OLMo: Accelerating the Science of Language Models

Dirk Groeneveld[α] **Iz Beltagy**[α]

Pete Walsh[α] **Akshita Bhagia**[α] **Rodney Kinney**[α] **Oyvind Tafjord**[α]

Ananya Harsh Jha[α] **Hamish Ivison**[αβ] **Ian Magnusson**[α] **Yizhong Wang**[αβ]

Shane Arora[α] **David Atkinson**[α] **Russell Authur**[α] **Khyathi Raghavi Chandu**[α]
Arman Cohan[γα] **Jennifer Dumas**[α] **Yanai Elazar**[αβ] **Yuling Gu**[α]
Jack Hessel[α] **Tushar Khot**[α] **William Merrill**[δ] **Jacob Morrison**[α]
Niklas Muennighoff **Aakanksha Naik**[α] **Crystal Nam**[α] **Matthew E. Peters**[α]
Valentina Pyatkin[αβ] **Abhilasha Ravichander**[α] **Dustin Schwenk**[α] **Saurabh Shah**[α]
Will Smith[α] **Emma Strubell**[αμ] **Nishant Subramani**[α] **Mitchell Wortsman**[β]

Pradeep Dasigi[α] **Nathan Lambert**[α] **Kyle Richardson**[α]
Luke Zettlemoyer[β] **Jesse Dodge**[α] **Kyle Lo**[α] **Luca Soldaini**[α]

Noah A. Smith[αβ] **Hannaneh Hajishirzi**[αβ]

[α]Allen Institute for Artificial Intelligence
[β]University of Washington [γ]Yale University
[δ]New York University [μ]Carnegie Mellon University

olmo@allenai.org

Abstract

Language models (LMs) have become ubiquitous in both NLP research and in commercial product offerings. As their commercial importance has surged, the most powerful models have become closed off, gated behind proprietary interfaces, with important details of their training data, architectures, and development undisclosed. Given the importance of these details in scientifically studying these models, including their biases and potential risks, we believe it is essential for the research community to have access to powerful, truly open LMs. To this end, this technical report details the first release of OLMo, a state-of-the-art, truly **O**pen **L**anguage **Mo**del and its framework to build and study the science of language modeling. Unlike most prior efforts that have only released model weights and inference code, we release OLMo and the whole framework, including training data and training and evaluation code. We hope this release will empower and strengthen the open research community and inspire a new wave of innovation.

Weights	https://huggingface.co/allenai/OLMo-7B
Code	https://github.com/allenai/OLMo
Data	https://huggingface.co/datasets/allenai/dolma
Evaluation	https://github.com/allenai/OLMo-Eval
Adaptation	https://github.com/allenai/open-instruct

1 Introduction

Language models have been at the center of NLP technologies for many years (Rosenfeld, 2000; Bengio et al., 2003; Mikolov et al., 2013; Peters et al., 2018; Brown et al., 2020). Recently, due to large-scale pretraining and human annotation for alignment, they have become commercially valuable (OpenAI, 2023). However, as their commercial value has increased, the largest models have become gated behind proprietary interfaces, with important details left undisclosed.

We believe that full access to open language models for the research community is critical to the scientific study of these models, their strengths and weaknesses, and their biases and risks. Accordingly, we introduce **OLMo**, a state-of-the-art, truly open language model and framework to build, study, and advance LMs, along with the training data, training and evaluation code, intermediate model checkpoints, and training logs.

Recent LM releases have varied in their degree of openness. For example, Mistral 8x7B provided model weights and a brief report (Jiang et al., 2024), while LLaMA came with in-depth adaptation training instructions (Touvron et al., 2023b), and Mosaic Pretrained Transformer came with many details, including the dataset distribution, though not the data itself (MosaicML NLP Team, 2023). Falcon's pretraining data was partially released (Almazrouei et al., 2023), and the most open models—the Pythia suite (Biderman et al., 2023) and BLOOM (BigScience et al., 2022)—released training code, model checkpoints, training data and more.

OLMo releases the whole framework from data to training to evaluation tools: multiple training checkpoints across multiple hardware types, training logs, and exact datasets used, with a permissive license. We are not the only team to do this; recent work from LLM360 targets similar goals (Liu et al., 2023). OLMo narrows the gap from their models to state-of-the-art capabilities of models like LLaMA2. This project has benefited from lessons learned from all of these previous efforts with their varying degrees of openness, and we believe that a large, diverse population of open models is the best hope for scientific progress on understanding language models and engineering progress on improving their utility.

The OLMo framework encompasses the tools and resources required for building and researching language models. For training and modeling, it includes full model weights, training code, training logs, ablations, training metrics in the form of Weights & Biases logs, and inference code. This first release includes four variants of our language model at the 7B scale corresponding to different architectures, optimizers, and training hardware, and one model at the 1B scale, all trained on at least 2T tokens. We are also releasing hundreds of intermediate checkpoints available as revisions on HuggingFace. For dataset building and analysis, it includes the full training data used for these models, including code that produces the training data, from AI2's Dolma (Soldaini et al., 2024), and WIMBD (Elazar et al., 2023) for analyzing pretraining data. For evaluation, it includes AI2's Catwalk (Groeneveld et al., 2023) for downstream evaluation and Paloma (Magnusson et al., 2023) for perplexity-based evaluation. For instruction-tuning, we released Open Instruct (Ivison et al., 2023; Wang et al., 2023), and we are currently using it to produce an adapted (instruction-tuned and RLHFed) version of OLMo, which we will release soon. Finally, all code and weights are released under the Apache 2.0 License.[1]

This is the first step in a long series of planned releases, continuing with larger models, instruction-tuned models, and more modalities and variants down the line. We therefore hope to catalyze research into as-yet poorly understood aspects of these models, for example, the relationship between pretraining data and model capabilities, the impact of design and hyperparameter choices, and various optimization methods and their impact on model training. In addition, we report on the lessons learned and important details necessary to successfully train language models at this scale.

2 OLMo Framework

This section describes the OLMo framework, consisting of the OLMo models (Section 2.1), our pre-training dataset, Dolma (Section 2.2), and our evaluation framework (Section 2.3).

[1] http://www.apache.org/licenses/LICENSE-2.0

2.1 OLMo Model and Architecture

We adopt a decoder-only transformer architecture based on Vaswani et al. (2017), and deliver 1B and 7B variants as described in Table 1, with a 65B version coming soon. Our specific architecture includes several improvements over the vanilla transformer from Vaswani et al. (2017) following other recent large language models like PaLM (Chowdhery et al., 2022), the LLaMA family (Touvron et al., 2023a,b), OpenLM (Gururangan et al., 2023), and Falcon (Almazrouei et al., 2023). Table 2 gives a comprehensive comparison of our 7B architecture to the similarly-sized models from these other families.

