A tour of data, eval and scaling language models with Olmo

David Heineman

Ai2

At Ai2, we build fully open LM recipes

To facilitate research and accelerate the science of LMs, we need language models that are fully open.

OLMo is being used to…

Attribute model behavior to pretraining data

LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia[1*] Sadhika Malladi[1*] Suchin Gururangan[2] Sanjeev Arora[1] Danqi Chen[1]

Inform policy by studying LM development ecosystem

Consent in Crisis: The Rapid Decline of the AI Data Commons

Weak-to-Strong Extrapolation Expedites Alignment

Chujie Zheng[1,2*] Ziqi Wang[3] Heng Ji[3] Minlie Huang[1†] Nanyun Peng[2†]
[1]The CoAI Group, DCST, BNRist, Tsinghua University
[2]University of California, Los Angeles [3]University of Illinois Urbana-Champaign
chujiezhengchn@gmail.com aihuang@tsinghua.edu.cn violetpeng@cs.ucla.edu

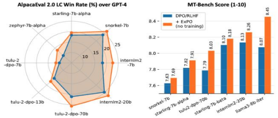


Figure 1: With *no additional training*, ExPO remarkably improves off-the-shelf DPO/RLHF models on HuggingFace across various model sizes and capabilities, as evaluated on two leading LLM benchmarks: AlpacaEval 2.0 [30] (left) and MT-Bench [61] (right). See Table 1 for full results.

Understand the effects of fine-tuning a pretrained model

From Pre-training Corpora to Large Language Models: What Factors Influence LLM Performance in Causal Discovery Tasks?

Tao Feng[1], Lizhen Qu[1], Niket Tandon[2], Zhuang Li[1], Xiaoxi Kang[1], Gholamreza Haffari[1]
[1]Monash University, [2]Allen Institute for AI

Demystifying Forgetting in Language Model Fine-Tuning with Statistical Analysis of Example Associations

Xisen Jin and Xiang Ren
University of Southern California
{xisenjin, xiangren}@usc.edu

Fishing for Magikarp: Automatically Detecting Under-trained Tokens in Large Language Models

Sander Land, Cohere, sander@cohere.com
Max Bartolo, Cohere, max@cohere.com

Amuro & Char: Analyzing the Relationship between Pre-Training and Fine-Tuning of Large Language Models

Kaiser Sun Mark Dredze
Johns Hopkins University
Baltimore, MD USA
{hsun74,mdredze}@cs.jhu.edu

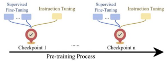
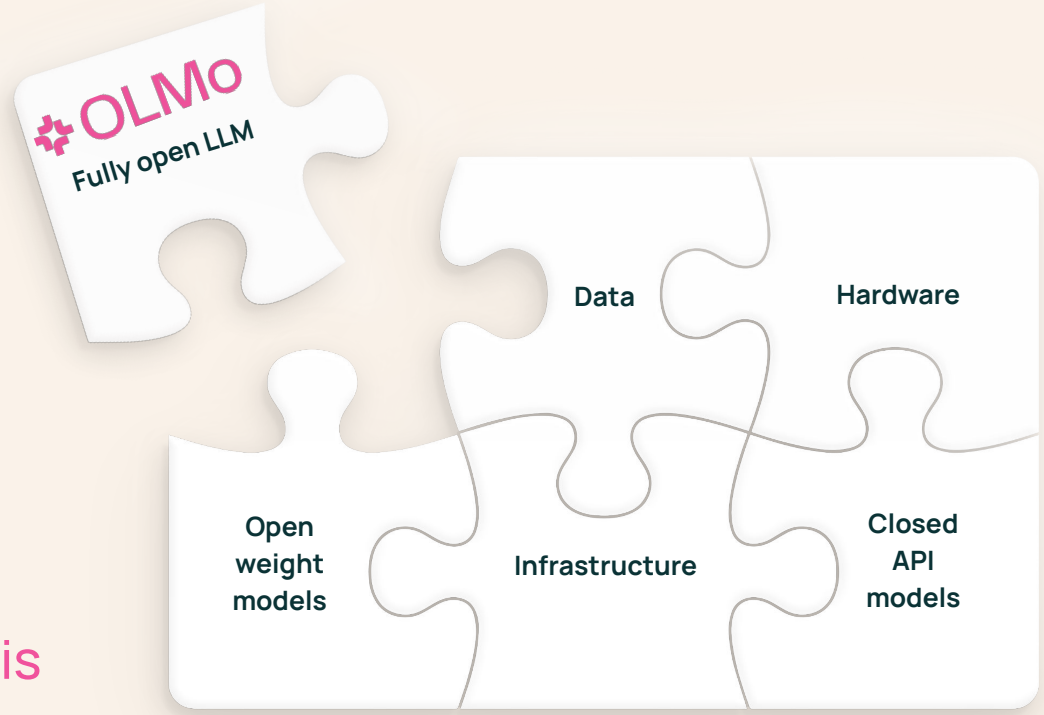


Figure 1: Illustration of the experimental scheme.

Ai2

What does fully open mean?

- Model weights
- Intermediate checkpoints
- Detailed recipes
- All the data
- Code to reproduce
- Documentation and analysis

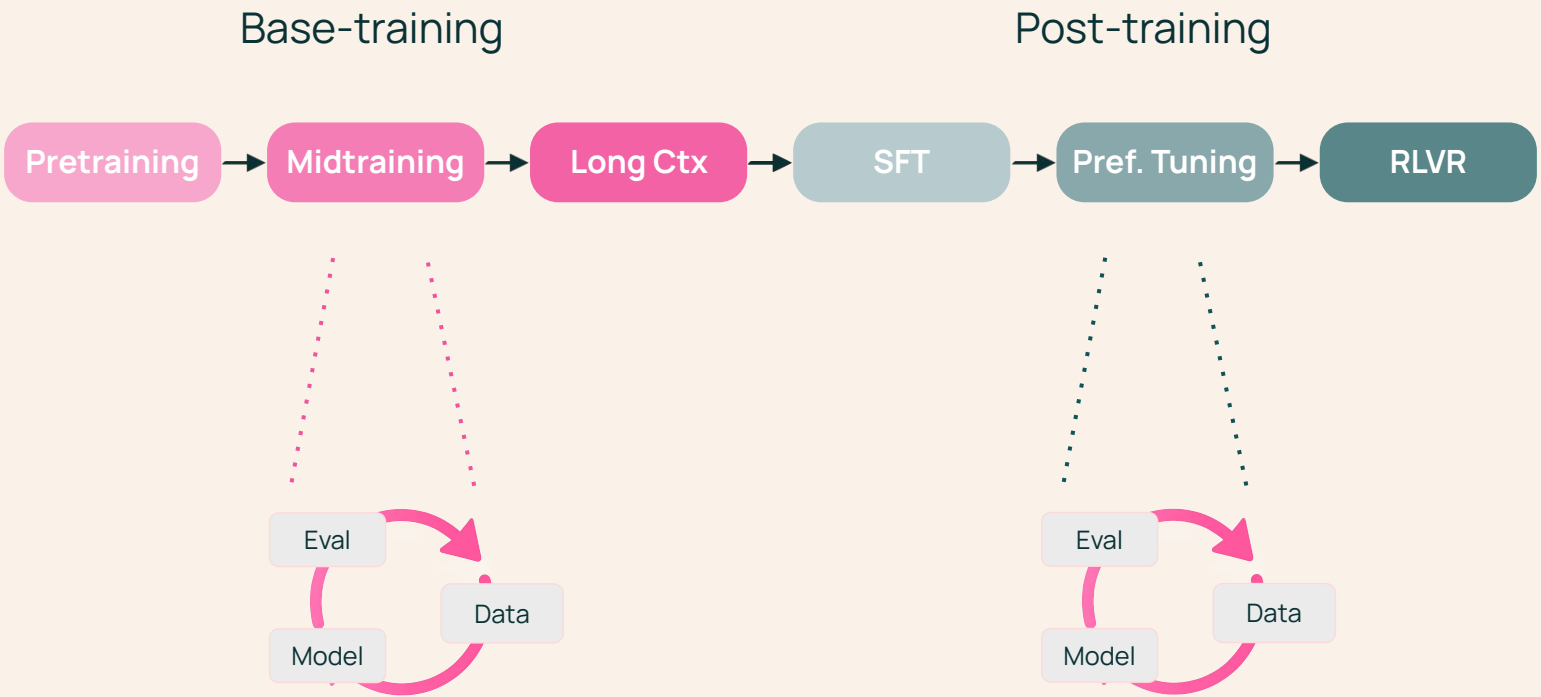


OLMo is being used to...

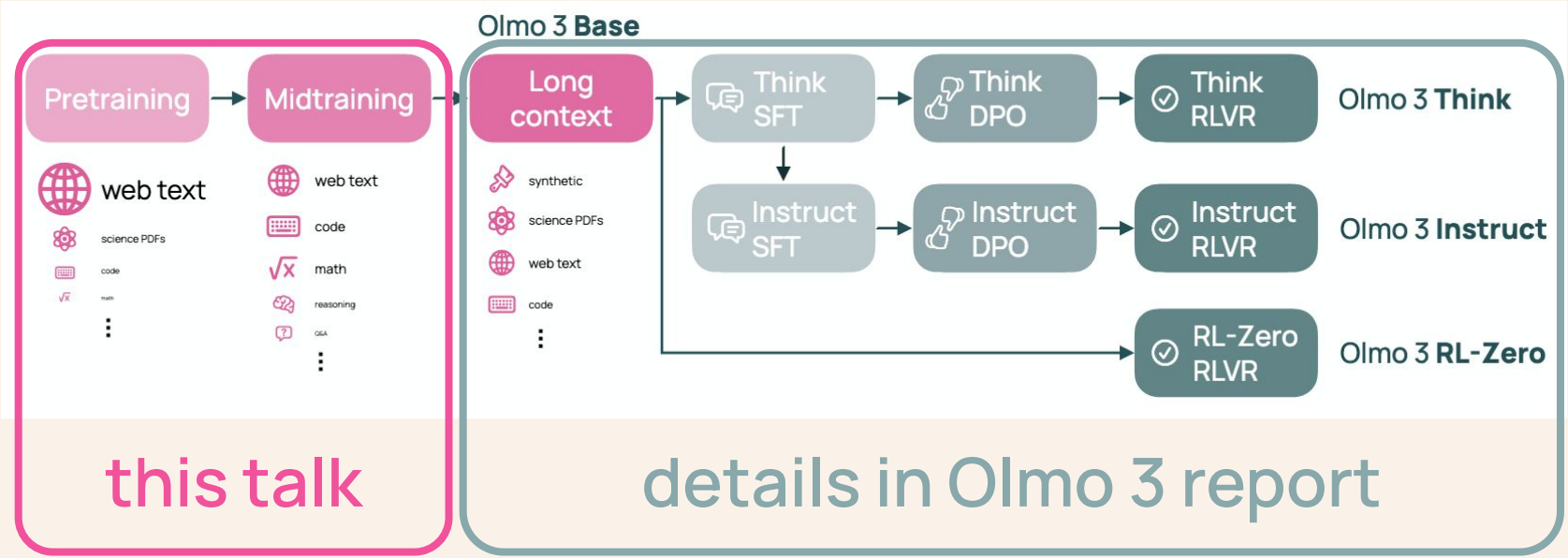
Develop new architectures

Study how models acquire/forget knowledge during training

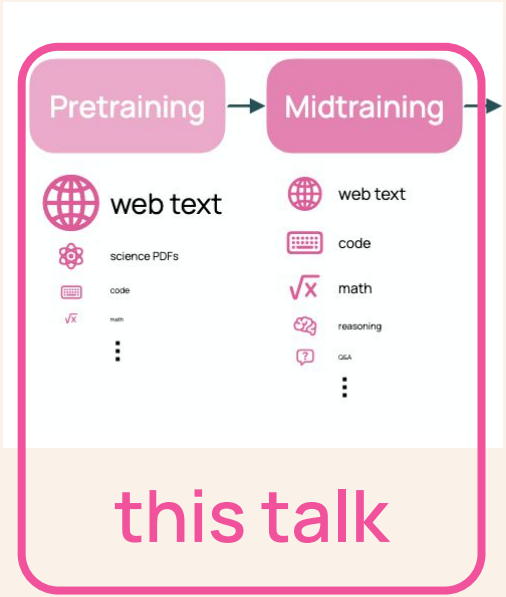
Olmo 3 Model Flow

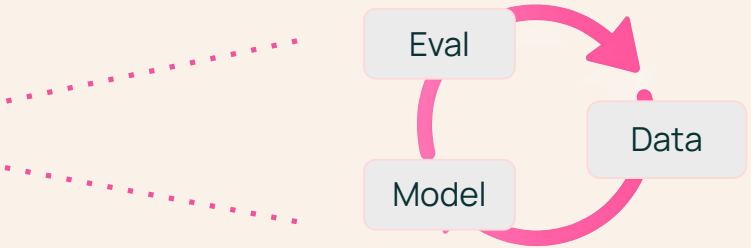


Olmo 3 Model Flow



Olmo 3 Model Flow





Outline

- Background
- Evaluation Methods
- Eval in Olmo 3
- Data Methods
- Data in Olmo 3
- Overview of Olmo 3

Signal and Noise: A Framework for Reducing Uncertainty in Language Model Evaluation

David Heineman[♤] Valentin Hofmann[♤♡] Ian Magnusson[♤♡] Yuling Gu[♤]
Noah A. Smith[♤♡] Hannaneh Hajishirzi[♤♡] Kyle Lo[♤] Jesse Dodge[♤]

