achieves Llama-2 scaling

To construct a baseline data mix, we grouped a set of publicly-available datasets into 4 broad categories as described in the main text: Large-Scale Common Crawl, Small-Scale Common Crawl, Domain Specific, and Code. We set the proportions for mixing these datasets based on a rough heuristic for the number of epochs each of these groups would be seen during the 1 trillion token training duration. Specifically, we choose 0.5 epochs for the Small-Scale Common Crawl and Domain Specific data and 1 epoch for Code. The remainder of the 1 trillion tokens are filled with Large-Scale Common Crawl. The exact proportions are in Table 5.

The rationale behind choosing these proportions is as follows: we expect the Small-Scale Common Crawl and Domain Specific data to be of high quality and we wanted them to be well represented on our 1 trillion token budget. Also, we wanted to emphasize coding ability and so we decided to sample code data at a high percentage—initial experiments indicated that a high percentage of code around 20% boosted programming and reasoning ability without negatively impacting language abilities. We then treat the Large-Scale CC as filler tokens that increase the diversity of our dataset and allow us to fill our token budget.

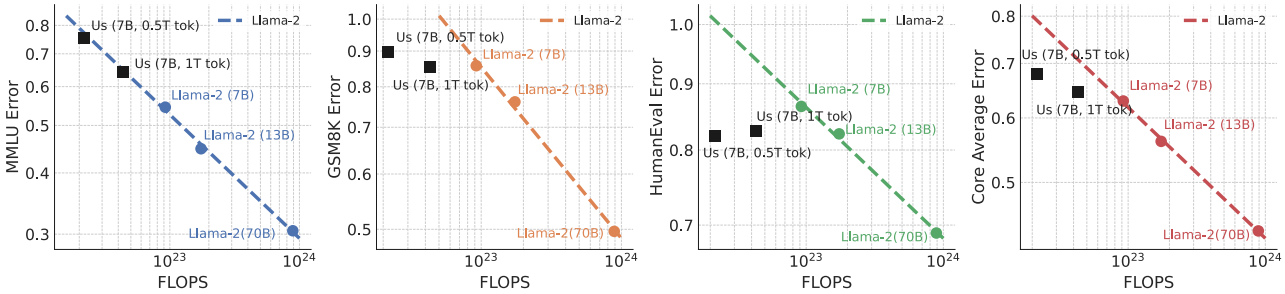

Figure 2. On key benchmarks, our 7B models trained with the data mix presented in Table 5 have errors *at or below* the error vs. FLOP scaling line of the Llama-2 family of models. We first evaluated the performance of the 7B, 13B, and 70B variants of the Llama-2 models on MMLU, GSM8K, HumanEval, and the Gauntlet v0.3 Core Average. We then performed linear regression on the log of the error on these metrics vs. the log of the FLOPS used to train these models. This scaling relationship is plotted as a dashed line in the log-log plots shown above; one can observe that the models in the Llama-2 family lie close to this scaling line. For MMLU, our models (square markers) lie on the Llama-2 scaling line. For the other metrics, our models are significantly below the scaling line.

Importantly, since the goal of our experimental setup is to demonstrate the utility of domain upsampling at the end of training (discussed in Section 3.1), we opt for choosing a reasonable heuristic for picking our initial data mix proportions without too much optimization. Table 6 and Figure 2 show the performance of this initial pretraining data mix for two 7B models trained for 0.5T and 1T tokens. This heuristic has indeed been validated by our empirical results; plotting error vs. FLOPs shows that our models lie on or below the Llama-2 scaling line on the Gauntlet v0.3 Core Average, MMLU, GSM8K, and HumanEval. Interestingly, though the overall performance scaling (as measured by Gaunlet v0.3 Core Average) is very similar, our particular data choices and mixing coefficients have led to slightly different tradeoffs. The model trained for 1T tokens outperforms the Llama-2 7B model trained for 2T tokens on GSM8K and HumanEval. This indicates that our models have better mathematical and programming ability despite being trained for half the number of tokens. We also provide a comparison to OpenLlama 7Bv2 (Geng & Liu, 2023), a 7B model that provides some open details about their data mix.

Dataset Category	Percentage	Tokens	Epochs (1T)
Large-Scale Common Crawl	34.35%	343.5B	0.148
Small-Scale Common Crawl	36.70%	367.0B	0.5
Domain Specific	7.17%	71.7B	0.5
Code	21.78%	217.8B	1

Table 5. Proportions for our pretraining data mix in terms of the 4 dataset groups. Code data was included at twice the proportion of other domain specific datasets to focus on boosting coding capabilities of our models. Large-scale Common Crawl was used to fill the remainder of the tokens once the other proportions were chosen.

