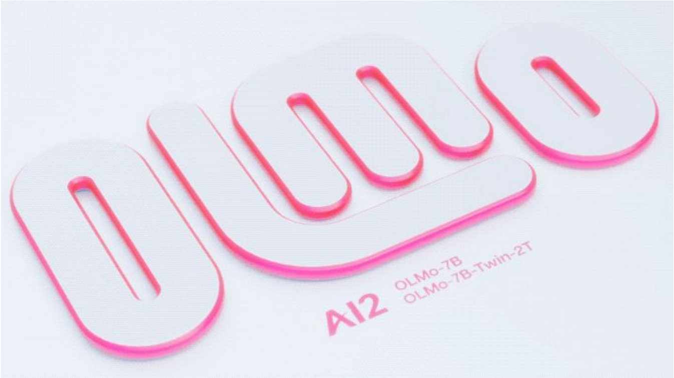
OLMo: Accelerating the Science of Language Models

Dirk Groeneveld[α] Iz Beltagy[α] Pete Walsh[α] Akshita Bhagia[α] Rodney Kinney[α] Oyvind Tafjord[α] Ananya Harsh Jha[α] Hamish Ivisonα[β] Ian Magnusson[α] Yizhong Wang[αβ] Shane Arora[α] David Atkinson[α] Russell Authur[α] Khyathi Raghavi Chandu[α] Arman Cohanγ[α] 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 Hajishirziα[β]

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

Aizawa Lab Paper Reading Group
01/03/24

Introduction



What is OLMo?

A new LLM and the first really fully open one with similar

OLMo: **O**pen **L**anguage **Mo**del

Contributions/ Main steps of the paper

- Create new dataset (in their previous paper Dolma)
- Train model on dataset from scratch
- Compare to existing similar model and obtain comparable performances
- Release everything involved in the process of creation to make it the most open existing model

https://github.com/allenai/OLMo

OLMo Framework - Model and Architecture

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.*

Classic **decoder-only transformer**+ improvement like PaLM, OpenLM, LLaMA and Falcon:

- No biases: Excluding bias term to improve training stability
- Non-parametric layer norm
- SwiGLU activation function
- Rotary Positional embeddings
- Different tokenizer -> vocabulary: 50,280 tokens

OLMo Framework - Model and Architecture

	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.

OLMo Framework - Pretraining Data: Dolma

"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."

Dolma:

3T tokens

5B documents

7 different data sources

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.

Training OLMo

Batch size: 4M tokens

format: bfloat16

	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.

Data

- 2T-token from their dataset
- Pipaline: concatenated, divided in chunks of 2048 tokens and shuffled

Hardware

Lumi supercomputer (AMD GPUs)

MosaicML NVIDIA GPU

OLMo Framework - Evaluation

In-Loop Training Ablations

- Throughout model training - every 1000 training steps (or ~4B training tokens)
- to make decisions about model design: optimizers, learning rate schedule,data mixtures…
- early and continuous signal on the quality of the model being trained

Downstream Evaluation

zero-shot performance on a set of 9 tasks corresponding to the commonsense reasoning task

Intrinsic Language Modeling Evaluation

- measure how OLMo-7B fits distributions of language
- Evaluated on 11 domains of text

Results - Downstream evaluation

zero-shot evaluation using rank classification approach

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.

Results

- measure how OLMo-7B fits distributions of language
- Evaluated on 11 domains of text

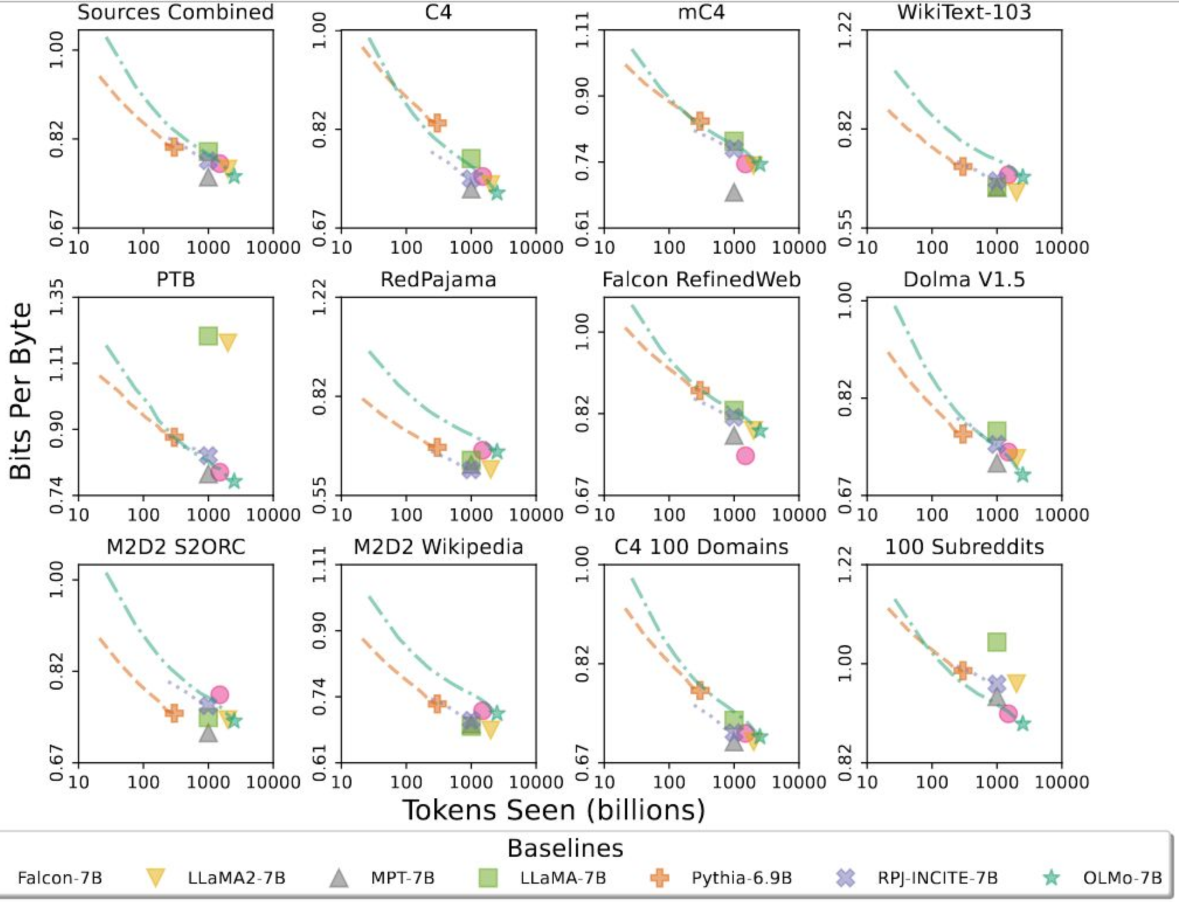


Figure 2: Bits per byte on 11 evaluation data sources from Paloma and their combination (Magnusson et al., 2023), decontaminated from OLMo's pretraining data. While models follow a general data scaling trend, sample efficiency is most favorable on in-distribution data. For example, OLMo-7B overtakes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

Power Consumption and Carbon Footprint

	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

136 824
repas végétariens
Comparer

23 815 700
km en tgv
Comparer

320 680
km en voiture
Comparer

9 612
repas avec du boeuf
Comparer

459 200
km en avion
Comparer

1 847
smartphones
Comparer

Why I choosed this article?

➖ Methods and results really similar to other models

➕ Fully open, everything is realised and publicly available:

🤗	**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