[♤]Allen Institute for Artificial Intelligence
[♡]Paul G. Allen School of Compu

Developing large language mo
small experiments, typically by
this work, we analyze specific
for such decisions, and interv
marks. We introduce two key
signal, a benchmark's ability
noise, a benchmark's sensitivi
demonstrate that benchmarks w
when making decisions at smal
law prediction error. These re
lead to more useful benchma
to directly affect signal or n
a metric that has better signa
leads to better reliability and in
noisy subtasks, to improve a
reliable multi-task evaluations,
intermediate checkpoints to re
conclude by recommending tha
existing benchmarks to use, ain
marks for these experiments, a
32B parameters, resulting in a
benchmark results, totaling 20

allenai/signal-and-no

1 Introduction

Language model development is exper
make decisions such as what architectu

Olmix: A Framework for Data Mixing Throughout LM Development

Mayee F. Chen[1,2] Tyler Murray[1] David Heineman[1] Matt Jordan[1] Hannaneh Hajishirzi[1,3]
Christopher Ré[2] Luca Soldaini[1] Kyle Lo[1,3]

[1]Allen Institute for AI [2]Stanford University [3]University of Washington

Code: Olmix Data: Olmix Contact: mfchen@cs.stanford.edu {lucas,kylel}@allenai.org

Abstract

Data mixing—determining the ra
training language models (LMs). W
applied during real-world LM devel
challenges. First, the configuratio
design choices across existing met
like data constraints. We conduct
design choices lead to a strong mix
LM development as datasets are ad
unaddressed by existing works, wh
the mixture after the domain set is
mixture reuse, a mechanism that re
by the update. Over a sequence o
mixture reuse matches the perform
compute and improves over trainin

1 Introduction

Figure 1 Two problems with data mixing
method? (2) How to efficiently mix under

Modern language models (LMs) are tr

Olmo 3

Olmo Team*

Allyson Ettinger[♥1] Amanda Bertsch[♥1,3] Bailey Kuehl[♥1] David Graham[♥1]
David Heineman[♥1] Dirk Groeneveld[♥1] Faeze Brahman[♥1] Finbarr Timbers[♥1]
Hamish Ivison[♥1,2] Jacob Morrison[♥1,2] Jake Poznanski[♥1] Kyle Lo[♥1,2] Luca Soldaini[♥1]
Matt Jordan[♥1] Mayee Chen[♥1,4] Michael Noukhovitch[♥1,5,6] Nathan Lambert[♥1]
Pete Walsh[♥1] Pradeep Dasigi[♥1] Robert Berry[♥1] Saumya Malik[♥1] Saurabh Shah[♥1]
Scott Geng[♥1,2] Shane Arora[♥1] Shashank Gupta[♥1] Taira Anderson[♥1] Teng Xiao[♥1]
Tyler Murray[♥1] Tyler Romero[♥1] Victoria Graf[♥1,2]

Akari Asai[1,3] Akshita Bhagia[1] Alexander Wettig[7] Alisa Liu[2] Aman Rangapur[1]
Chloe Anastasiades[1] Costa Huang[1] Dustin Schwenk[1] Harsh Trivedi[1] Ian Magnusson[1,2]
Jaron Lochner[1] Jiacheng Liu[1] Lester James V. Miranda[1] Maarten Sap[1,3] Malia Morgan[1]
Michael Schmitz[1] Michal Guerquin[1] Michael Wilson[1] Regan Huff[1] Ronan Le Bras[1]
Rui Xin[2] Rulin Shao[2] Sam Skjonsberg[1] Shannon Zejiang Shen[8] Shuyue Stella Li[2]
Tucker Wilde[1] Valentina Pyatkin[1] Will Merrill[1] Yapei Chang[9] Yuling Gu[1] Zhiyuan Zeng[1,2]

Ashish Sabharwal[1] Luke Zettlemoyer[2] Pang Wei Koh[1,2]
Ali Farhadi[1,2] Noah A. Smith[♥1,2] Hannaneh Hajishirzi[♥1,2]

[1]Allen Institute for AI [2]University of Washington [3]Carnegie Mellon University [4]Stanford University [5]Mila
[6]Université de Montréal [7]Princeton University [8]Massachusetts Institute of Technology [9]University of Maryland

*Olmo 3 was a team effort; authors sorted alphabetically. ♥marks core contributors. See author contributions here.

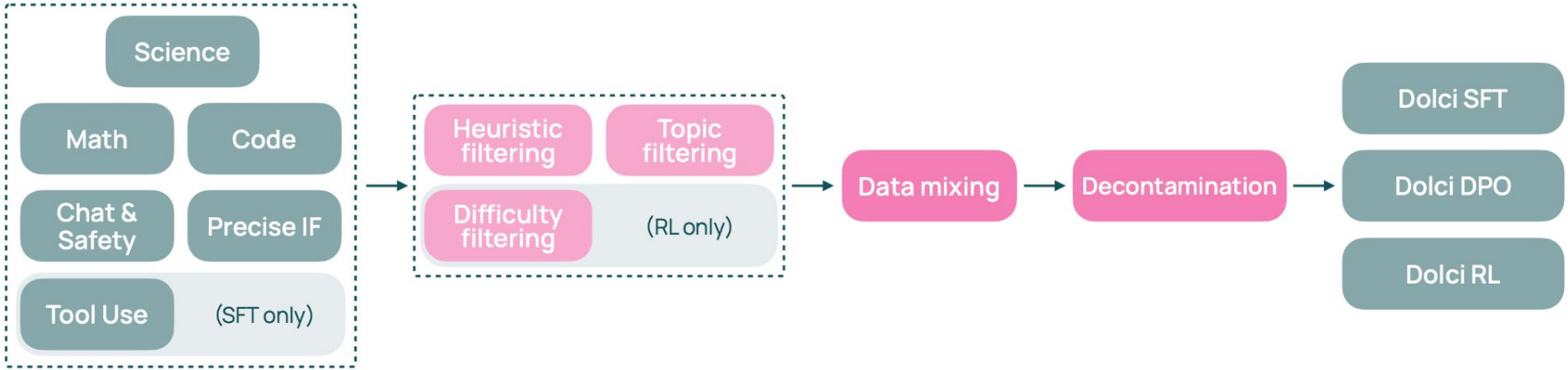
Olmo 3 Base: Olmo-3-1025-7B Olmo-3-1125-32B
Olmo 3 Think: Olmo-3-7B-Think Olmo-{3|3.1}-32B-Think
Olmo 3 Instruct: Olmo-3-7B-Instruct Olmo-3.1-32B-Instruct
Olmo 3 RL Zero: Olmo-3-7B-RL-Zero-{Math|Code|IF|General|Mix} Olmo-3.1-7B-RL-Zero-{Math|Code}
Base Data: Pretrain: Dolma 3 Mix Midtrain: Dolma 3 Dolmino Mix Long-ctx: Dolma 3 Longmino Mix
Think Data: Dolci-Think-{SFT|DPO|RL}-7B Dolci-Think-{SFT|DPO|RL}-32B
Instruct Data: Dolci-Instruct-{SFT|DPO|RL}
RL-Zero Data: Dolci-RL-Zero-{Math|Code|IF|General}-7B Dolci-RL-Zero-Mix-7B
Training Code: OLMo-core (pretrain) Open Instruct (posttrain)
Data Code: datamap-rs (data processing) duplodocus (deduplication) dolma3 (data recipes)
Eval Code: OLMES (eval suite) decon (eval decontamination)
Training Logs: Olmo-3-7B-{Base|Think|Instruct|RL-Zero} Olmo-3-32B-{Base|Think|Instruct}
Demo: 32B Think 32B Instruct 7B Think 7B Instruct
Contact: olmo@allenai.org

Abstract

We introduce Olmo 3, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. Olmo 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire model flow, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, Olmo 3.1 Think 32B, is the strongest fully-open thinking model released to-date.

Ai2

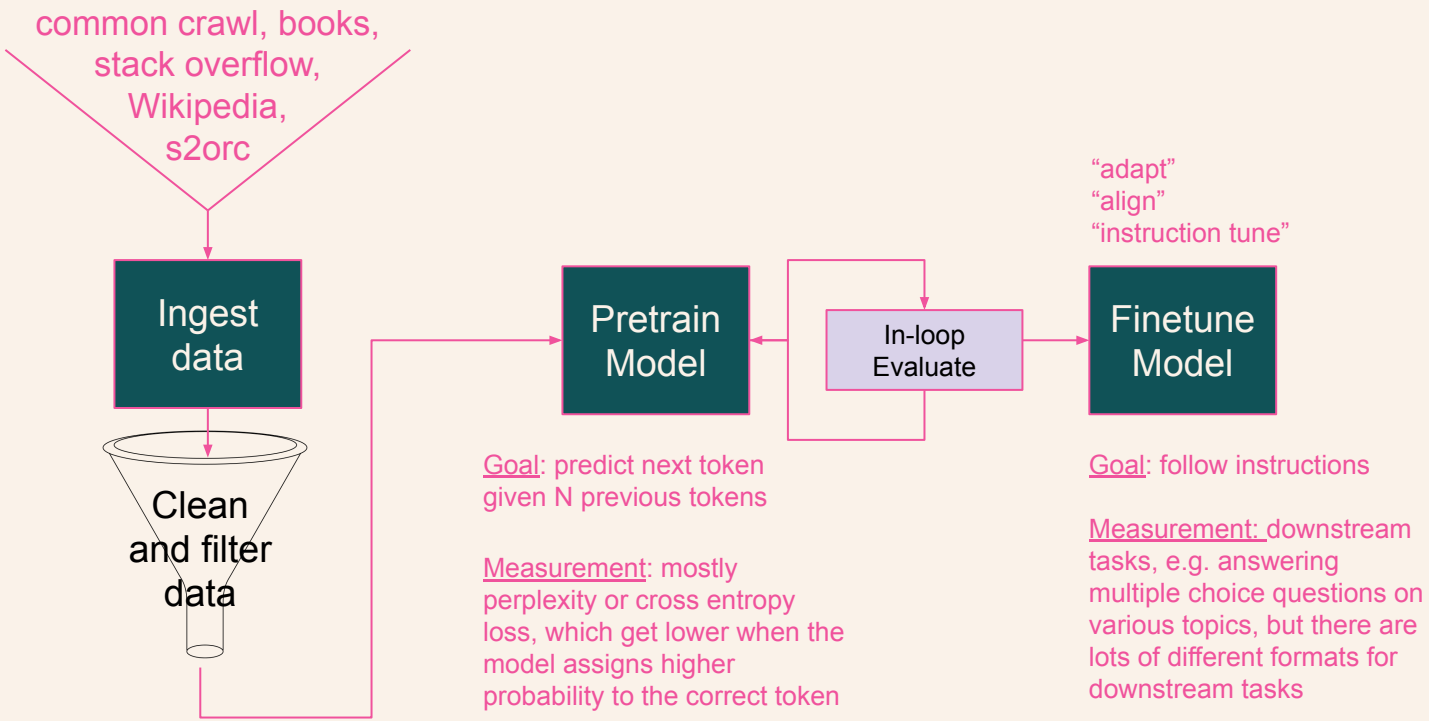
Posttrain for Olmo 3 Think



SFT Data
Modular data curation & mixing

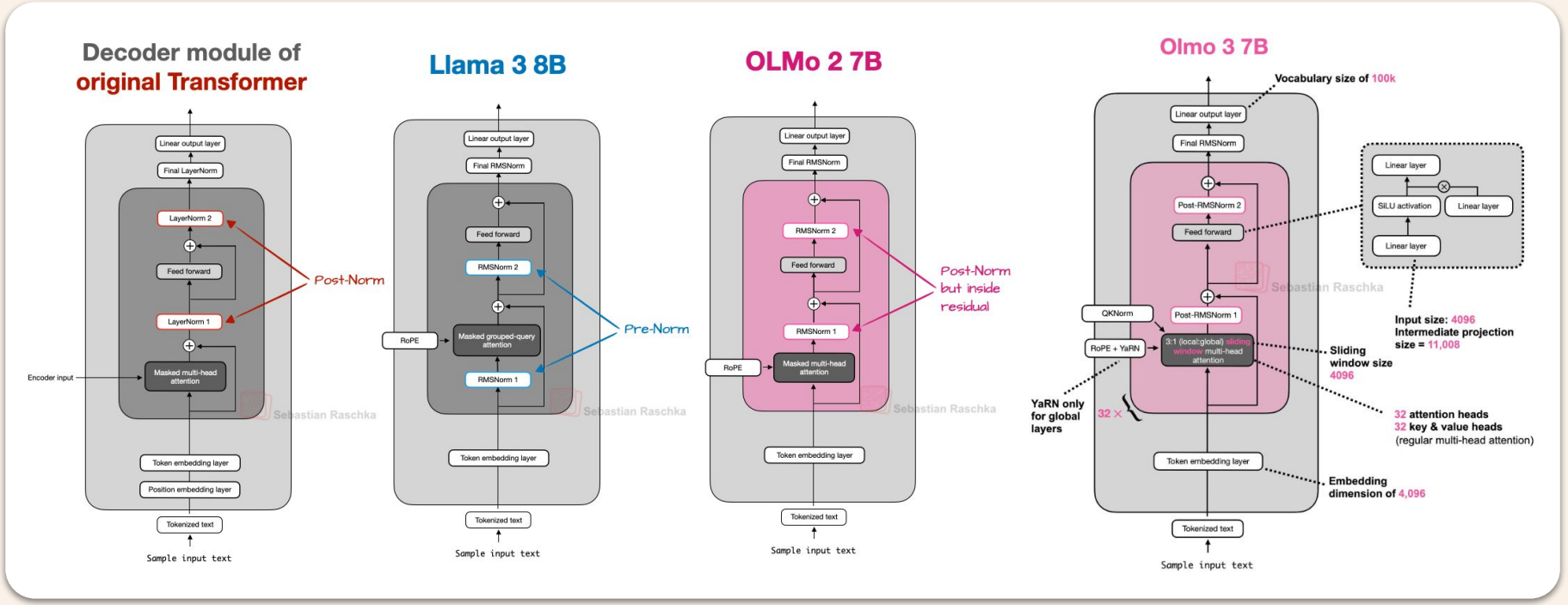
Preference data
Learn from contrast between data points