Size	Layers	Hidden Size	Attention Heads	Tokens Trained
1B	16	2048	16	2T
7B	32	4086	32	2.46T
65B*	80	8192	64	

Table 1: OLMo model sizes and the maximum number of tokens trained to.
* *At the time of writing our 65B model is still training.*

We generally select hyperparameters by optimizing for training throughput on our hardware while minimizing the risk of loss spikes and slow divergence. We ablate choices through our in-loop evaluation setting, given available computational sources (Section 2.3). Table 2 compares our design choices with recent state-of-the-art open language models. Our main changes over the vanilla transformer architecture can be summarized as follows:

1. **No biases.** Following LLaMA, PaLM, and others, we exclude all bias terms from our architecture in order to improve training stability.
2. **Non-parametric layer norm.** We use the non-parametric formulation of layer norm (Ba et al., 2016) in which there is no affine transformation within the norm, i.e. no "adaptive gain" (or bias). We believe this was the safest option and it was also the fastest compared to the other variants we considered: parametric layer norm and RMSNorm (Zhang and Sennrich, 2019).
3. **SwiGLU activation function.** Like LLaMA, PaLM, and others we use the SwiGLU activation function (Shazeer, 2020) instead of ReLU, and following LLaMA the activation hidden size is approximately $\frac{8}{3}d$, but increased to the closest multiple of 128 (e.g. 11,008 for our 7B model) to improve throughput.[2]
4. **Rotary positional embeddings (RoPE).** Like LLaMA, PaLM, and others we replace absolute positional embeddings with rotary positional embeddings (RoPE; Su et al., 2021).
5. **Vocabulary.** We use a modified version of the BPE-based tokenizer from GPT-NeoX-20B (Black et al., 2022) with additional tokens for masking personal identifiable information (PII). The final vocabulary size is 50,280. However, to maximize training throughput we increase the size of the corresponding embedding matrix in our model to 50,304 so that it's a multiple of 128.

2.2 Pretraining Data: Dolma

Despite progress in access to model parameters, pretraining datasets are still not as open. Pretraining data are often not released alongside open models (let alone closed models) and documentation about such data is often lacking in detail that would be needed to reproduce or fully understand the work. This has made it difficult to support certain threads of language model research, such as understanding how training data impacts model capabilities and limitations. To facilitate open research on language model pretraining, we built and released our pretraining dataset, Dolma—a diverse, multi-source corpus of 3T tokens across 5B documents acquired from 7 different data

[2] Since SwiGLU is a "gated" activation function, the output is half the size of the input. So technically our inputs to SwiGLU have a dimensionality of $2 \times 11{,}008 = 22{,}016$ for our 7B model.

	OLMo-7B	LLaMA2-7B	OpenLM-7B	Falcon-7B	PaLM-8B
Dimension	4096	4096	4096	4544	4096
Num heads	32	32	32	71	16
Num layers	32	32	32	32	32
MLP ratio	~8/3	~8/3	~8/3	4	4
Layer norm type	non-parametric	RMSNorm	parametric	parametric	parametric
Positional embeddings	RoPE	RoPE	RoPE	RoPE	RoPE
Attention variant	full	GQA	full	MQA	MQA
Biases	none	none	in LN only	in LN only	none
Block type	sequential	sequential	sequential	parallel	parallel
Activation	SwiGLU	SwiGLU	SwiGLU	GeLU	SwiGLU
Sequence length	2048	4096	2048	2048	2048
Batch size (instances)	2160	1024	2048	2304	512
Batch size (tokens)	~4M	~4M	~4M	~4M	~1M
Weight tying	no	no	no	no	yes

Table 2: LM architecture comparison at the 7–8B scale. In the "layer norm type" row, "parametric" and "non-parametric" refer to the usual layer norm implementation with and without adaptive gain and bias, respectively.

Source	Doc Type	UTF-8 bytes *(GB)*	Documents *(millions)*	GPT-NeoX tokens *(billions)*
Common Crawl	web pages	9,022	3,370	2,006
The Stack	code	1,043	210	342
C4	web pages	790	364	174
Reddit	social media	339	377	80
peS2o	STEM papers	268	38.8	57
Project Gutenberg	books	20.4	0.056	5.2
Wikipedia, Wikibooks	encyclopedic	16.2	6.2	3.7
Total		**11,519**	**4,367**	**2,668**

Table 3: Composition of Dolma.

sources that are (1) commonly seen in large-scale language model pretraining and (2) accessible to the general public (Soldaini et al., 2024). Table 3 provides a high-level overview of the amount of data from each source.

Dolma is built using a pipeline of (1) language filtering, (2) quality filtering, (3) content filtering, (4) deduplication, (5) multi-source mixing, and (6) tokenization. We refer the reader to the Dolma report (Soldaini et al., 2024) for more details about its design principles, details about its construction, and a more detailed summary of its contents. The report provides additional analyses and experimental results from training language models on intermediate states of Dolma to share what we learned about important data curation practices, including the role of content or quality filters, deduplication, and mixing data from multiple sources. We keep documents from each source separate, both during curation as well as in the final release. We open-sourced our high-performance data curation tools; this toolkit can be used to further experiment on Dolma, reproduce our work, and enable fast and easy curation of pretraining corpora. Finally, we also open-sourced our WIMBD tool (Elazar et al., 2023) to help with dataset analysis.

2.3 Evaluation

We perform model evaluation at two stages: *online* evaluation to make decisions for model design and *offline* evaluation to evaluate model checkpoints. For offline evaluation, we use the Catwalk framework (Groeneveld et al., 2023), our publicly available evaluation tool with access to a wide range of datasets and task formats. Using Catwalk, we perform downstream evaluation as well as intrinsic language modeling evaluation on our new perplexity benchmark, Paloma (Magnusson et al., 2023).

For both downstream and perplexity evaluation, we use our fixed evaluation pipeline to compare results against several publicly available models.

In-Loop Training Ablations Throughout model training, we perform downstream evaluations to make decisions around model architecture, initialization, optimizers, learning rate schedule, and data mixtures. We call this our *online* evaluation as it runs in-loop every 1000 training steps (or ~4B training tokens) and provides an early and continuous signal on the quality of the model being trained. These evaluations rely on many of the core tasks and experiment settings used for our *offline* evaluation detailed in Section 4.1, which also mirrors the task and evaluation structure of the EleutherAI eval harness (Gao et al., 2023).

Downstream Evaluation Following much previous work (Brown et al., 2020; Black et al., 2022; Touvron et al., 2023a,b, *inter alia*), we report zero-shot performance on a set of downstream tasks. Our evaluation suite consists of 9 core tasks corresponding closely to the commonsense reasoning task set reported by Touvron et al. (2023a) and Touvron et al. (2023b) (see Table 6 for a list of tasks). Given the scale of the models being evaluated, such tasks were selected at the beginning of model development due to their naturalness (e.g., all can formulated as text completion scoring tasks) and ability to provide meaningful signals throughout training (see Figure 1).

Intrinsic Language Modeling Evaluation To measure how OLMo-7B fits distributions of language beyond held-out training data, we use Paloma (Magnusson et al., 2023), a new perplexity benchmark that includes 585 different domains of text. Domains range from nytimes.com to r/depression on Reddit and are drawn from 18 separate data sources, such as C4 (Raffel et al., 2020), in stratified samples. This allows for more equal inclusion of text domains that are under-represented in their source corpora.

We aim not just to compare OLMo-7B against other models for best performance, but also to demonstrate how it enables fuller and more controlled scientific evaluations. OLMo-7B is the largest LM with explicit decontamination for perplexity evaluation. Following the approach described in Paloma, we remove any pretraining document with paragraphs leaked from Paloma evaluation data. Without decontamination, other models risk underestimating perplexity (i.e., overestimating the model's out-of-sample fit). We also release intermediate checkpoints, allowing richer comparisons with two other models that release checkpoints, Pythia-6.9B (Biderman et al., 2023) and RPJ-INCITE-7B (Together Computer, 2023) (see Figure 2).

3 Training OLMo

This section describes our pretraining setup, including our distributed training framework (Section 3.1), optimizer settings (Section 3.2), data preparation (Section 3.3), and hardware (Section 3.4).