	Us (7B)		Llama-2	OpenLlama
Benchmark	0.5T tok	1T tok	7B (2T tok)	7Bv2 (1T tok)
MMLU (5-shot)	24.70	35.69	**45.51**	40.38
GSM8K (8-shot)	10.16	**14.71**	14.25	7.05
HumanEval (pass@1)	**18.02**	17.23	13.55	15.20
Gauntlet v0.3				
Core Average	32.13	35.37	**37.05**	32.96
World Knowledge	39.29	41.77	**50.94**	43.79
Commonsense Reasoning	30.52	**38.38**	35.48	34.91
Language Understanding	61.47	61.52	**65.02**	61.00
Symbolic Problem Solving	14.10	16.28	**22.23**	19.09
Reading Comprehension	29.36	**37.02**	35.05	23.82
Programming (HE)	**18.02**	17.23	13.55	15.20

Table 6. Full evaluation results for the models presented in Figure 2. We note that a 7B model trained with our data mix for 1T tokens outperforms Llama2-7B—a model trained for 2T tokens—on GSM8K, HumanEval, and the Commonsense Reasoning and Reading Comprehension subsets of the Gauntlet v0.3. We also compare to OpenLlama 7Bv2, a similar model with a publicly available data mix trained for 1T tokens. Note that HumanEval (HE) is the sole component of the programming section of the Gauntlet.

B. Changing the duration of domain upsampling enables us to navigate the trade-off between targeting specific domains and general purpose language models

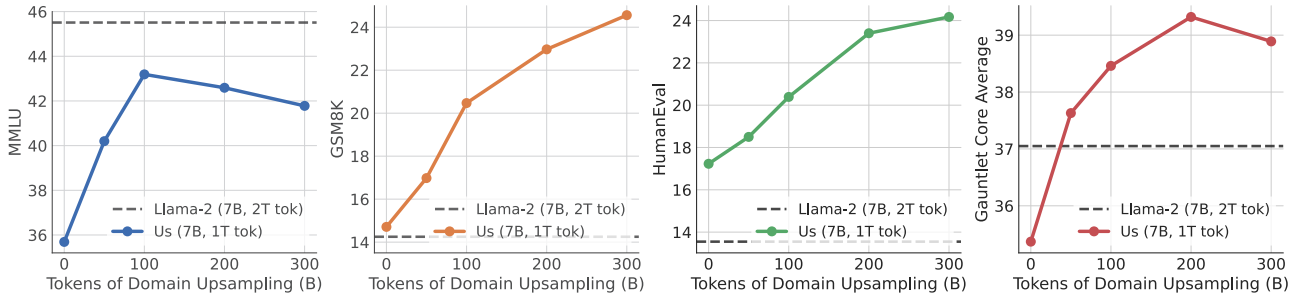

Figure 3. Ablating the duration of Domain Upsampling (DU). Here we consider performing DU for the final 5%, 10%, 20%, and 30% (50B, 100B, 200B, and 300B tokens) of training for a 7B model for a total duration 1T tokens. We observe that GSM8K and HumanEval performance continue to improve with increased DU while MMLU and the Gauntlet Core Average peak at 10% and 20% respectively. Looking across the metrics presented in Table 7, we conclude that (1) DU for the final 10%-20% of training provides the best trade-off and (2) the mix used for DU should not be used for the entire duration of training.

The success of domain upsampling for the last 20% of training raises the question: are the improvements from an end-of-training data intervention or are they from overall better data mix proportions? Phrased another way, are the data mix proportions in Table 2 better than our initial data mix and would training a model for 1T tokens with this data mix lead to better performance? In this section, we provide evidence that this is not the case and in fact, treating domain upsampling as an end-of-training data intervention helps us better tradeoff domain specific improvements and general language modeling capabilities.

To identify when in training this intervention should be applied, we ablate our previous experiment by performing domain upsampling for the last 5%, 10%, 20%, and 30% of training. The results of this experiment are shown in Figure 3 and Table 7. Note, while the math and programming related benchmarks, such as HumanEval, GSM8K and related Gauntlet v0.3 subscores, continue to improve as we increase the fraction of training that uses domain upsampling, other benchmarks reach optimal performance at 20% or less. For example, MMLU peaks at 10% and Gauntlet v0.3 Core Average peaks at 20%. Thus, as we increase the fraction of training with domain upsampling beyond 20%, improvements on math and coding benchmarks come at the cost of performance on general language modeling abilities.

	Us (7B, 1T tok)				
Benchmark	0%	5%	10%	20%	30%
MMLU (5-shot)	35.69	40.20	**43.19**	42.59	41.78
GSM8K (8-shot)	14.71	16.98	20.47	22.97	**24.56**
HumanEval (pass@1)	17.23	18.50	20.39	23.40	**24.17**
Gauntlet v0.3					
Core Average	35.37	37.63	38.46	**39.32**	38.89
World Knowledge	41.77	43.52	**44.72**	44.19	43.71
Commonsense Reasoning	38.38	42.97	42.33	**42.59**	42.19
Language Understanding	**61.52**	61.05	60.41	60.08	60.35
Symbolic Problem Solving	16.28	18.50	19.55	20.23	**20.44**
Reading Comprehension	37.02	41.23	43.35	**45.45**	42.50
Programming (HE)	17.23	18.50	20.39	23.40	**24.17**

Table 7. Full evaluation results for the models presented in Figure 3.