Scaled up RL
Enable thinking & adaptive RL


common crawl, books, stack overflow, Wikipedia, s2orc
Ingest data
Clean and filter data
Pretrain Model
In-loop Evaluate
Finetune Model
“adapt”
“align”
“instruction tune”
Goal: predict next token given N previous tokens
Measurement: mostly perplexity or cross entropy loss, which get lower when the model assigns higher probability to the correct token
Also some downstream task evals at checkpoints.
Goal: follow instructions
Measurement: downstream tasks, e.g. answering multiple choice questions on various topics, but there are lots of different formats for downstream tasks

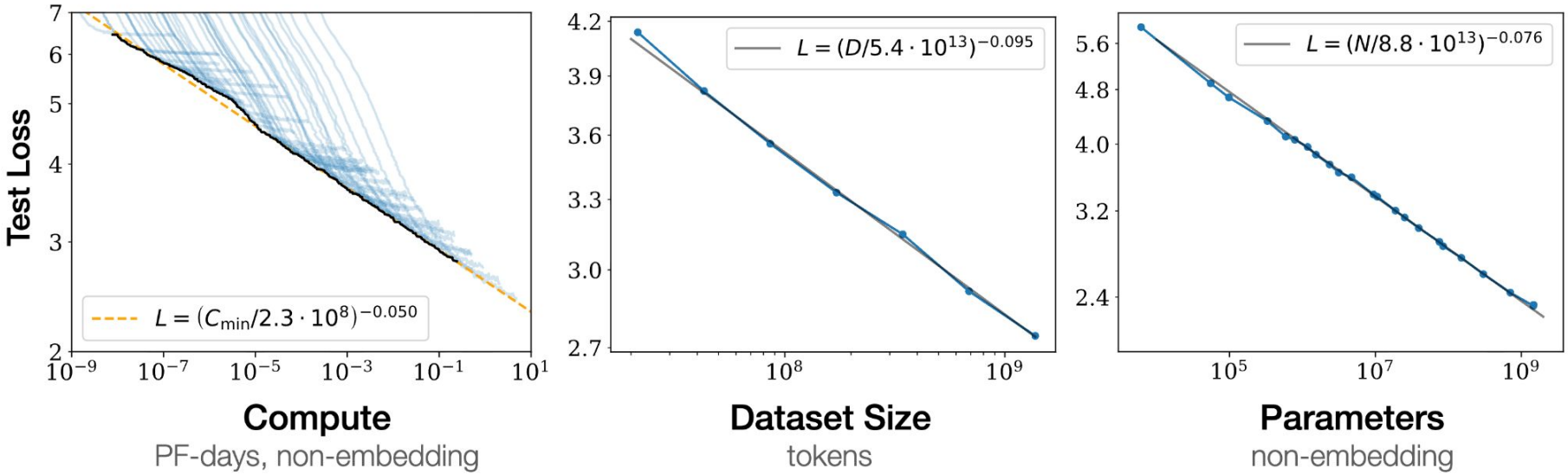
Background

Background: Architecture Changes



magazine.sebastianraschka.com/p/the-big-llm-architecture-comparison

Background: Scaling Pretrained Models

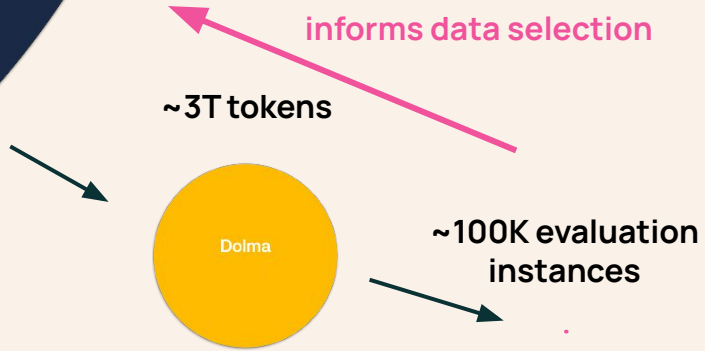


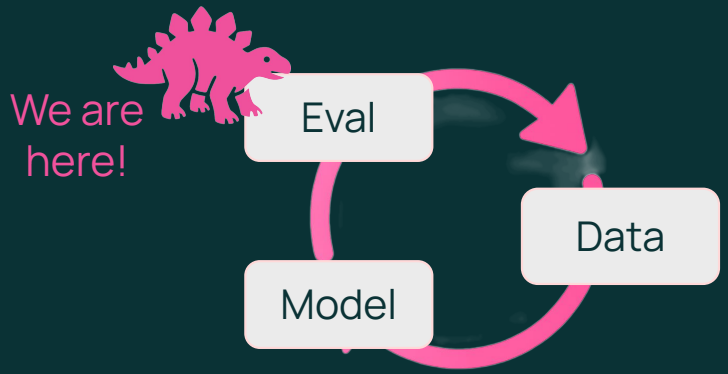
Scaling Laws for Neural Language Models (Kaplan et al., 2020)

Background: Data

Rough estimate:
text in ~30M books in the US Library of Congress ≈ between 1/400 and 1/24 PB

Common Crawl
~1 PB





Signal and Noise: A Framework for Reducing Uncertainty in Language Model Evaluation

David Heineman[μ] Valentin Hofmann[μσ] Ian Magnusson[μσ] Yuling Gu[μ]
Noah A. Smith[μσ] Hannaneh Hajishirzi[μσ] Kyle Lo[μ] Jesse Dodge[μ]

[μ]Allen Institute for Artificial Intelligence
[σ]Paul G. Allen School of Computer Science & Engineering, University of Washington
contact: davidh@allenai.org

Abstract

Developing large language models is expensive and involves making decisions with small experiments, typically by evaluating on large, multi-task evaluation suites. In this work, we analyze specific properties which make a benchmark more reliable for such decisions, and interventions to design higher-quality evaluation benchmarks. We introduce two key metrics that show differences in current benchmarks: `signal`, a benchmark's ability to separate better models from worse models, and `noise`, a benchmark's sensitivity to random variability between training steps. We demonstrate that benchmarks with a better `signal-to-noise` ratio are more reliable when making decisions at small scale, and those with less `noise` have lower scaling law prediction error. These results suggest that improving `signal` or `noise` will lead to more useful benchmarks, so we introduce three interventions designed to directly affect `signal` or `noise`. For example, we propose that switching to a metric that has better `signal` and `noise` (e.g., perplexity rather than accuracy) leads to better reliability and improved scaling law error. We also find that filtering noisy subtasks, to improve an aggregate `signal-to-noise` ratio, leads to more reliable multi-task evaluations. We also find that averaging the output of a model's intermediate checkpoints to reduce `noise` leads to consistent improvements. We conclude by recommending that those creating new benchmarks, or selecting which existing benchmarks to use, aim for high `signal` and low `noise`. We use 30 benchmarks for these experiments, and 375 open-weight language models from 60M to 32B parameters, resulting in a new, publicly available dataset of 900K evaluation benchmark results, totaling 200M instances.

`allenai/signal-and-noise` `datasets/allenai/signal-and-noise`

1 Introduction

Language model development is expensive. During the development process, researchers need to make decisions such as what architecture to use, what training methods to employ, and what data to

How to make sense of the wall of results?

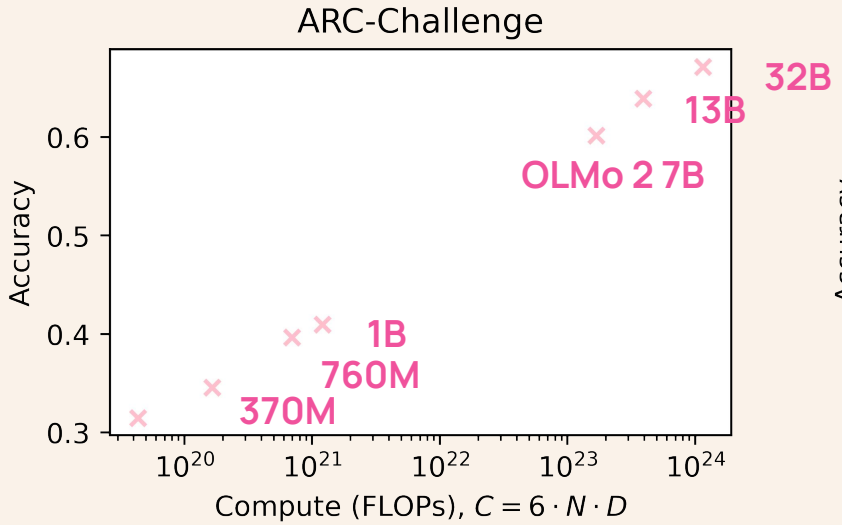
	Fully-open Models						Open-weight Models					
	Olmo 3 32B	Marin 32B	Apertus 70B	Gaperon 24B	LLM 360 K2V2 70B	OLMo 2 32B	Qwen 2.5 32B	Gemma 3 27B	Mistral 3.1 24B	Seed 36B	Gemma 2 27B	Llama 3.1 70B
OlmoBaseEval Math	61.9	49.3	39.7	20.7	46.2	53.9	64.7	63.2	59.5	15.3	57.5	62.0
GSM8k	80.6	69.1	63.0	33.3	66.7	77.6	81.1	81.3	79.3	26.9	76.3	81.2
GSM Symbolic	61.2	42.0	38.6	14.5	44.4	53.1	56.2	61.2	59.1	10.3	57.3	64.6
MATH	43.8	36.8	17.4	14.2	27.4	31.0	56.7	47.0	40.1	8.7	38.8	40.2
OlmoBaseEval Code	39.7	30.8	23.3	19.4	35.2	20.5	48.3	41.6	42.4	54.9	41.0	36.3
BigCodeBench	43.7	34.5	24.0	17.0	39.8	22.2	48.1	44.0	46.4	50.7	43.4	43.4
HumanEval	65.8	52.3	32.5	31.2	51.2	29.4	65.6	62.1	65.5	71.3	57.5	57.4
DeepSeek LeetCode	2.0	1.3	1.2	0.0	2.3	0.8	8.0	5.8	0.1	13.0	4.7	0.2
DS 1000	29.4	26.3	17.8	11.0	25.4	20.4	43.3	34.3	36.3	44.0	29.7	29.5
MBPP	59.6	52.1	37.6	36.7	53.5	37.1	69.8	60.0	61.9	72.0	61.7	55.5
MultiPL HumanEval	36.0	18.5	18.4	13.0	31.3	10.5	49.7	37.7	39.0	69.2	40.3	32.2
MultiPL MBPPP	41.5	30.5	31.3	26.5	42.8	23.2	53.6	47.2	47.7	63.8	49.7	35.9
OlmoBaseEval MC$_{\text{STEM}}$	74.5	75.9	70.0	56.2	75.6	75.3	82.2	80.2	81.5	83.4	75.6	80.1
ARC MC	94.7	93.4	90.7	72.7	93.0	94.4	97.0	95.8	96.2	97.3	94.1	95.2
MMLU STEM	70.8	68.4	57.8	45.3	64.7	64.7	79.7	74.9	76.1	82.8	65.8	70.0
MedMCQA MC	57.6	61.8	55.9	42.6	63.7	60.2	68.8	64.7	68.8	69.6	61.8	67.8
MedQA MC	53.8	60.8	52.4	35.4	61.4	62.2	68.4	68.7	70.4	70.1	61.0	72.3
SciQ MC	95.5	95.1	93.3	84.9	95.3	95.1	97.1	96.8	96.3	97.1	95.1	95.4
OlmoBaseEval MC$_{\text{Non-STEM}}$	85.6	84.5	78.5	64.1	83.5	84.2	89.3	86.7	87.9	89.0	83.2	86.1
MMLU Humanities	78.3	78.9	74.1	56.7	79.3	79.7	85.0	80.5	82.7	85.7	79.3	83.4
MMLU Social Sci.	84.0	83.7	79.2	58.9	84.9	84.5	88.4	86.2	88.6	90.1	85.8	87.4
MMLU Other	75.1	75.4	70.1	55.4	76.3	75.6	81.2	80.2	81.9	82.4	76.9	79.4
CSQA MC	82.3	80.1	76.9	60.6	78.6	81.2	89.9	79.0	80.5	81.1	78.1	79.0
PiQA MC	85.6	90.5	79.0	72.0	87.3	87.7	93.3	90.3	91.0	92.5	89.0	91.5
SocialIQA MC	83.9	82.4	79.3	71.3	81.2	82.3	86.6	81.2	81.0	84.9	81.0	83.5
CoQA Gen2MC MC	96.4	93.9	87.5	67.3	92.0	94.4	96.8	95.8	94.9	96.9	94.3	95.1
DROP Gen2MC MC	87.2	71.0	56.5	48.0	64.8	68.6	86.6	84.6	86.5	90.1	66.6	70.3
Jeopardy Gen2MC MC	92.3	95.3	93.2	77.0	95.3	96.6	97.0	95.9	97.2	96.2	92.0	97.1
NaturalQs Gen2MC MC	78.0	81.0	71.9	47.5	82.4	78.6	79.9	82.0	84.6	81.4	74.5	82.4
SQuAD Gen2MC MC	98.2	97.6	95.7	90.0	96.7	97.4	97.9	97.7	97.9	98.1	97.5	97.7
OlmoBaseEval GenQA	79.8	80.3	75.0	65.3	77.1	79.1	68.5	73.5	78.0	76.0	72.9	81.6
HellaSwag RC	84.8	87.2	84.5	75.2	87.6	87.5	86.3	86.0	86.2	84.8	86.7	88.4
Winogrande RC	90.3	90.5	87.7	80.3	88.9	89.4	87.5	91.3	90.8	89.3	90.8	91.7
Lambada	75.7	76.7	74.8	58.3	76.8	77.0	76.2	77.5	79.3	76.1	76.9	79.6
Basic Skills	93.5	91.1	87.5	83.2	90.6	88.7	94.2	94.9	91.9	96.0	93.2	92.4
DROP	80.9	76.5	56.3	59.4	69.7	76.3	53.7	75.9	74.9	76.1	73.2	78.3
Jeopardy	75.3	80.5	77.2	58.9	79.8	79.1	74.0	82.1	80.3	77.4	80.7	84.0
NaturalQs	49.0	55.1	43.1	33.5	47.6	51.4	39.3	49.2	45.1	30.7	47.1	53.1
SQuAD	94.5	94.4	90.7	89.3	91.2	94.0	64.9	92.4	92.6	89.1	93.0	92.9
CoQA	74.1	70.7	72.8	49.8	61.5	68.7	40.4	12.4	61.1	64.4	14.9	73.9
OlmoBaseEval HeldOut												
LBPP	21.8	17.3	8.1	4.3	13.4	8.2	40.3	17.7	30.3	42.6	19.7	11.8
BBH	77.6	70.1	58.8	36.6	73.2	64.6	81.1	77.4	81.4	85.0	74.8	80.8
MMLU Pro MC	49.7	48.1	39.6	21.3	45.3	46.9	61.1	53.1	58.9	62.2	47.6	50.4
Deepmind Math	29.6	26.7	20.1	28.3	32.5	22.0	40.7	30.4	35.3	31.3	27.6	40.2