3.1 Distributed Training Framework

We train our models using the *ZeRO* optimizer strategy (Rajbhandari et al., 2019) via PyTorch's `FSDP` framework (Zhao et al., 2023), which reduces memory consumption by sharding the model weights and their corresponding optimizer state across GPUs. At the 7B scale, this enables training with a micro-batch size of 4096 tokens per GPU on our hardware (see Section 3.4). For OLMo-1B and -7B models, we use a constant global batch size of approximately 4M tokens (2048 instances, each with a sequence length of 2048 tokens). For OLMo-65B model (currently training), we use a batch size warmup that starts at approximately 2M tokens (1024 instances), then doubles every 100B tokens until reaching approximately 16M tokens (8192 instances).

Size	Peak LR	Betas	Epsilon	Weight Decay	Batch Size (tokens)
1B	`4.0E-4`	(0.9, 0.95)	`1.0E-5`	0.1	~4M
7B	`3.0E-4`	(0.9, 0.95)	`1.0E-5`	0.1	~4M
65B*	`1.5E-4`	(0.9, 0.95)	`1.0E-5`	0.1	~2M → ~4M → ~8M → ~16M

Table 4: AdamW pretraining hyperparameters for OLMo models.
* *At the time of writing our 65B model is still training.*

To improve throughput, we employ mixed-precision training (Micikevicius et al., 2017) through `FSDP`'s built-in settings and PyTorch's `amp` module. The latter ensures that certain operations like the softmax always run in full precision to improve stability, while all other operations run in half-precision with the `bfloat16` format. Under our specific settings, the sharded model weights and optimizer state local to each GPU are kept in full precision. The weights within each transformer block are only cast to `bfloat16` when the full-sized parameters are materialized on each GPU during the forward and backward passes. Gradients are reduced across GPUs in full precision.

3.2 Optimizer

We use the AdamW optimizer (Loshchilov and Hutter, 2019) with the hyperparameters shown in Table 4. For all model sizes, we warm up the learning rate over 5000 steps (~21B tokens) and then decay it linearly from there down to a tenth of the peak learning rate over the remainder of training. After the warm-up period, we clip gradients such that the total l^2-norm of the parameter gradients[3] does not exceed 1.0. Table 5 gives a comparison of our optimizer settings at the 7B scale to those of other recent LMs that also used AdamW.

3.3 Data

We built our training dataset out of a 2T-token sample from our open dataset, Dolma (Soldaini et al., 2024), which we describe in Section 2.2. The tokens from every document are concatenated together after appending a special `EOS` token to the end of each document, and then we group consecutive chunks of 2048 tokens to form training instances. The training instances are shuffled in the exact same way for each training run. The data order and exact composition of each training batch can be reconstructed from the artifacts we release.

All of our released models have been trained to at least 2T tokens (a single epoch over our training data), and some have been trained beyond that by starting a second epoch over the data with a different shuffling order. The impact of repeating this small amount of data should be negligible according to prior work (Muennighoff et al., 2023).

3.4 Hardware

In order to verify that our codebase could be used on both NVIDIA and AMD GPUs without any loss in performance, we trained models on two different clusters:

- **LUMI:** Provided by the LUMI supercomputer,[4] we used up to 256 nodes on this cluster, where each node consists of 4x AMD MI250X GPUs with 128GB of memory[5] and 800Gbps of interconnect.

[3] During gradient clipping all of the model's parameters are treated as a single big vector (as if all parameters were flattened and concatenated together), and we take the ℓ_2-norm over the corresponding single gradient vector. This is the standard way to clip gradients in PyTorch.

[4] `https://www.lumi-supercomputer.eu`

[5] The MI250X is a dual-chip module, meaning in practice that each physical device consists of two logical devices, so each node has 8 logical GPU devices with 64GB of memory each.

	OLMo-7B	LLaMA2-7B	OpenLM-7B	Falcon-7B
warmup steps	5000	2000	2000	1000
peak LR	3.0E-04	3.0E-04	3.0E-04	6.0E-04
minimum LR	3.0E-05	3.0E-05	3.0E-05	1.2E-05
weight decay	0.1	0.1	0.1	0.1
beta1	0.9	0.9	0.9	0.99
beta2	0.95	0.95	0.95	0.999
epsilon	1.0E-05	1.0E-05	1.0E-05	1.0E-05
LR schedule	linear	cosine	cosine	cosine
gradient clipping	global 1.0	global 1.0	global 1.0	global 1.0
gradient reduce dtype	FP32	FP32	FP32	BF16
optimizer state dtype	FP32	most likely FP32	FP32	FP32

Table 5: Comparison of pretraining optimizer settings at the 7B scale. Each model in this table used AdamW as its optimizer.

- **MosaicML:** Provided by MosaicML[6] (Databricks), we used 27 nodes on this cluster, where each node consists of 8x NVIDIA A100 GPUs with 40GB of memory and 800Gbps interconnect.

Despite minor differences in batch size to optimize for training throughput, both runs resulted in nearly identical performance on our evaluation suite by 2T tokens.

4 Results

The checkpoint used for evaluating OLMo-7B is trained until 2.46T tokens on the Dolma (Soldaini et al., 2024) dataset with a linear learning rate decay schedule mentioned in Section 3.2. In our experiments, we find that tuning this checkpoint further on Dolma dataset for 1000 steps with the learning rate linearly decayed to 0 boosts model performance on perplexity and end-task evaluation suites described in Section 2.3. We compare OLMo with other publicly available models including LLaMA-7B (Touvron et al., 2023a), LLaMA2-7B (Touvron et al., 2023b), MPT-7B (MosaicML NLP Team, 2023), Pythia-6.9B (Biderman et al., 2023), Falcon-7B (Almazrouei et al., 2023) and RPJ-INCITE-7B (Together Computer, 2023).

4.1 Downstream evaluation

Setup Our core **downstream evaluation suite** (see Table 6) consists of: arc (both `arc_easy` and `arc_challenge`) (Clark et al., 2018), `boolq` (Clark et al., 2019), `openbookqa` (Mihaylov et al., 2018), `sciq` (Welbl et al., 2017), `hellaswag` (Zellers et al., 2019), `piqa` (Bisk et al., 2020), `copa` (Roemmele et al., 2011) and `winogrande` (Sakaguchi et al., 2021). In Appendix A, we also report results on an additional set of auxiliary tasks outside of our core evaluation set that we found to have less stable performance trends (see Figure 4). We note that our downstream evaluation suite is still under development and that additional results and analysis will be reported in a future version.

In all cases, we perform zero-shot evaluation using the rank classification approach popularized by Brown et al. (2020). Under this approach, candidate text completions (e.g., different multiple-choice options) are ranked by likelihood (usually normalized by some normalization factor), and prediction accuracy is reported. While Catwalk implements several common likelihood normalization strategies, including normalizing by number of tokens (per-token normalization) (Brown et al., 2020; Liang et al., 2022), by number of characters (per-character normalization) (Gao et al., 2023), as well as incorporating an answer's unconditional likelihood (Brown et al., 2020), we selected the normalization strategies for each dataset separately. Specifically, we used unconditional normalization for `arc` and `openbookqa`, per-token normalization for `hellaswag`, `piqa`, and `winogrande` and no normalization for `boolq`, `copa`, and `sciq` (i.e., tasks formulated as single token prediction tasks).