This apparent trade-off indicates that the domain upsampling data mix proportions are not incontrovertibly better than the initial data mix, and training with it for the full 1T token duration would not lead to a better general purpose language model. We do not rule out that there is an alternate mix that achieves similar performance as the 20% domain upsampling experiment when trained for the full training duration. However, finding such a mix is expensive to iterate on for the full training run. Thus, the strength of domain upsampling is that it gives us a tool to navigate this tradeoff between targeted domains and general language modeling abilities with experiments that are an order of magnitude cheaper.

C. Gauntlet v0.3

The Gauntlet v0.3 is a aggregation of benchmark developed by Mosaic Research. Rather than reporting a monolithic metric in which all scores are aggregated together, the individual benchmarks were grouped into six broad competencies corresponding to different capabilities we want our LLMs to have:

1. **World Knowledge:** Measures the model's factual knowledge across a range of subjects.
2. **Commonsense Reasoning:** Evaluates the model's ability to do basic reasoning tasks that require commonsense knowledge of objects, their properties, and their behaviors.
3. **Language Understanding:** Assesses the model's ability to understand structure and properties of language.
4. **Symbolic Problem Solving:** Tests the model's ability to solve a diverse range of symbolic tasks including arithmetic, logical reasoning, algorithms, and algebra.
5. **Reading Comprehension:** Measures a model's ability to answer questions based on information in a passage of text.
6. **Programming:** Quantifies the ability to generate code from docstring descriptions.

These divisions allow for more fine-grained comparison between models and is especially useful for understanding how datasets affect different capabilities of the model. The random baseline of each metric was subtracted out before aggregating. For example, if the metric is 4-option multiple choice questions giving a random baseline of 25% and the model achieves 30% then this would be aggregated as $(0.3 - 0.25)/(1 - 0.25) = 0.0667$, essentially rescaling accuracy above change to be between 0 and 1. If the random baseline is approximately 0, then the metric is reported as is. Table 8 list the benchmarks in each category.

Benchmark	Citation
World Knowledge	
Jeopardy (3-shot)	(Wolfe et al., 2022)
MMLU (5-shot)	(Hendrycks et al., 2020)
BIG-bench Wikidata (3-shot)	(Srivastava et al., 2022)
ARC-easy (3-shot)	(Clark et al., 2018)
ARC-challenge (3-shot)	(Clark et al., 2018)
TriviaQA-Subsampled (3-shot)	(Joshi et al., 2017)
Commonsense Reasoning	
BIG-bench Strategy QA	(Srivastava et al., 2022)
BIG-bench Strange Stories	(Srivastava et al., 2022)
COPA (0-shot)	(Roemmele et al., 2011)
PIQA (10-shot)	(Bisk et al., 2020)
SIQA (3-shot)	(Sap et al., 2019)
Openbook QA (10-shot)	(Mihaylov et al., 2018)
Commonsense QA (0-shot)	(Talmor et al., 2018)
Language Understanding	
LAMBADA	(Paperno et al., 2016)
HellaSwag	(Zellers et al., 2019)
Winograd (3-shot)	(Levesque et al., 2012)
Winogrande (5-shot)	(Sakaguchi et al., 2021)
Symbolic Problem Solving	
BIG-bench Elementary Math QA (1-shot)	(Srivastava et al., 2022)
BIG-bench Dyck Languages (5-shot)	(Srivastava et al., 2022)
BIG-bench Operators (3-shot)	(Srivastava et al., 2022)
Simple Arithmetic (with spaces, 5-shot)	(MosaicML et al., 2023)
Simple Arithmetic (no spaces, 5-shot)	(MosaicML et al., 2023)
GSM8K (8-shot)	(Cobbe et al., 2021)
SVAMP (5-shot)	(Patel et al., 2021)
AGI Eval LSAT AR (5-shot)	(Zhong et al., 2023)
Reading Comprehension	
SQuAD (3-shot)	(Rajpurkar et al., 2016)
BoolQ	(Clark et al., 2019)
CoQA	(Reddy et al., 2019)
AGI Eval LSAT RC (5-shot)	(Zhong et al., 2023)
AGI Eval LSAT LR (5-shot)	(Zhong et al., 2023)
AGI Eval SAT En (5-shot)	(Zhong et al., 2023)
Programming	
HumanEval (pass@1)	(Chen et al., 2021)

Table 8. Metrics included in the Gauntlet v0.3. Evaluation metrics are 0-shot unless otherwise denoted.