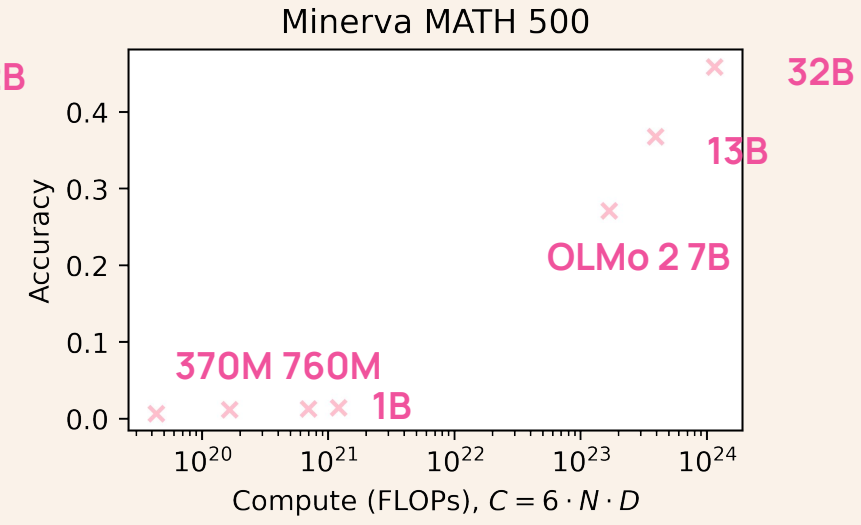
Table 2 Results comparing Olmo 3 Base 32B to other base models using the OlmoBaseEval Main suite (details in Section §3.3). OLMO 3 was not evaluated on held-out benchmarks prior to release.

Ai2

We evaluate many families of abilities (QA, Math, Coding, …)

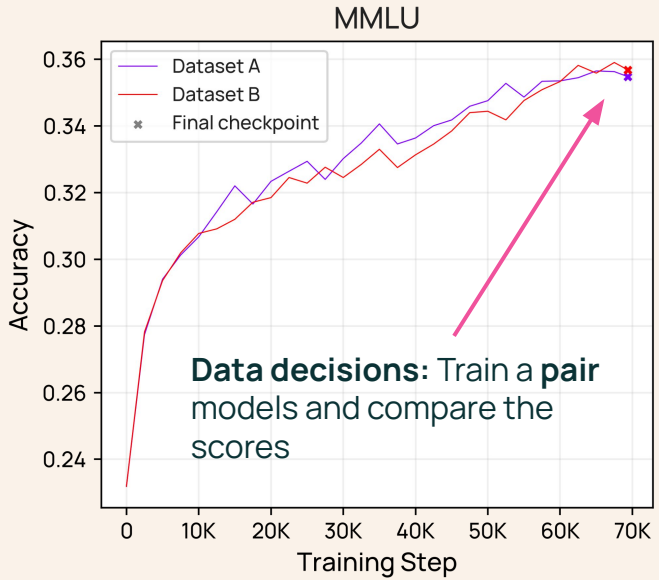
… but how to get signal for small-scale decisions?

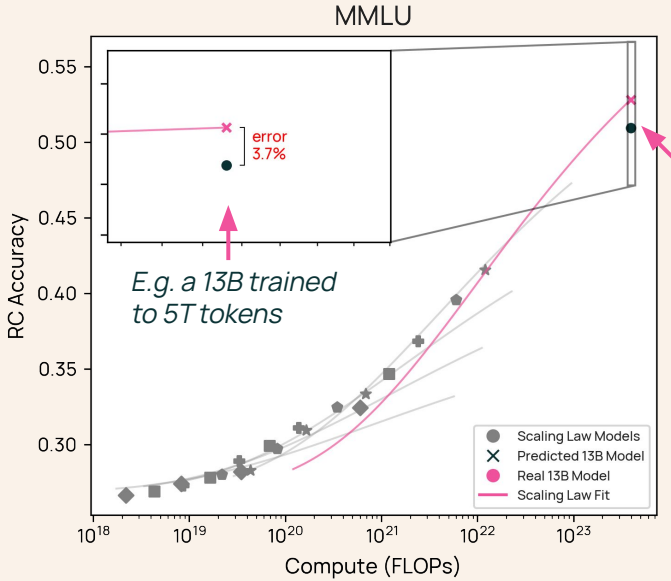




Establishing Task Scaling Laws via Compute-Efficient Model Ladders (COLM 2024)

Making decisions in pretraining == extrapolating eval

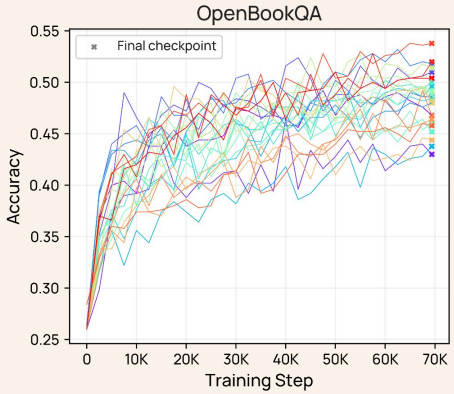


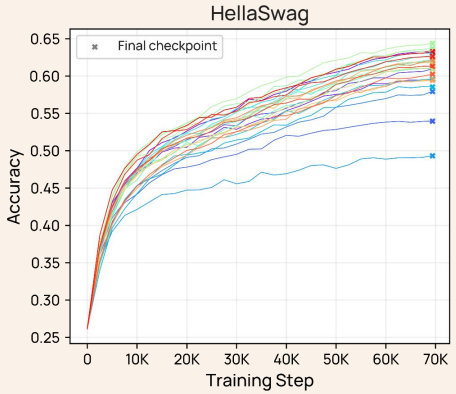


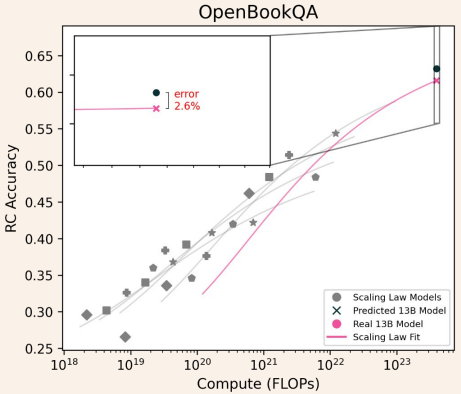
Scaling laws: Train many **small** models and extrapolate the performance

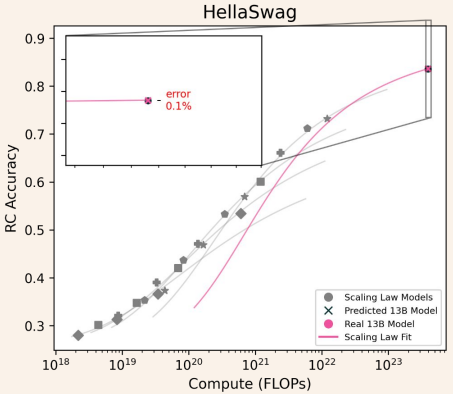
DataDecide: How to Predict Best Pretraining Data with Small Experiments (ICML, 2025)
Establishing Task Scaling Laws via Compute-Efficient Model Ladders (COLM, 2025)

Ai2



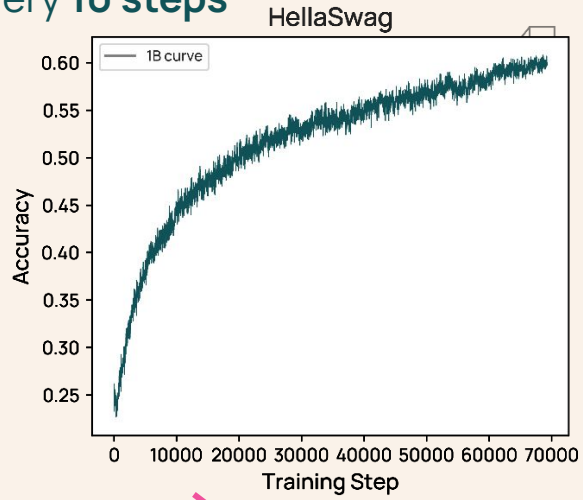


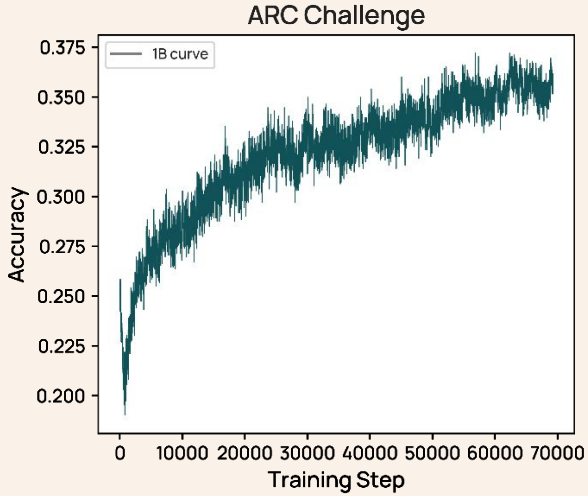


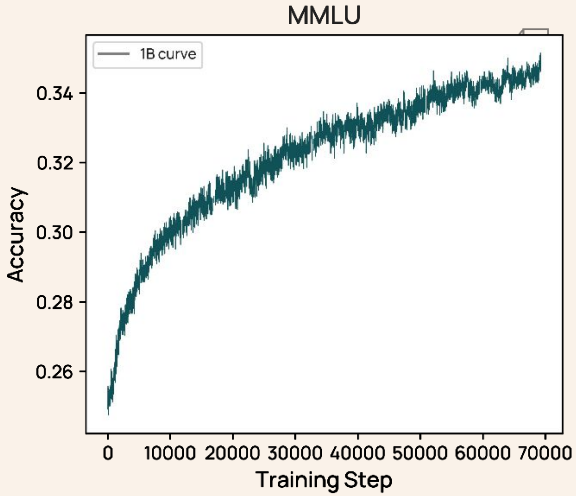


Why do so many predictions fail - but some don't?