[6] `https://www.mosaicml.com`

7B Models	arc challenge	arc easy	boolq	copa	hella-swag	open bookqa	piqa	sciq	wino-grande	avg.
Falcon	47.5	70.4	74.6	86.0	75.9	53.0	78.5	93.9	68.9	72.1
LLaMA	44.5	57.0	73.1	85.0	74.5	49.8	76.3	89.5	68.2	68.7
LLaMA2	39.8	57.7	73.5	87.0	74.5	48.4	76.4	90.8	67.3	68.4
MPT	46.5	70.5	74.2	85.0	77.6	48.6	77.3	93.7	69.9	71.5
Pythia	44.2	61.9	61.1	84.0	63.8	45.0	75.1	91.1	62.0	65.4
RPJ-INCITE	42.8	68.4	68.6	88.0	70.3	49.4	76.0	92.9	64.7	69.0
OLMo-7B	48.5	65.4	73.4	90.0	76.4	50.4	78.4	93.8	67.9	71.6

Table 6: Zero-shot evaluation of OLMo-7B and 6 other publicly available comparable model checkpoints on 9 core tasks from the downstream evaluation suite described in Section 2.3. For OLMo-7B, we report results for the 2.46T token checkpoint.

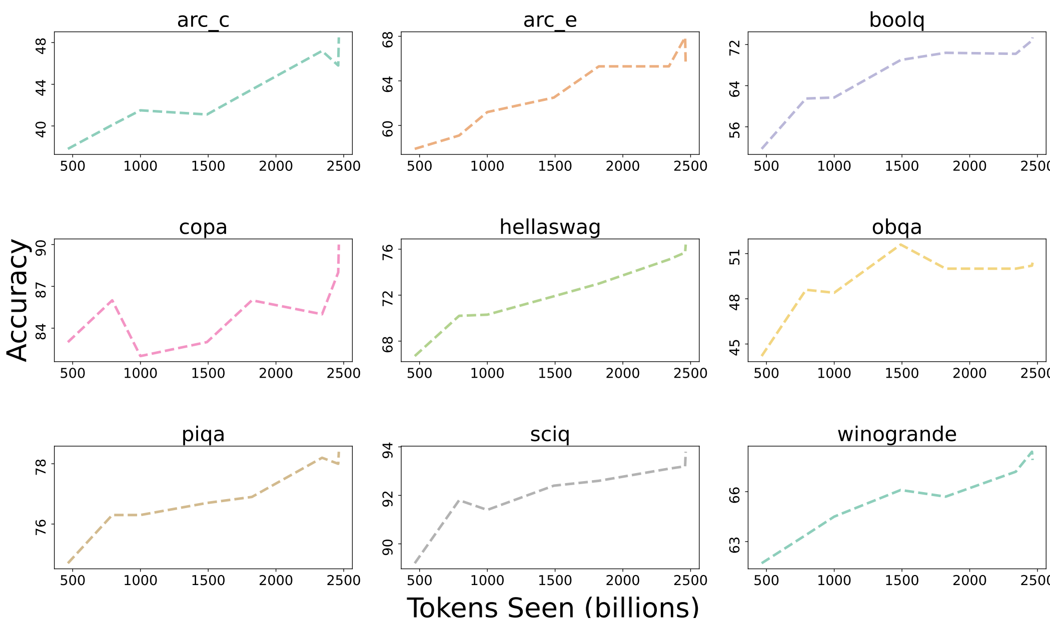

Figure 1: Accuracy score progression of OLMo-7B on 9 core end-tasks score from Catwalk evaluation suite described in Section 2.3. We can see the benefit of decaying LR to 0 in the final 1000 steps of training on 7/9 end-tasks.

Results Table 6 summarizes the result of zero-shot evaluation of OLMo-7B and compares it against 6 other publicly available models of comparable size. We report results on 9 core tasks from our evaluation suite described in Section 2.3. Our OLMo-7B checkpoint outperforms all other publicly available models on 2 end-tasks and remains in top-3 on 8/9 end-tasks from the evaluation suite. On aggregate, OLMo-7B is competitive against all 6 publicly available model checkpoints in our comparison table.

In Figure 1 we plot the accuracy score progression of 9 core end-tasks. All tasks, except OBQA, show an upward trend in accuracy numbers as OLMo-7B is trained on more tokens. A sharp upward tick in accuracy of many tasks between the last and the second to last step shows us the benefit of linearly reducing the LR to 0 over the final 1000 training steps. See Table 8 in Appendix A for additional evaluation results and discussion.

4.2 Intrinsic language modeling evaluation

Setup For intrinsic evaluations, Paloma proposes a range of analyses, from inspection of performance in each domain separately to more summarized results over combinations of domains. We

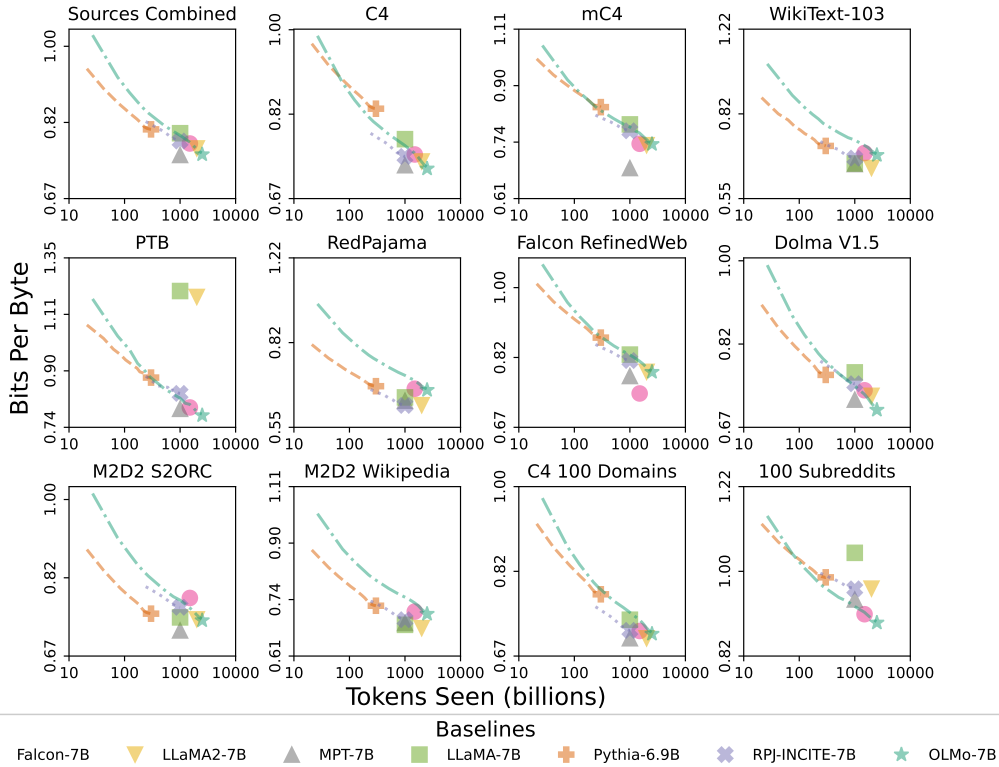

Figure 2: Bits per byte on 11 evaluation data sources from Paloma and their combination (Magnusson et al., 2023), decontaminated from OLMo's pretraining data. While models follow a general data scaling trend, sample efficiency is most favorable on in-distribution data. For example, OLMo-7B overtakes all other models on C4, perhaps from having 88.8% Common Crawl pretraining data.

report results at two levels of granularity: the aggregate performance over 11 of the 18 sources in Paloma as in Magnusson et al. (2023), as well as more fine-grained results over each of these sources individually. This particular subset of 11 sources from Paloma excludes sources that are not publicly available, involve fringe or toxic text, or consist of code data not supported by Paloma's decontamination approach. This leaves C4 (Raffel et al., 2020), mC4-en (Chung et al., 2023), Wikitext 103 (Merity et al., 2016), Penn Treebank (Marcus et al., 1999; Nunes, 2020), RedPajama (Together Computer, 2023), Falcon-RefinedWeb (Penedo et al., 2023), Dolma (Soldaini et al., 2024), M2D2 S2ORC (Reid et al., 2022), M2D2 Wikipedia (Reid et al., 2022), C4 100 domains (Chronopoulou et al., 2022), and Dolma 100 Subreddits (Soldaini et al., 2024). To allow for a fair comparison between models with different vocabularies, we report bits per byte as defined by Gao et al. (2020) over the test sets of these sources.

Results In the *Sources Combined* subplot of Figure 2, we show the performance of OLMo-7B against 6 comparably-sized language models on the combination of 11 data sources from Paloma. Overall we find OLMo to have a competitive fit, especially given its training data was explicitly decontaminated against Paloma. As seen through the comparison of final models (see shapes) as well intermediate checkpoints (see dashed lines), the OLMo results follow similar scaling trends of other models. Note that the performance of intermediate checkpoints is influenced by where that checkpoint occurs in the learning rate schedule. So models trained for fewer steps will tend to have steeper training curves without necessarily being more sample efficient if training duration were fixed across all models. MPT-7B, nevertheless, stands out as improving ahead of the other models in this subplot. This could be due to a number of factors, including pretraining data composition and its match to the domains in Paloma (e.g., MPT trains on 27% non-Common Crawl data rather than 18% for LLaMA, 12.2% for RedPajama, and 11.2% for OLMo) as well as various data preprocessing decisions (e.g., MPT's use of semantic deduplication by Abbas et al., 2023, on C4).

The remaining subplots in Figure 2 provide more fine-grained analysis by reporting bits per byte separately for each of the 11 data sources that are combined in the aggregated Paloma metric. From this we see greater variation in sample efficiency, largely driven by the similarity of training and evaluation distributions. Notably, OLMo-7B fares well on evaluations predominated by Common Crawl, such as C4, though different ways of postprocessing Common Crawl are best fit by models trained with that specific data, such as Falcon-7B on Falcon RefinedWeb. Meanwhile, OLMo-7B is less sample efficient compared to other models on sources less related to scraped web text, such as WikiText-103, M2D2 S2ORC, and M2D2 Wikipedia. The RedPajama evaluation shows a similar pattern, perhaps as only 2 of its 7 domains are from Common Crawl, and Paloma weights domains within each source equally. Since heterogeneous data from curated sources like Wikipedia and ArXiv papers is much less abundant than scraped web text, maintaining sample efficiency for fit to these distributions of language will be challenging as pretraining corpora are scaled.

4.3 Power Consumption and Carbon Footprint

Following previous literature (Strubell et al., 2019; Patterson et al., 2021; Wu et al., 2022; Dodge et al., 2022), we estimate the total energy consumed and carbon released while pretraining our models by calculating the total power consumption required for training, and then multiplying it by the carbon emission intensity of the power grid where the model was trained. While reporting these operational emissions is standard practice, it does not account for other sources of emissions such as the embodied emissions due to the manufacturing, transportation and disposal of hardware and datacenter infrastructure, lifetime operational emissions due to use, rebound effects, or other environmental impacts such as water consumption or mining. Thus our estimates should be viewed as lower bounds.

We calculate the total power consumption for our models by measuring the power consumption of a single node every 25ms, calculating an average across the entire training run, and multiplying by the total number of nodes. We then account for the energy efficiency of the data center by multiplying the previous total by a power usage effectiveness (PUE) factor, which we set to 1.1, representing a conservative 10% energy consumption overhead typical of energy efficient datacenters.[7][8] We estimate that pretraining our 7B models consumed **239 MWh** of energy.

To calculate carbon emissions, we multiply the total power consumption by a carbon intensity factor, measured in kg CO_2 emitted per KWh, based on the physical location of the data center where each model was trained. The model trained on A100-40GB GPUs was trained in Australia, so we assume a carbon intensity factor of 0.610, the national average for Australia in 2022.[9] The model trained on MI250X GPUs was trained in the LUMI supercomputer, which runs on 100% renewable, carbon-neutral energy, so we assume a carbon intensity factor of 0. LUMI is powered entirely by hydroelectric power and some sources (Ubierna et al., 2022) measure the carbon intensity factor of hydroelectric power to be 0.024, which would imply total carbon emissions of 3.54 tCO_2eq.[10] However, we rely on the official LUMI data for our calculations, and thus we estimate total pretraining emissions of **69.78 tCO_2eq**.[11] In Table 7 we compare our models with other previously released models based on publicly available information.

We hope that openly releasing our models can reduce future emissions by allowing others to avoid the need to pretrain models from scratch, and give insights into the true cost of developing state of the art models. We also highlight that our estimates are lower bounds, because they do not include other critical pieces of development such as debugging, hyperparameter tuning, and downtime.

[7] https://www.nrel.gov/computational-science/measuring-efficiency-pue.html

[8] https://www.google.com/about/datacenters/efficiency/

[9] https://www.cleanenergyregulator.gov.au/Infohub/Markets/Pages/qcmr/december-quarter-2022/Emissions-Reduction.aspx

[10] https://www.lumi-supercomputer.eu

[11] These metrics were in part collected using Carbonara's AI agent and monitoring platform. Learn more at: https://trycarbonara.com

	GPU Type	GPU Power Consumption (MWh)	Power Usage Effectiveness	Carbon Intensity (kg CO_2e/KWh)	Carbon Emissions (tCO_2eq)
Gopher-280B	TPU v3	1,066	1.08	0.330	380
BLOOM-176B	A100-80GB	433	1.2	0.057	30
OPT-175B	A100-80GB	324	1.1	0.231	82
T5-11B	TPU v3	77	1.12	0.545	47
LLaMA-7B	A100-80GB	33	1.1	0.385	14
LLaMA2-7B	A100-80GB	74	1.1	0.385	31
OLMo-7B	MI250X	135	1.1	0.000*	0*
OLMo-7B	A100-40GB	104	1.1	0.610	70

Table 7: CO_2 emissions during pretraining. We estimate the total carbon emissions for various models using publicly available data on PUE, carbon intensity of local power grid, and reported power consumption. Numbers for Gopher-280B (Rae et al., 2022), BLOOM-176B (Luccioni et al., 2022), OPT-175B (Zhang et al., 2022), T5-11B (Patterson et al., 2021), LLaMA (Touvron et al., 2023a), and LLaMA2 (Touvron et al., 2023b) are taken from their respective papers. See Section 4.3 for details on how tCO2eq was calculated.

* LUMI runs entirely on hydroelectric power[11] and some estimates (Ubierna et al., 2022) measure the intensity factor of hydroelectric power to be 0.024, implying total emissions of 3.54 tCO_2eq.