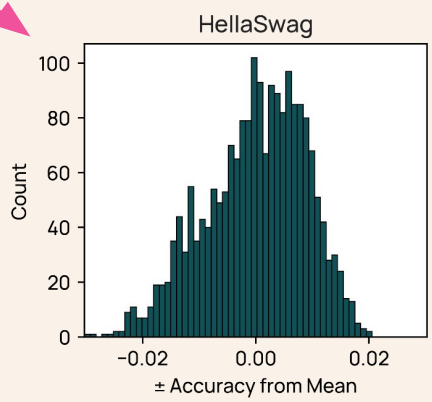
1B-5xC + eval every **10 steps**

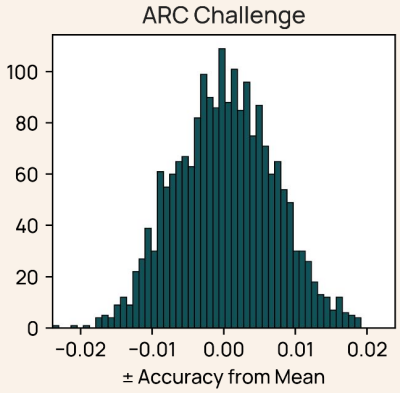


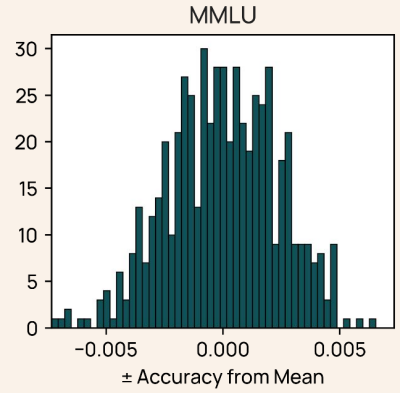




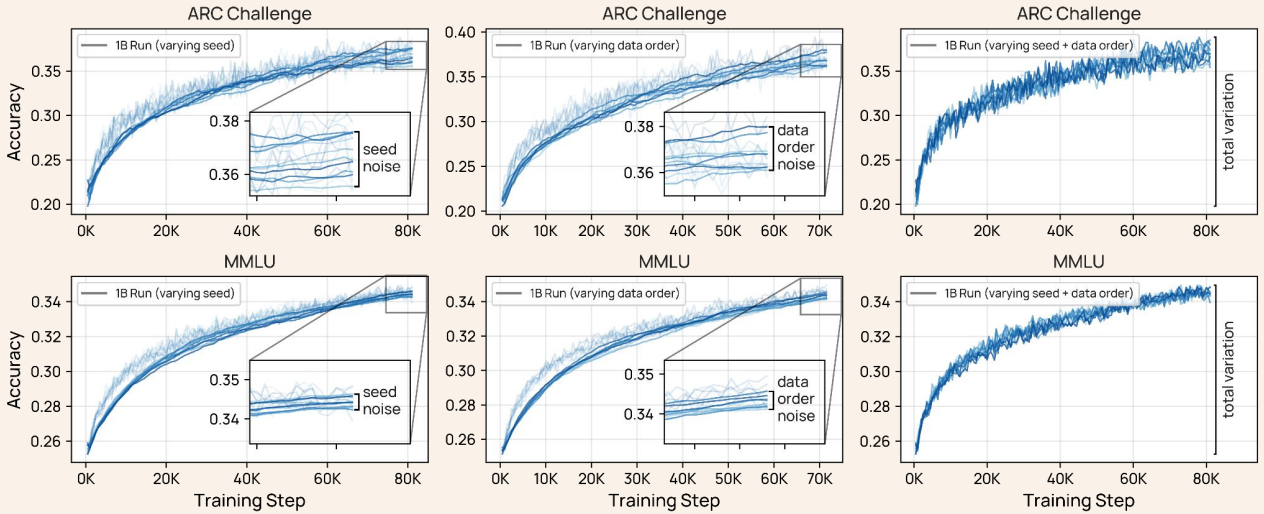
Final 20% of checkpoints





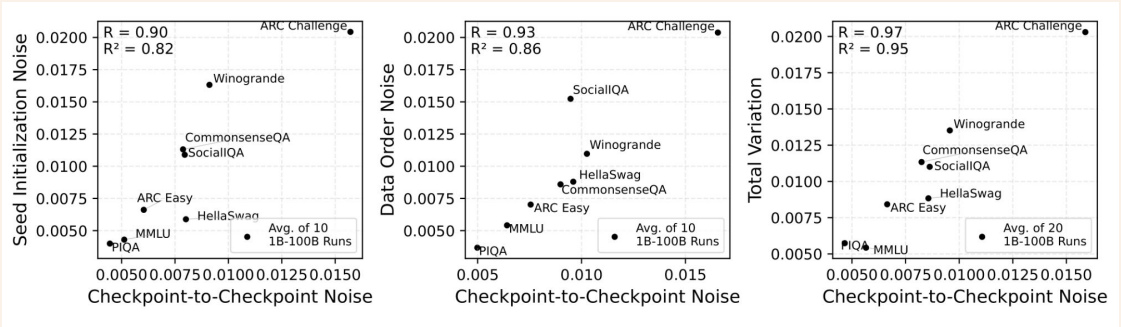


Many sources of noise ….



10 1B-5xC models varying: **data order** and **random seed**

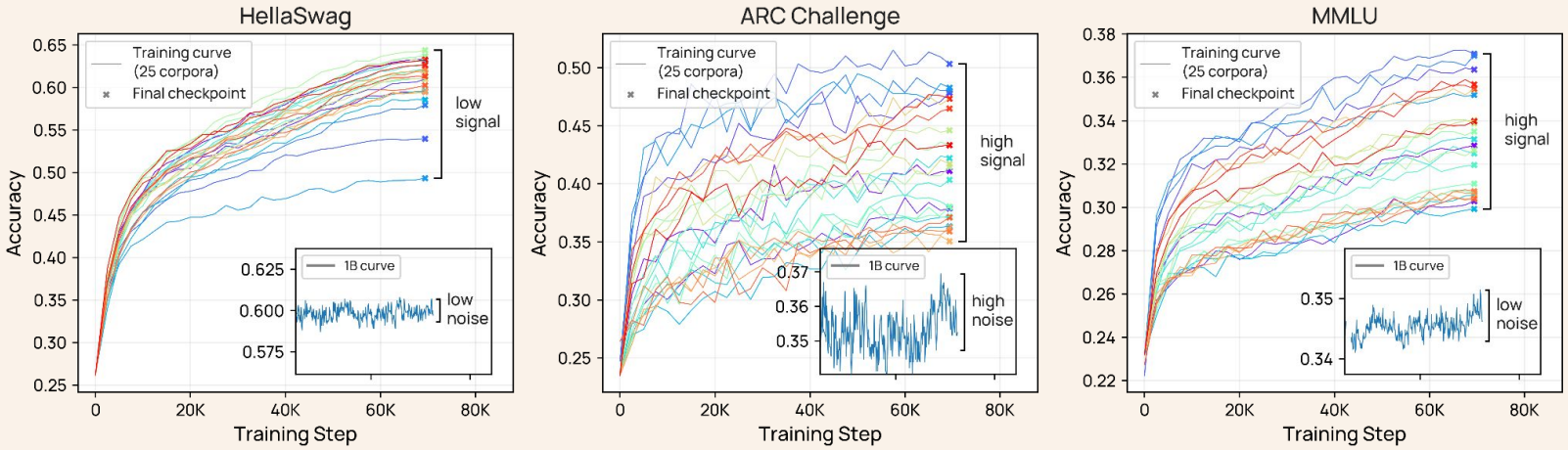
… seed noise correlates with step-to-step noise!



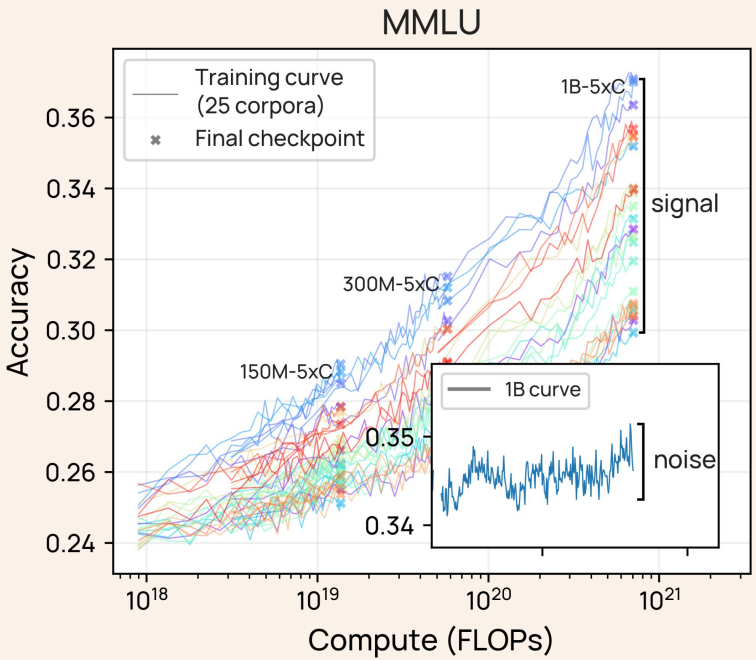
+ Total variation correlates w/ step-to-step noise

Ai2

Inter-checkpoint variance is not the whole story! We need to measure both signal and noise



Ai2



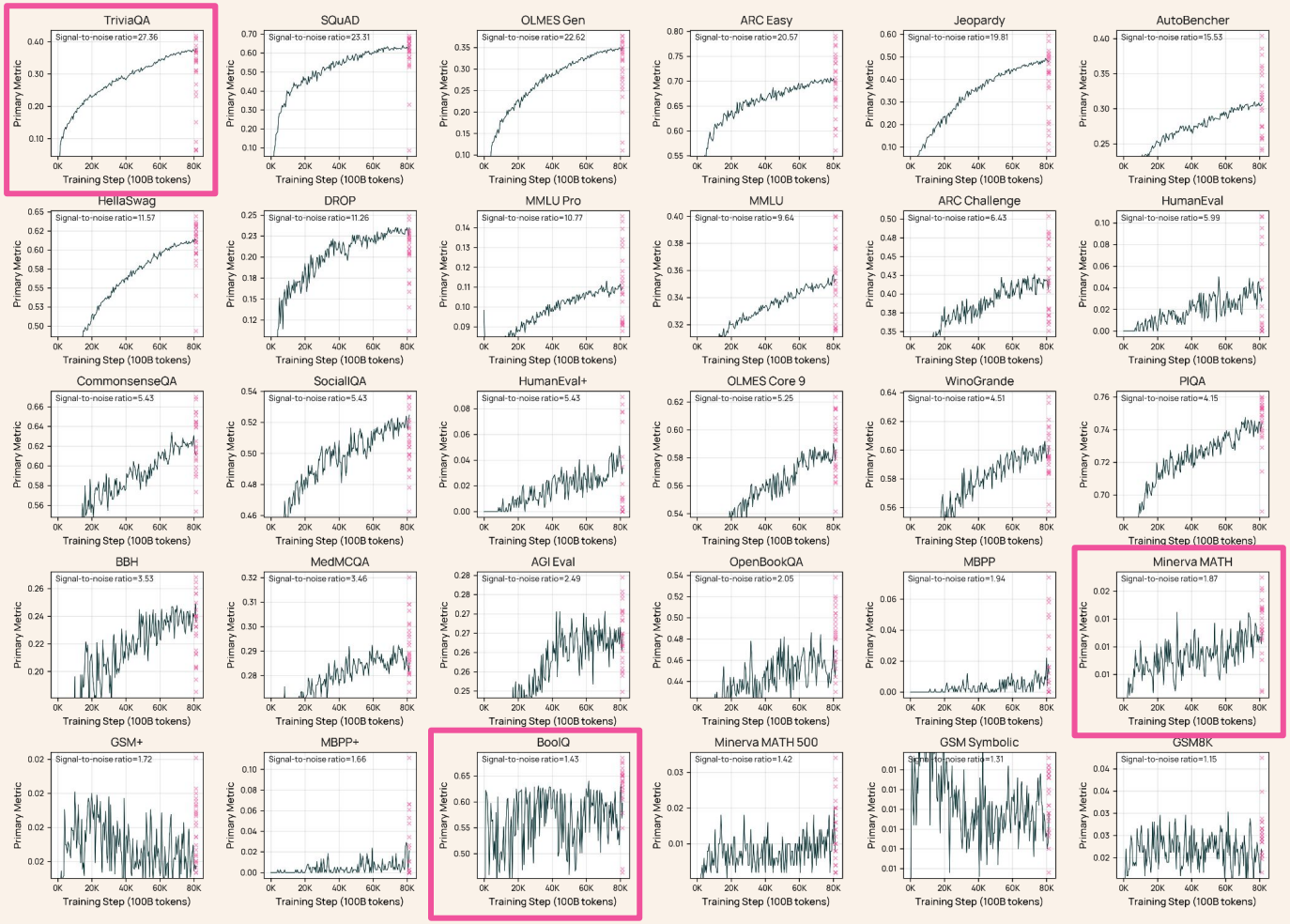
Signal (max difference between model scores):

$$\text{Rel. Dispersion}(M) = \max_{j,k} |m_j - m_k| / \bar{m}$$

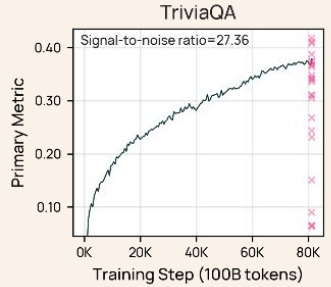
Noise (ckpt-to-ckpt noise using last n steps):

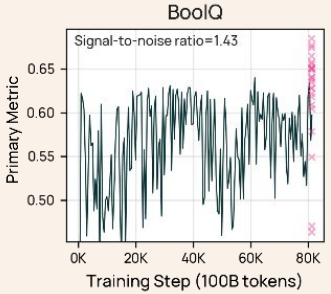
$$\text{Rel. Std.}(m) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (m_i - \bar{m})^2} / \bar{m}$$

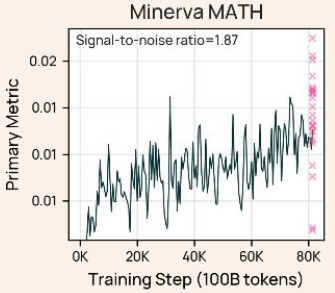
$$\text{Signal-to-Noise Ratio} = \frac{\text{Rel. Dispersion(final train checkpoint)}}{\text{Rel. Std.(final } n \text{ train checkpoints)}}$$



← SNR at the 1B scale

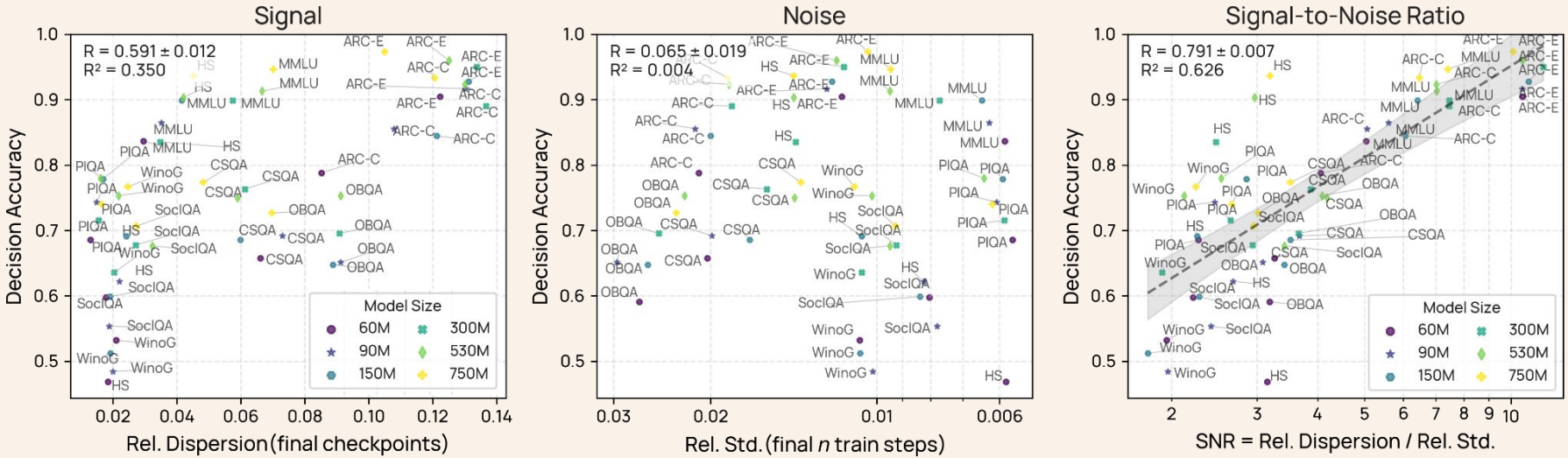






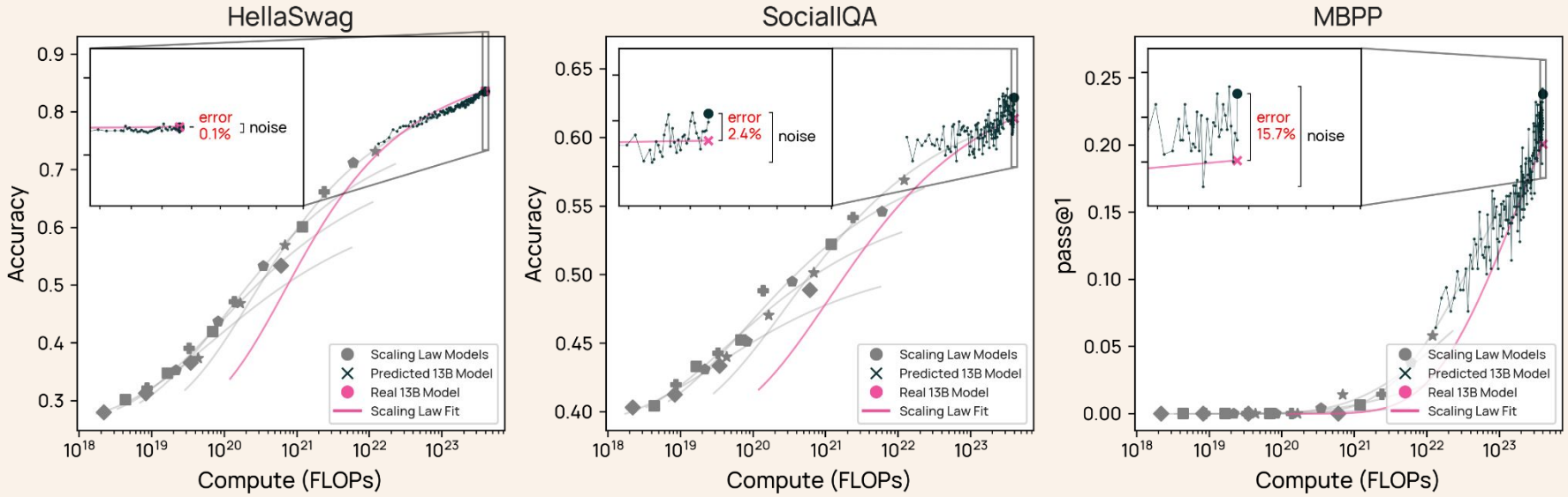
Ai2

Only signal or noise alone do not explain rank agreement from small to large scale… … but the signal-to-noise ratio does!



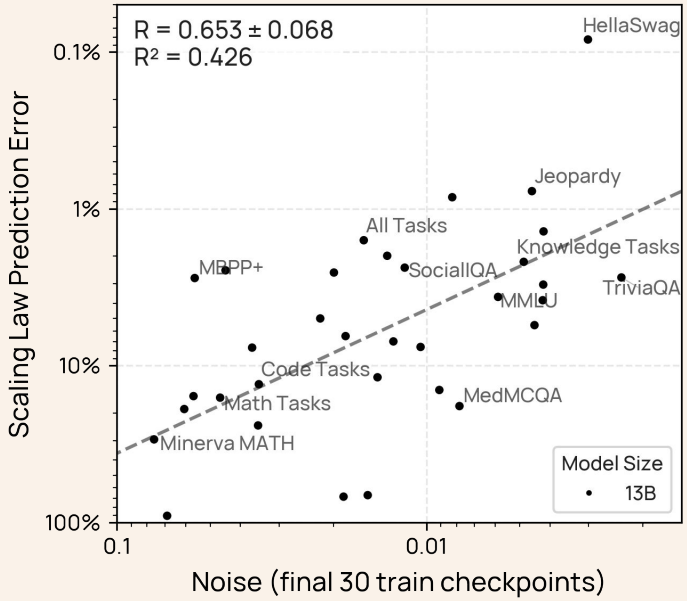
Ai2

Predicting task performance using scaling laws is sensitive to noise!

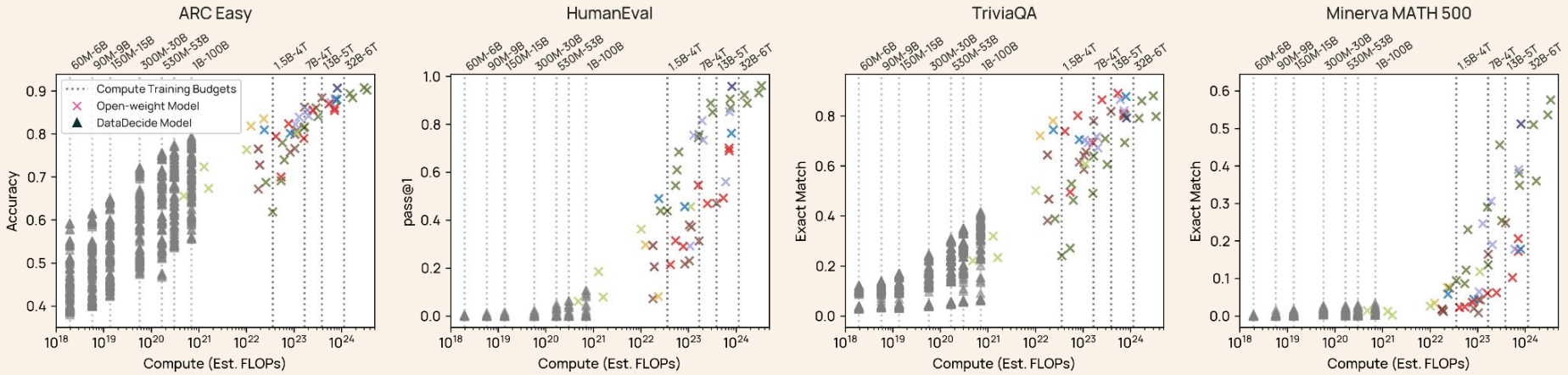


Ai2

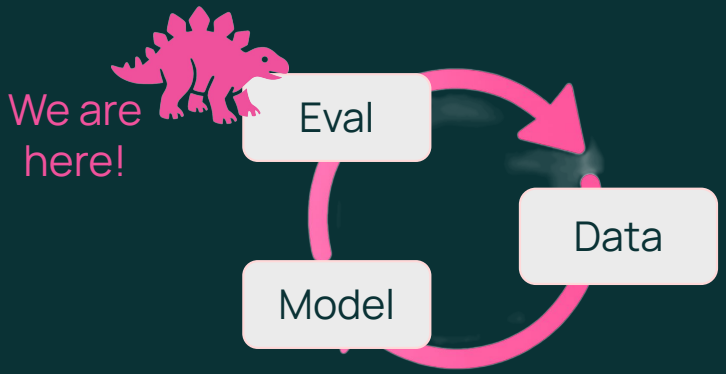
Predicting task performance using scaling laws is sensitive to noise!



Ai2



The signal-to-noise ratio is specifically useful as signal of benchmarks changes for large train compute budgets



Olmo 3

Olmo Team*

Allyson Ettinger[♥1] Amanda Bertsch[♥1,3] Bailey Kuehl[♥1] David Graham[♥1]
David Heineman[♥1] Dirk Groeneveld[♥1] Faeze Brahman[♥1] Finbarr Timbers[♥1]
Hamish Ivison[♥1,2] Jacob Morrison[♥1,2] Jake Poznanski[♥1] Kyle Lo[♥1,2] Luca Soldaini[♥1]
Matt Jordan[♥1] Mayee Chen[♥1,4] Michael Noukhovitch[♥1,5,6] Nathan Lambert[♥1]
Pete Walsh[♥1] Pradeep Dasigi[♥1] Robert Berry[♥1] Saumya Malik[♥1] Saurabh Shah[♥1]
Scott Geng[♥1,2] Shane Arora[♥1] Shashank Gupta[♥1] Taira Anderson[♥1] Teng Xiao[♥1]
Tyler Murray[♥1] Tyler Romero[♥1] Victoria Graf[♥1,2]

Akari Asai[1,3] Akshita Bhagia[1] Alexander Wettig[7] Alisa Liu[2] Aman Rangapur[1]
Chloe Anastasiades[1] Costa Huang[1] Dustin Schwenk[1] Harsh Trivedi[1] Ian Magnusson[1,2]
Jaron Lochner[1] Jiacheng Liu[1] Lester James V. Miranda[1] Maarten Sap[1,3] Malia Morgan[1]
Michael Schmitz[1] Michal Guerquin[1] Michael Wilson[1] Regan Huff[1] Ronan Le Bras[1]
Rui Xin[2] Rulin Shao[2] Sam Skjonsberg[1] Shannon Zejiang Shen[8] Shuyue Stella Li[2]
Tucker Wilde[1] Valentina Pyatkin[1] Will Merrill[1] Yapei Chang[3] Yuling Gu[1] Zhiyuan Zeng[1,2]

Ashish Sabharwal[1] Luke Zettlemoyer[2] Pang Wei Koh[1,2]
Ali Farhadi[1,2] Noah A. Smith[♥1,2] Hannaneh Hajishirzi[♥1,2]

[1]Allen Institute for AI [2]University of Washington [3]Carnegie Mellon University [4]Stanford University [5]Mila
[6]Université de Montréal [7]Princeton University [8]Massachusetts Institute of Technology [9]University of Maryland

*Olmo 3 was a team effort; authors sorted alphabetically. ♥marks core contributors. See author contributions here.

- **Olmo 3 Base:** `Olmo-3-1025-7B` `Olmo-3-1125-32B`
- **Olmo 3 Think:** `Olmo-3-7B-Think` `Olmo-{3|3.1}-32B-Think`
- **Olmo 3 Instruct:** `Olmo-3-7B-Instruct` `Olmo-3.1-32B-Instruct`
- **Olmo 3 RL Zero:** `Olmo-3-7B-RL-Zero-{Math|Code|IF|General|Mix}` `Olmo-3.1-7B-RL-Zero-{Math|Code}`

- **Base Data:** `Pretrain: Dolma 3 Mix` `Midtrain: Dolma 3 Dolmino Mix` `Long-ctx: Dolma 3 Longmino Mix`
- **Think Data:** `Dolci-Think-{SFT|DPO|RL}-7B` `Dolci-Think-{SFT|DPO|RL}-32B`
- **Instruct Data:** `Dolci-Instruct-{SFT|DPO|RL}`
- **RL-Zero Data:** `Dolci-RL-Zero-{Math|Code|IF|General}-7B` `Dolci-RL-Zero-Mix-7B`

3.3 Experimental Design and Evaluation

Model development requires many iterative data and training decisions. However, benchmarks are not perfect decision-making tools: different evaluations are only sensitive for making development decisions across specific ranges of scale and capability (Magnusson et al., 2025). Models trained at small compute scales are known to exhibit random-chance performance on math, code, and multiple-choice question answering (MCQA) tasks (Wei et al., 2022; Gu et al., 2024b), and benchmark noise can reduce the ability to trust small differences in scores (Heineman et al., 2025). To address these problems, we develop **OlmoBaseEval**, a collection of benchmark suites to support decision-making during base model development. OlmoBaseEval features the

parameter scales. Olmo 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire **model flow**, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, **Olmo 3.1 Think 32B**, is the strongest fully-open thinking model released to-date.