5 Artifacts Released

By sharing artifacts from all pipeline stages, we aim to encourage open research and reduce duplicated, often costly efforts, by academics and practitioners. We release the following:

1. The training and modeling code.[12].
2. The trained model weights for the 7B model[13], 7B-twin-2T[14], and the 1B model[15]. For all the models, we release not only the final model weights but also 500+ intermediate checkpoints at intervals of 1000 steps.
3. The training data Dolma (Soldaini et al., 2024).[16]
4. Dolma's toolkit to construct new datasets [17], and WIMBD (Elazar et al., 2023) for dataset analysis[18].
5. The evaluation code[19] using Catwalk[20] for downstream evaluation (Groeneveld et al., 2023) and Paloma[21] for perplexity-based evaluation (Magnusson et al., 2023).

We intend to follow up on this release with another one soon that includes the following:

1. Training logs, ablations, and findings.
2. Weights & Biases logs for our training runs.
3. Adapted OLMo with instruction-tuning and RLHF, including its training and evaluation code and data using our Open Instruct[22] library (Wang et al., 2023; Ivison et al., 2023).

[12]https://github.com/allenai/OLMo
[13]https://huggingface.co/allenai/OLMo-7B
[14]https://huggingface.co/allenai/OLMo-7B-Twin-2T
[15]https://huggingface.co/allenai/OLMo-1B
[16]https://huggingface.co/datasets/allenai/dolma
[17]https://github.com/allenai/dolma
[18]https://github.com/allenai/wimbd
[19]https://github.com/allenai/OLMo-Eval
[20]https://github.com/allenai/catwalk
[21]https://paloma.allen.ai
[22]https://github.com/allenai/open-instruct

6 License

Our goal is to facilitate scientific development and empower the scientific community, so we favor permissive licenses that give users flexibility in using our resources and artifacts. As such, all code and weights are released under the Apache 2.0 License.[23] Some licenses used by other organizations for recent model releases prohibit using the outputs from their models to train artificial intelligence or machine learning systems, while we expressly allow users to do so. We also do not limit commercial use. We hope that our models can make other models better. We recognize that the risk for misuse of our models is relatively low, as language models that have *not* been adapted as chatbots have primarily been used as scientific artifacts not as products with broad public adoption (our models have not been adapted as chatbots). In addition, over the past year there have been a number of comparable models released with very permissive licenses, so using a more strict license for our work will not remove the overall risk in the field. We believe this tradeoff on the side of being more open is the best option.

7 Conclusion and Future Work

This technical report presents our first release of OLMo, a state-of-the-art, truly open language model and its framework to build and study the science of language modeling. Unlike most prior efforts that have only released model weights and inference code, we release OLMo and the whole framework, including training data and training and evaluation code. Soon, we will also release training logs, ablations, findings and Weights & Biases logs. We are also exploring the adaptation of OLMo with instruction tuning and different flavors of RLHF. We are going to release the adapted models as well as all of our model adaptation code and data.

We intend to continuously support and extend OLMo and its framework, and continue to push the boundaries of open LMs to empower the open research community. To that end, we look forward to bringing different model sizes, modalities, datasets, safety measures, and evaluations into the OLMo family. We hope this and future releases will empower and strengthen the open research community and inspire a new wave of innovation.

Author Contributions

OLMo would not have been possible without the help of our many teammates and collaborators. We list author contributions (in alphabetical order) below:

Contributors to **pretraining dataset construction and tooling** (Dolma) include Russell Authur, Iz Beltagy, Akshita Bhagia, Khyathi Chandu, Jesse Dodge, Yanai Elazar, Dirk Groeneveld, Rodney Kinney, Kyle Lo, Aakanksha Naik, Abhilasha Ravichander, Dustin Schwenk, Luca Soldaini, and Nishant Subramani.

Contributors to **model training and architecture** include Shane Arora, Iz Beltagy, Akshita Bhagia, Matthew E. Peters, Dirk Groeneveld, Ananya Harsh Jha, William Merrill, Jacob Morrison, Niklas Muennighoff, Dustin Schwenk, Saurabh Shah, Pete Walsh, and Mitchell Wortsman.

Contributors to **evaluation suite and tooling** include Akshita Bhagia, Arman Cohan, Pradeep Dasigi, Jesse Dodge, Dirk Groeneveld, Yuling Gu, Tushar Khot, Kyle Richardson, Oyvind Tajford, and Pete Walsh.

Contributors to **model adaptation** include Iz Beltagy, Pradeep Dasigi, Jack Hessel, Hamish Ivison, Nathan Lambert, Valentina Pyatkin, Pete Walsh, and Yizhong Wang.

Contributors to **license creation and risk assessment** include David Atkinson, Jesse Dodge, Jennifer Dumas, Crystal Nam, and Will Smith.

The OLMo project was led by Hannaneh Hajishirzi and Noah A. Smith.

[23] http://www.apache.org/licenses/LICENSE-2.0

Acknowledgements

OLMo would not have been possible without the support of many individuals and institutions. The experimental components of this work were made possible through a partnership with AMD and CSC, enabling use of the LUMI supercomputer, and Kempner Institute at Harvard University. We thank Jonathan Frankle and the team at MosaicML (now Databricks) for sharing their experiences with FSDP, and building the code base that OLMo is based on. We thank our teammates Taira Anderson, Michelle Benedict, Jon Borchardt, Evie Cheng, Arnavi Chheda, Johann Dahm, Matt Latzke, Kelsey MacMillan, Aaron Sarnat, Carissa Schoenick, Sam Skjonsberg, Michael Schmitz, Michael Wilson, Caitlin Wittlif, and the entire IT team, for their help with the website, design, internal and external communications, budgeting, and other activities that supported smooth progress on this project. Finally, we also express gratitude for the helpful discussions and feedback from our teammates at AI2 and close collaborators, including Prithviraj (Raj) Ammanabrolu, Peter Clark, Nicole DeCario, Doug Downey, Ali Farhadi, Ian Ferreira, Väinö Hatanpää, Sham M. Kakade, Julien Launay, Sydney Levine, Pekka Manninen, Franzi Roessner, Maarten Sap, Ludwig Schmidt, and Yulia Tsvetkov.

References

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

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra-Aimée Cojocaru, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models. *ArXiv*, abs/2311.16867, 2023. URL https://api.semanticscholar.org/CorpusID:265466629.

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016. URL https://api.semanticscholar.org/CorpusID:8236317.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003. URL https://api.semanticscholar.org/CorpusID:221275765.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

BigScience, 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 arXiv:2211.05100*, 2022.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

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. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022. URL https://arxiv.org/abs/2204.06745.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1120. URL https://aclanthology.org/D16-1120.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher 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. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783.

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 Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben 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 Garcia, 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 Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

Alexandra Chronopoulou, Matthew Peters, and Jesse Dodge. Efficient hierarchical domain adaptation for pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1336–1351, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.96. URL https://aclanthology.org/2022.naacl-main.96.

Hyung Won Chung, Noah Constant, Xavier García, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *ArXiv*, abs/2304.09151, 2023. URL https://api.semanticscholar.org/CorpusID:258187051.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

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 arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Jesse Dodge, Taylor Prewitt, Remi Tachet Des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. Measuring the carbon intensity of ai in cloud instances, 2022. URL https://dl.acm.org/doi/10.1145/3531146.3533234.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *International Joint Conference on Natural Language Processing*, 2005. URL https://www.microsoft.com/en-us/research/publication/automatically-constructing-a-corpus-of-sentential-paraphrases/.