Ai2

OLMo 2 → Olmo 3 base evals:

+ capabilities (math, code)
+ coverage of science QA, Basic Skills
+ coverage across formats (Gen2MC)

3 new methods for Olmo 3 Base eval:

(1) clustering tasks, (2) scaling analysis,
(3) signal-to-noise analysis

task	split	# inst (total)	# shots	metric	reference
Multiple-choice tasks					
ARC-Challenge (ARC_C)	Test	1172	5	pmi	(Clark et al., 2018)
BoolQ	Val	1000 (3270)	5	none	(Clark et al., 2019)
HellaSwag (HSwag)	Val	1000 (10042)	5	char	(Zellers et al., 2019)
MMLU†	Test	14042	5	char	(Hendrycks et al., 2021a)
WinoGrande (WinoG)	Val	1267	5	none	(Sakaguchi et al., 2020)
Generative tasks					
DROP	Val	1000 (9536)	5	F1	(Dua et al., 2019)
Natural Questions (NatQs)	Val	1000 (3610)	5	F1	(Kwiatkowski et al., 2019)
Held-out tasks					
AGIEval English	Test	2646	1	MCF	(Zhong et al., 2024)
GSM8K	Test	1319	8 (CoT)	EM	(Cobbe et al., 2021)
MMLU-Pro	Test	12032	5	MCF	(Wang et al., 2024)
TriviaQA	Val	7993	5	F1	(Joshi et al., 2017)

Table 20 Details of OLMES benchmarks used in OLMo 2 evaluation, with standardized choices of dataset split, number of instances to use, along with total number if sampling was used. For multiple-choice tasks, when using the Cloze/Completion Formulation (CF), the "metric" column specifies which normalization scheme to use. Following the OLMES standard, we evaluate each model using both the MCF (Multiple-Choice Formulation) and CF formulations, and the best performing one is used. For efficiency reasons, we limit MMLU and held-out multiple-choice evaluations to MCF only as all the relevant models strongly prefer that format for these tasks.

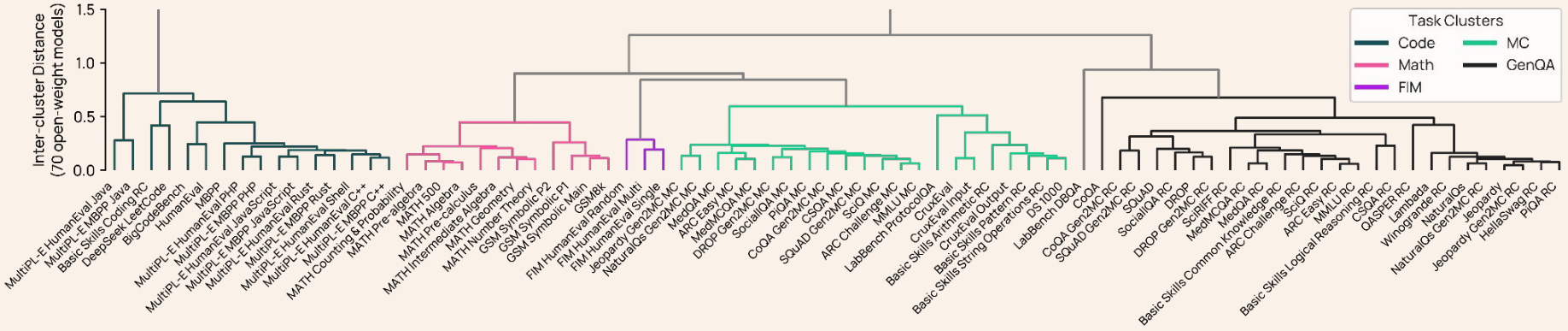
	task	ICL	format	metric	temp	top-p	max toks	p@k (n)	# sub
Base Main Suite									
Math	GSM8K* (2021)	8^{α}	CoT EM	pass@k	0.6	0.6	512	1, 4 (8)	-
	GSM Symbolic* (2024)	8^{α}	CoT EM	pass@k	0.6	0.6	512	1, 4 (8)	3
	Minerva MATH* (2022)	4^{α}	CoT EM	pass@k	0.6	0.6	1024	1, 4 (4)	7
	MATH 500* (2022; 2023)	4^{α}	CoT EM	pass@k	0.6	0.6	1024	1, 16 (32)	-
Code	HumanEval* (2021)	3	Code Exec	pass@k	0.6	0.6	512	1, 16 (32)	-
	MBPP* (2021)	3	Code Exec	pass@k	0.6	0.6	512	1, 16 (32)	-
	BigCodeBench* (2024)	3	Code Exec	pass@k	0.6	0.6	1280	1 (5)	-
	DS 1000* (2022)	3	Code Exec	pass@k	0.6	0.6	1024	1 (5)	-
	Deepseek LeetCode* (2024)	0	Code Exec	pass@k	0.6	0.6	512	1, 16 (32)	-
	MultiPL-E HumanEval* (2022)	0	Code Exec	pass@k	0.6	0.6	1024	1, 16 (32)	6
	MultiPL-E MBPP* (2022)	0	Code Exec	pass@k	0.6	0.6	1024	1, 16 (32)	6
FIM	HumEval FIM Single* (2022)	0	FIM	pass@1	0.8	0.95	512	1 (10)	-
	HumEval FIM Random* (2022)	0	FIM	pass@1	0.8	0.95	512	1 (5)	-
	HumEval FIM Multi* (2022)	0	FIM	pass@1	0.8	0.95	512	1 (1)	-
STEM QA	ARC (2018)	5	MC	Acc	-	-	-	-	2
	MMLU STEM (2021b)	5	MC	Acc	-	-	-	-	19
	MedMCQA* (2022)	5	MC	Acc	-	-	-	-	-
	MedQA* (2021)	5	MC	Acc	-	-	-	-	-
	SciQ* (2017)	5	MC	Acc	-	-	-	-	-
Non-STEM QA	MMLU Humanities (2021b)	5	MC	Acc	-	-	-	-	13
	MMLU Social Sci. (2021b)	5	MC	Acc	-	-	-	-	12
	MMLU Other (2021b)	5	MC	Acc	-	-	-	-	14
	CSQA (2019)	5	MC	Acc	-	-	-	-	-
	PiQA (2020)	5	MC	Acc	-	-	-	-	-
	SocialIQA (2019)	5	MC	Acc	-	-	-	-	-
	DROP Gen2MC* (§A.3.2; 2019)	5	MC	Acc	-	-	-	-	-
	Jeopardy Gen2MC* (§A.3.2; 2024)	5	MC	Acc	-	-	-	-	-
	NaturalQs Gen2MC* (§A.3.2; 2019)	5	MC	Acc	-	-	-	-	-
	SQuAD Gen2MC* (§A.3.2; 2016)	5	MC	Acc	-	-	-	-	-
	CoQA Gen2MC* (§A.3.2; 2019)	$0^{\dagger}$	MC	Acc	-	-	-	-	-
	Basic Skills* (§A.3.2)	5	MC	Acc	-	-	-	-	6
GenQA	HellaSwag (2019)	5	$RC_{per\text{-}char}$	Acc	-	-	-	-	-
	WinoGrande (2020)	5	RC_{none}	Acc	-	-	-	-	-
	Lambada (2016)	0	$RC_{per\text{-}char}$	Acc	-	-	-	-	-
	Basic Skills* (§A.3.2)	5	$RC_{per\text{-}token}$	Acc	-	-	-	-	6
	DROP (2019)	5	GenQA	F1	0	1	100	-	-
	Jeopardy (2024)	5	GenQA	F1	0	1	50	-	-
	NaturalQs (2019)	5	GenQA	F1	0	1	50	-	-
	SQuAD (2016)	5	GenQA	F1	0	1	50	-	-
	CoQA (2019)	$0^{\dagger}$	GenQA	F1	0	1	50	-	-
Base Held-out Suite									
	MMLU Pro (2024a)	5	MC	Acc	-	-	-	-	13
	LBPP* (2024)	0	Code Exec	pass@k	0.6	0.6	4096	1 (32)	-
	Deepmind Math* (2019)	5	CoT EM	pass@k	0.6	0.6	2048	1 (1)	-
	BigBench Hard (2022)	3	CoT EM	Acc	0.6	0.6	512	1 (1)	55

Table 43 Details of the OLMO 3 base evaluation suite. Tasks were formatted as multiple-choice (MC), rank choice (RC, following the setup in Gu et al. (2024b)), short-form generative (GenQA), chain-of-thought with exact-match scoring (CoT EM), code exeuction (Code Exec) or fill-in-the-middle coding (FIM). * = new additions to the base OLMo 2 suite (OLMo et al., 2024); † = few-shot examples are built-in the task; $^{\alpha}$ = human-written few-shot examples.

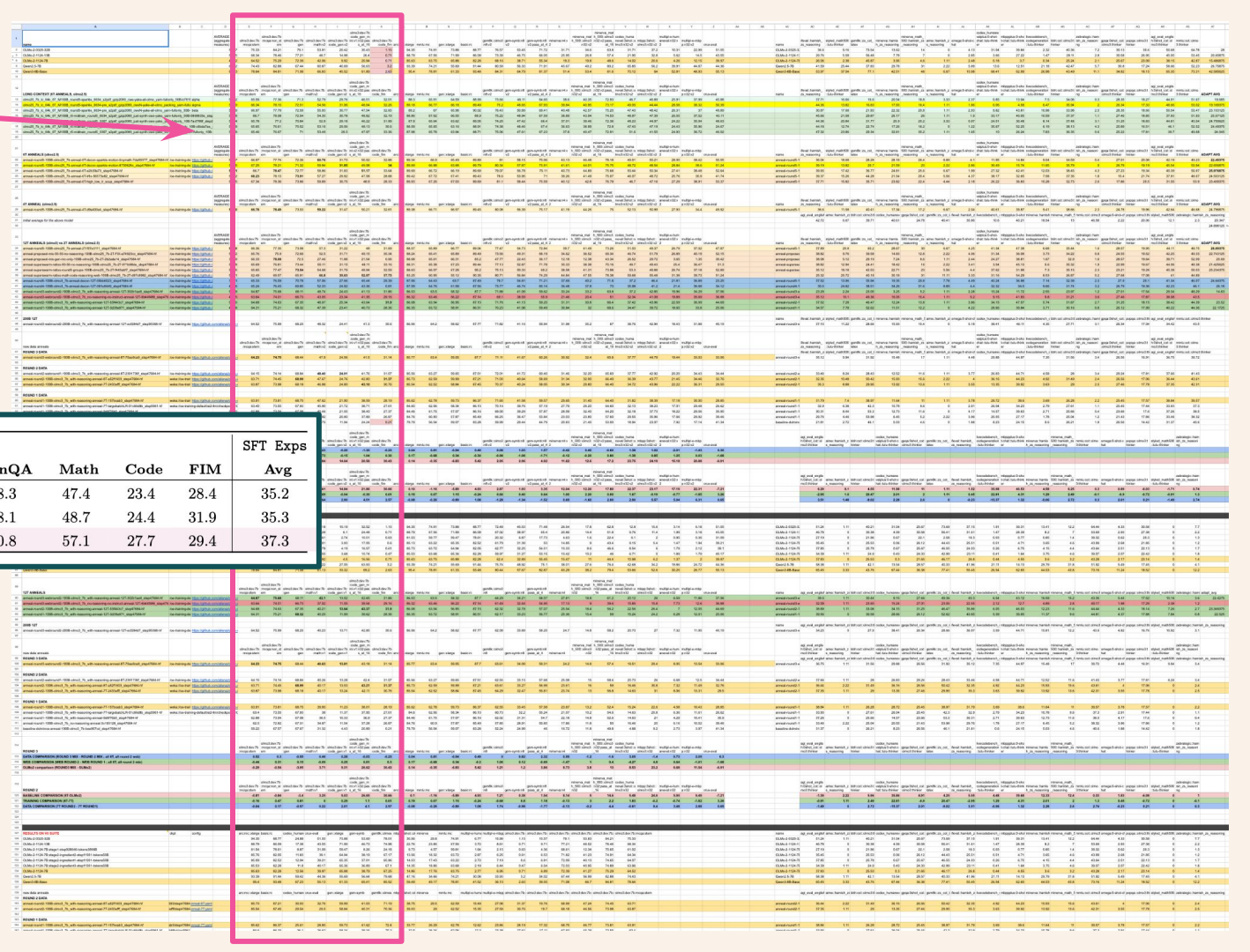
Ai2

1

How to handle large # of benchmarks? Group tasks into "clusters" (Math, Code, GenQA) and track multi-task averages



Task averages simplify decision making (e.g. in midtraining)

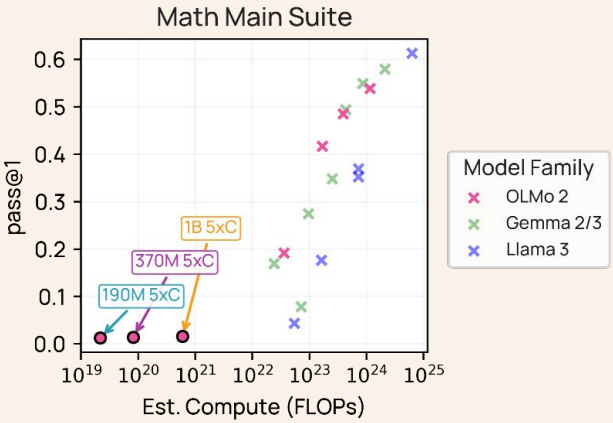


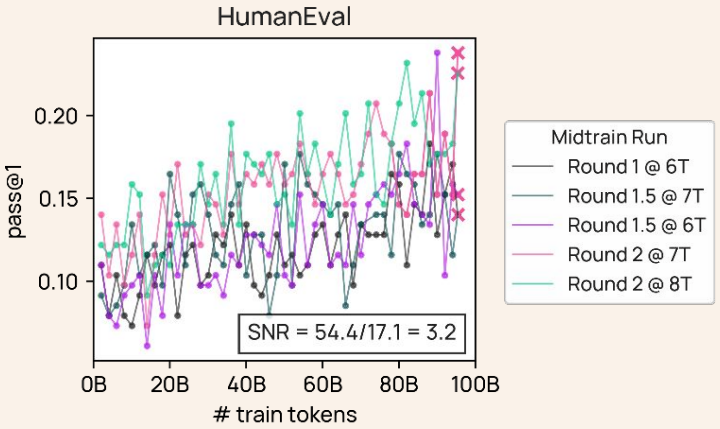
	OLMoBaseEval						SFT Exps
Mix	Avg	MC STEM	MC Non-STEM	GenQA	Math	Code	FIM	Avg
Round 1	49.7	64.3	75.2	68.3	47.4	23.4	28.4	35.2
Round 3	50.7	64.9	75.7	68.1	48.7	24.4	31.9	35.3
Round 5	53.1	65.3	76.1	70.8	57.1	27.7	29.4	37.3