Yanai Elazar, Akshita Bhagia, Ian H. Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hanna Hajishirzi, Noah A. Smith, and Jesse Dodge. What's in my big data? *ArXiv*, abs/2310.20707, 2023. URL https://api.semanticscholar.org/CorpusID:264803575.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Sidney Greenbaum and Gerald Nelson. The international corpus of english (ICE) project. *World Englishes*, 15(1):3–15, mar 1996. doi: 10.1111/j.1467-971x.1996.tb00088.x. URL https://doi.org/10.1111%2Fj.1467-971x.1996.tb00088.x.

Dirk Groeneveld, Anas Awadalla, Iz Beltagy, Akshita Bhagia, Ian Magnusson, Hao Peng, Oyvind Tafjord, Pete Walsh, Kyle Richardson, and Jesse Dodge. Catwalk: A unified language model evaluation framework for many datasets. *arXiv preprint arXiv:2312.10253*, 2023. URL https://arxiv.org/abs/2312.10253.

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. URL https://github.com/mlfoundations/open_lm/. GitHub repository.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2, 2023. URL https://arxiv.org/abs/2311.10702.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/abs/2401.04088.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022. URL https://arxiv.org/abs/2211.09110.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *CoRR*, abs/2007.08124, 2020. URL https://arxiv.org/abs/2007.08124.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023. URL https://arxiv.org/abs/2312.06550.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model, 2022. URL https://arxiv.org/abs/2211.02001.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, et al. Paloma: A benchmark for evaluating language model fit. *arXiv preprint arXiv:2312.10523*, 2023.

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Treebank-3, 1999. URL https://catalog.ldc.upenn.edu/LDC99T42.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843, 2016. URL https://api.semanticscholar.org/CorpusID:16299141.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Frederick Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. *ArXiv*, abs/1710.03740, 2017. URL https://api.semanticscholar.org/CorpusID:3297437.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018. URL https://arxiv.org/abs/1809.02789.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*, 2013. URL https://api.semanticscholar.org/CorpusID:16447573.

MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL www.mosaicml.com/blog/mpt-7b. Accessed: 2023-05-05.

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

Davide Nunes. Preprocessed penn tree bank, 2020. URL https://zenodo.org/record/3910021.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Antonis Papasavva, Savvas Zannettou, Emiliano De Cristofaro, Gianluca Stringhini, and Jeremy Blackburn. Raiders of the lost kek: 3.5 years of augmented 4chan posts from the politically incorrect board. *Proceedings of the International AAAI Conference on Web and Social Media*, 14:885–894, may 2020. doi: 10.1609/icwsm.v14i1.7354. URL https://doi.org/10.1609%2Ficwsm.v14i1.7354.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training, 2021. URL https://arxiv.org/abs/2104.10350.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra-Aimée 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*, abs/2306.01116, 2023. URL https://api.semanticscholar.org/CorpusID:259063761.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *ArXiv*, abs/1802.05365, 2018. URL https://api.semanticscholar.org/CorpusID:3626819.

Mohammad Taher Pilehvar and José Camacho-Collados. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *CoRR*, abs/1808.09121, 2018. URL http://arxiv.org/abs/1808.09121.

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 Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby 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 Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher, 2022. URL https://arxiv.org/abs/2112.11446.

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. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2019. URL https://api.semanticscholar.org/CorpusID:203736482.

Machel Reid, Victor Zhong, Suchin Gururangan, and Luke Zettlemoyer. M2D2: A massively multi-domain language modeling dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 964–975, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.63.

Manoel Horta Ribeiro, Jeremy Blackburn, Barry Bradlyn, Emiliano De Cristofaro, Gianluca Stringhini, Summer Long, Stephanie Greenberg, and Savvas Zannettou. The evolution of the manosphere across the web. *Proceedings of the International AAAI Conference on Web and Social Media*, 15:196–207, may 2021. doi: 10.1609/icwsm.v15i1.18053. URL https://doi.org/10.1609%2Ficwsm.v15i1.18053.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011. URL https://aaai.org/papers/02418-2418-choice-of-plausible-alternatives-an-evaluation-of-commonsense-causal-reasoning/.

Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021. URL https://dl.acm.org/doi/abs/10.1145/3474381.

Noam M. Shazeer. Glu variants improve transformer. *ArXiv*, abs/2002.05202, 2020. URL https://api.semanticscholar.org/CorpusID:211096588.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Pete Walsh, Hannaneh Hajishirzi, Noah A. Smith, Luke Zettlemoyer, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2024.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021. URL https://api.semanticscholar.org/CorpusID:233307138.

Together Computer. RedPajama: An Open Source Recipe to Reproduce LLaMA training dataset, April 2023. URL 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*, abs/2302.13971, 2023a. URL https://api.semanticscholar.org/CorpusID:257219404.

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, 2023b. URL https://arxiv.org/abs/2307.09288.

María Ubierna, Cristina Díez Santos, and Sara Mercier-Blais. *Water Security and Climate Change: Hydropower Reservoir Greenhouse Gas Emissions*, pages 69–94. Springer Singapore, Singapore, 2022. ISBN 978-981-16-5493-0. doi: 10.1007/978-981-16-5493-0_5. URL https://doi.org/10.1007/978-981-16-5493-0_5.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

David Vilares and Carlos Gómez-Rodríguez. HEAD-QA: A healthcare dataset for complex reasoning. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 960–966, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1092. URL https://aclanthology.org/P19-1092.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ArXiv*, abs/1804.07461, 2018. URL https://arxiv.org/abs/1804.07461.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources, 2023. URL https://arxiv.org/abs/2306.04751.

Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*, 2017. URL https://arxiv.org/abs/1707.06209.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga Behram, James Huang, Charles Bai, Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin S. Lee, Bugra Akyildiz, Maximilian Balandat, Joe Spisak, Ravi Jain, Mike Rabbat, and Kim Hazelwood. Sustainable ai: Environmental implications, challenges and opportunities, 2022. URL https://arxiv.org/abs/2111.00364.

Savvas Zannettou, Barry Bradlyn, Emiliano De Cristofaro, Haewoon Kwak, Michael Sirivianos, Gianluca Stringini, and Jeremy Blackburn. What is gab: A bastion of free speech or an alt-right echo chamber. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1007–1014, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356404. doi: 10.1145/3184558.3191531. URL https://doi.org/10.1145/3184558.3191531.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019. URL https://arxiv.org/abs/1905.07830.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *ArXiv*, abs/1910.07467, 2019. URL https://api.semanticscholar.org/CorpusID:113405151.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

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. *Proc. VLDB Endow.*, 16:3848–3860, 2023. URL https://api.semanticscholar.org/CorpusID:258297871.

A Additional Evaluation

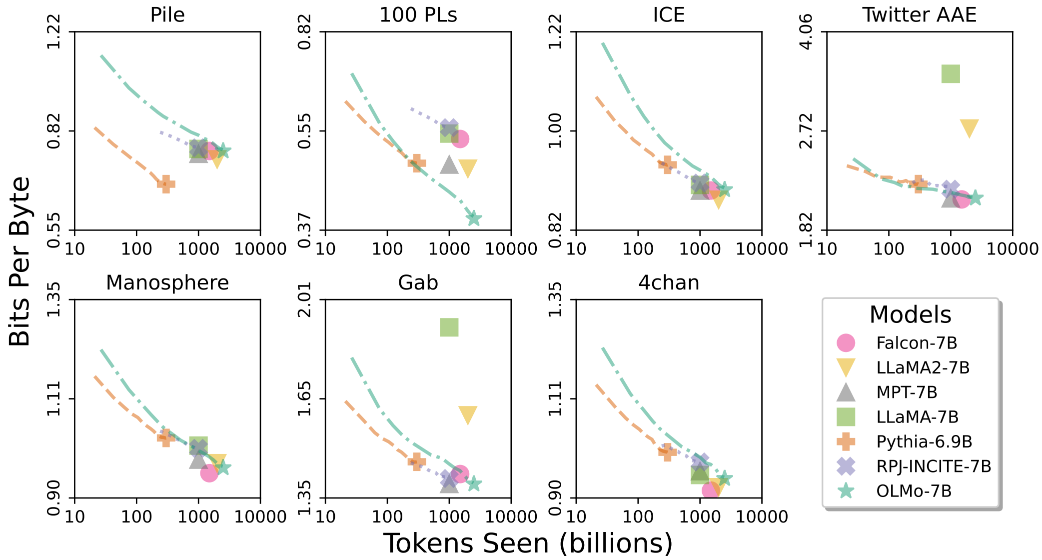

Figure 3: Bits per byte for each of the 7 remaining Paloma data sources not aggregated in Figure 2.

Additional perplexity results In Figure 3 we provide results for each of the 7 data sources in Paloma (Magnusson et al., 2023) that are excluded from the combined metric in Figure 2. Some of these sources such as Pile (Gao et al., 2020) and ICE (Greenbaum and Nelson, 1996) are not publicly available at this time. Dolma 100 Programming Languages (Soldaini et al., 2024) consists of code data that is not supported by the decontamination approach used in Paloma. TwitterAAE (Blodgett et al., 2016), along with ICE, are datasets for targeted analyses of disparities in performance between different dialects and as such should be evaluated separately. And finally, the Manosphere, Gab, and 4chan corpora (Ribeiro et al., 2021; Zannettou et al., 2018; Papasavva et al., 2020) are intended to examine model fit to language from fringe online communities that are studied for prevalent hate speech and toxicity. Thus minimizing perplexity on these fringe corpora is not always desirable.

One notable result here is that OLMo-7B is much farther ahead of the other models on Dolma 100 Programming Languages (100 PLs). Note that this effect may be due in part to underestimation from contamination, as decontaminating code data is beyond the scope of the method in Paloma. At the same time other models that are trained on code data from GitHub such as RPJ-INCITE-7B, that are just as likely to have contamination, fair much worse. Another factor then is that OLMo-7B trains on code data with exactly the same post-processing as that in 100 PLs while the code data in other models will have been processed differently. Similarly, Pile evaluation demonstrates these in-distribution and potential contamination effects as Pythia-6.9B achieves top performance despite being trained on almost an order of magnitude fewer tokens than OLMo-7B.

The results on the remaining 5 targeted sources should be interpreted with care, as Paloma often finds that perplexity on these sources is dominated by superficial features such as low average document length rather than fit to that which would actually be salient to members of these speech communities. TwitterAAE and Gab have among the shortest documents in Paloma contributing to unusually high bits per byte in this figure. Other than these two, the models are notably very closely grouped in a data scaling trend in ICE, Manosphere, and 4chan.

Additional end-task results Next, in Table 8, we provide results from zero-shot evaluation of OLMo-7B on 6 additional end-tasks apart from the 9 in our core evaluation suite. These tasks are `headqa_en` (Vilares and Gómez-Rodríguez, 2019), `logiqa` (Liu et al., 2020), `mrpc` (Dolan and Brockett, 2005), `qnli` (Wang et al., 2018), `wic` (Pilehvar and Camacho-Collados, 2018), and `wnli` (Wang et al., 2018).

	headqa_en	logiqa	mrpc	qnli	wic	wnli	avg.
Falcon-7B	38.6	23.7	62.8	49.8	49.5	47.9	45.4
LLaMA-7B	37.0	20.9	68.4	48.4	50.0	56.3	46.8
LLaMA2-7B	38.3	22.7	64.2	49.2	50.3	49.3	45.7
MPT-7B	37.4	22.9	67.7	52.1	48.1	47.9	46.0
Pythia-6.9B	40.1	21.5	65.4	53.8	55.0	38.0	45.6
RPJ-INCITE-7B	36.9	27.8	58.8	53.8	48.9	57.8	47.3
OLMo-7B	37.3	23.4	68.4	49.1	50.2	56.3	47.5

Table 8: Zero-shot evaluation of OLMo-7B on 6 additional end-tasks apart from the 9 present in our core evaluation suite. Once again, we compare OLMo-7B to 6 other model checkpoints which are publicly available. We find that OLMo-7B outperforms the other models on aggregate taken over 6 additional end-tasks from this table, however these tasks were also found to provide limited signal during training (see Figure 4).

We note, however, that in contrast to our core evaluation set described in Section 4.1, we found these additional end-tasks to have less stable performance during model development, and to provide a limited signal. This is illustrated in Figure 4, where we see the progress of task performance throughout training to be more random (compare with the more stable upward trends in Figure 1). While tasks such as `mrpc` and `wic` appear more stable, they offered additional difficulties related to performance being tied to random chance (e.g., `wic`) or the tendency of models to make spurious predictions (e.g., always predicting a single label) that either inflate or deflate performance due to dataset class imbalances (e.g., `mrpc`). We therefore caution against relying too heavily on these tasks when measuring model performance throughout training and comparing models.

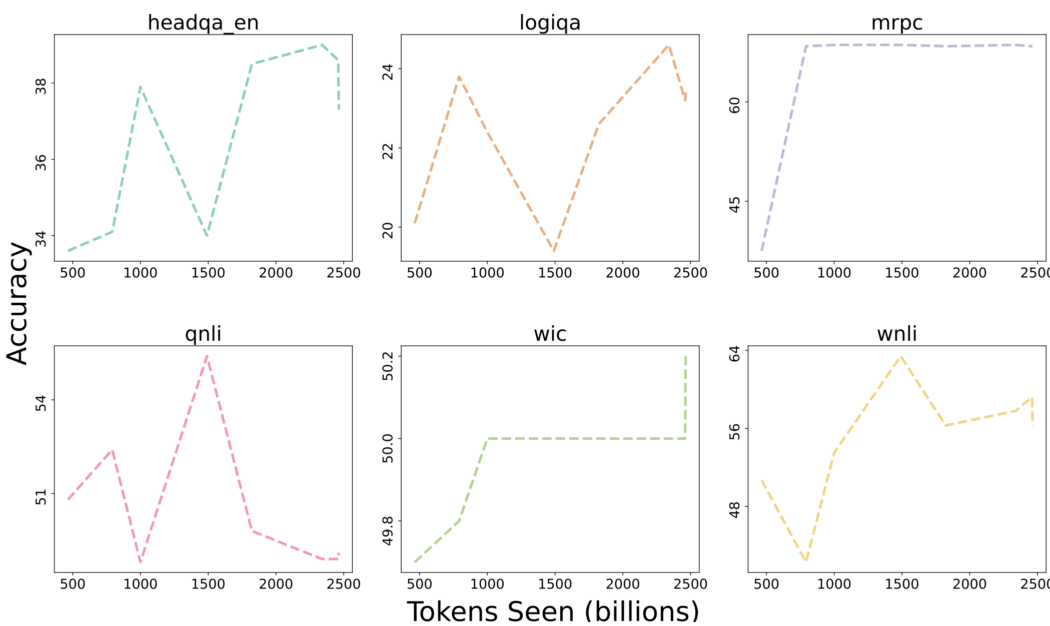

Figure 4: Accuracy score progression of OLMo-7B on 6 additional end-tasks. The performance of these additional end-tasks was unstable and provided limited signal during model development.