Ai2

2 + 3

Two additional problems:



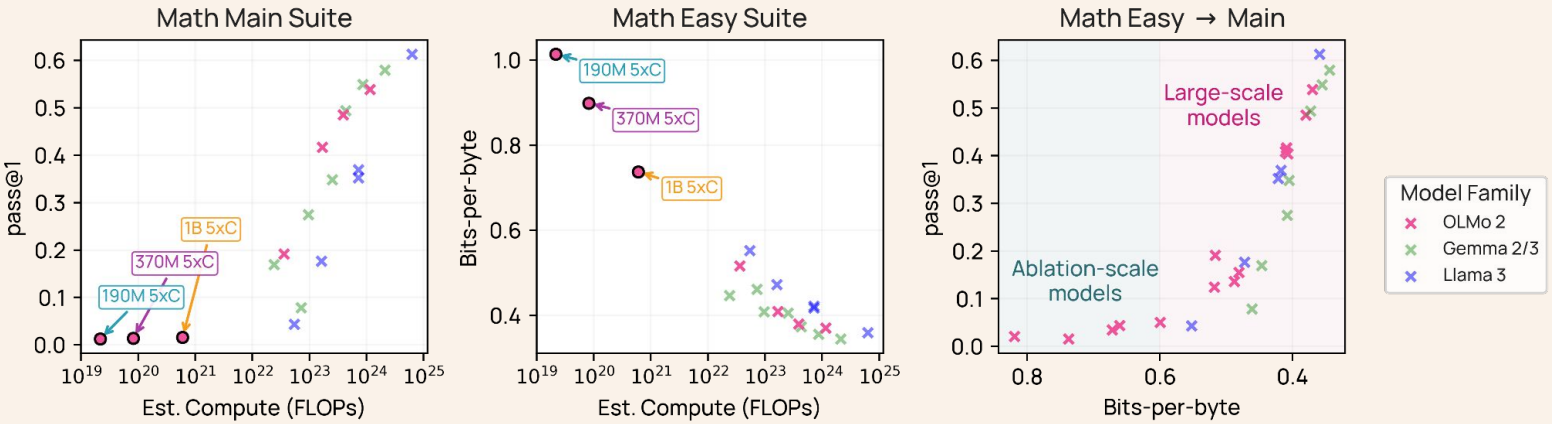


Signal at small scales
(e.g. 30M to 1B)

Noise at large scales
(e.g.midtraining)

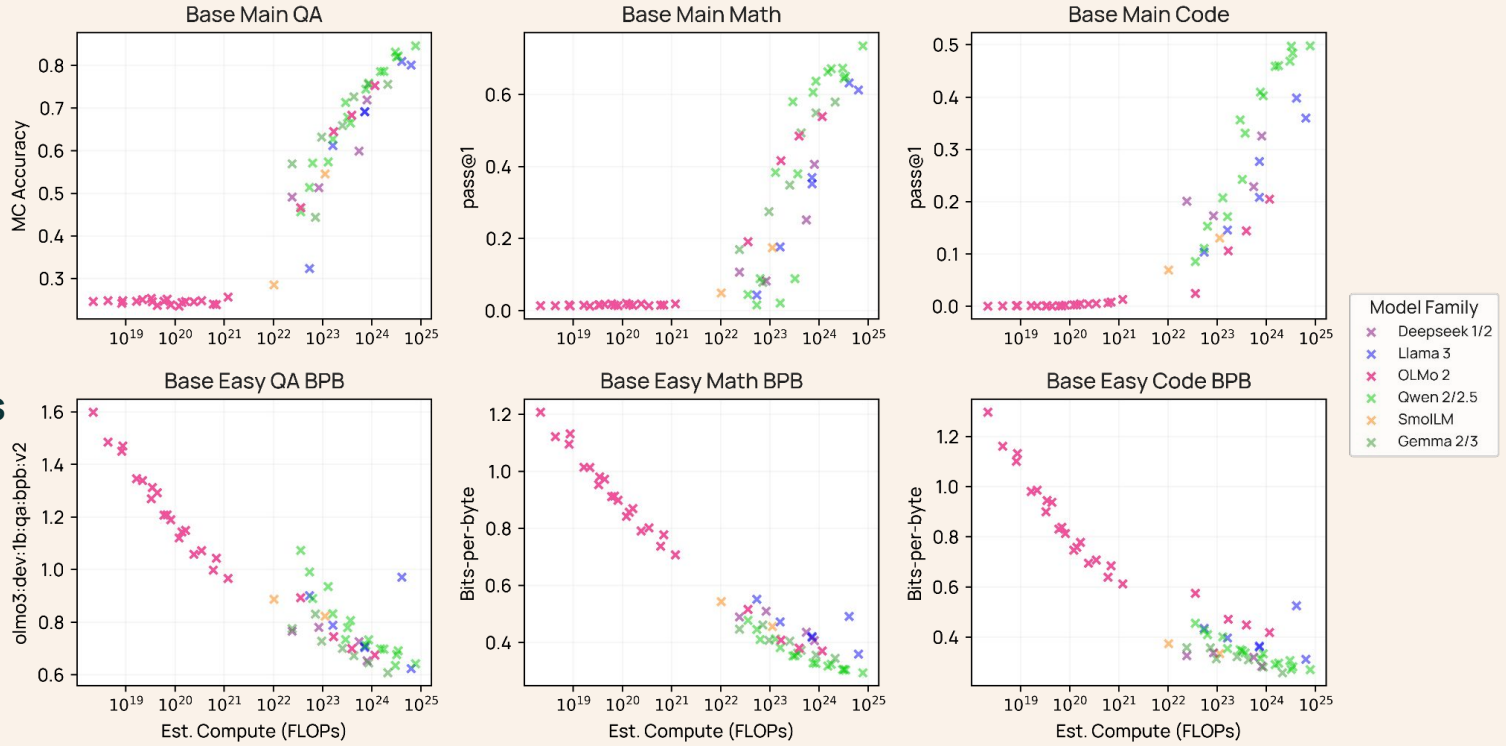
Ai2

2 Small-scale metrics:



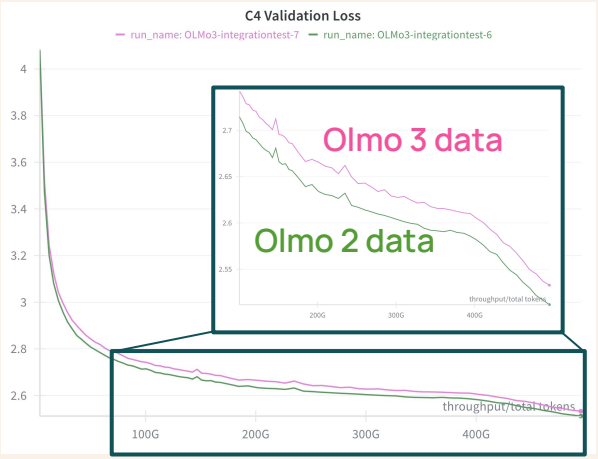
2 Small-scale metrics:

Separate decision-making suite using BPB over continuations

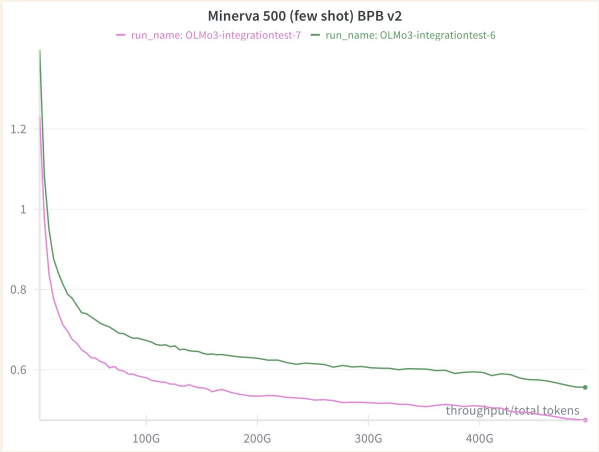


2 Small-scale metrics:

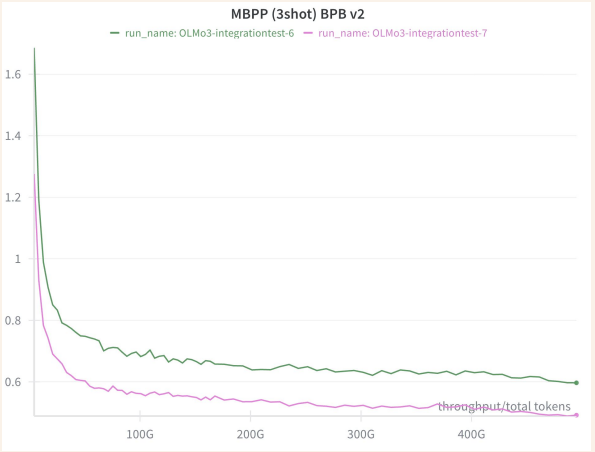
C4 Loss



Math 500 BPB



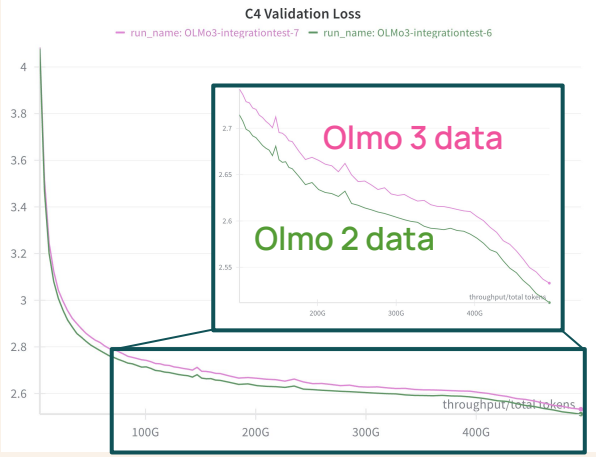
MBPP BPB (Python Coding)

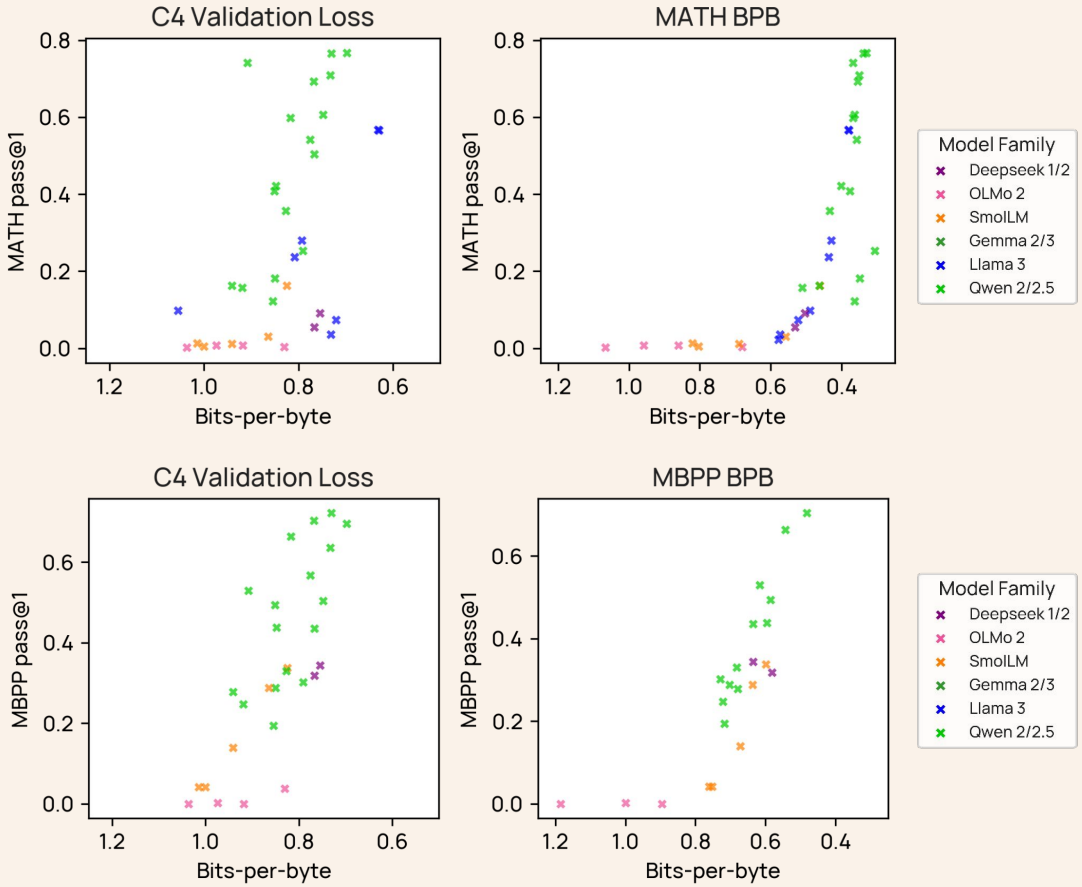


At 7B 500B tokens, Olmo 3 data has a worse fit than Olmo 2 data on C4 loss, but better on math, code text

2 Small-scale metrics:

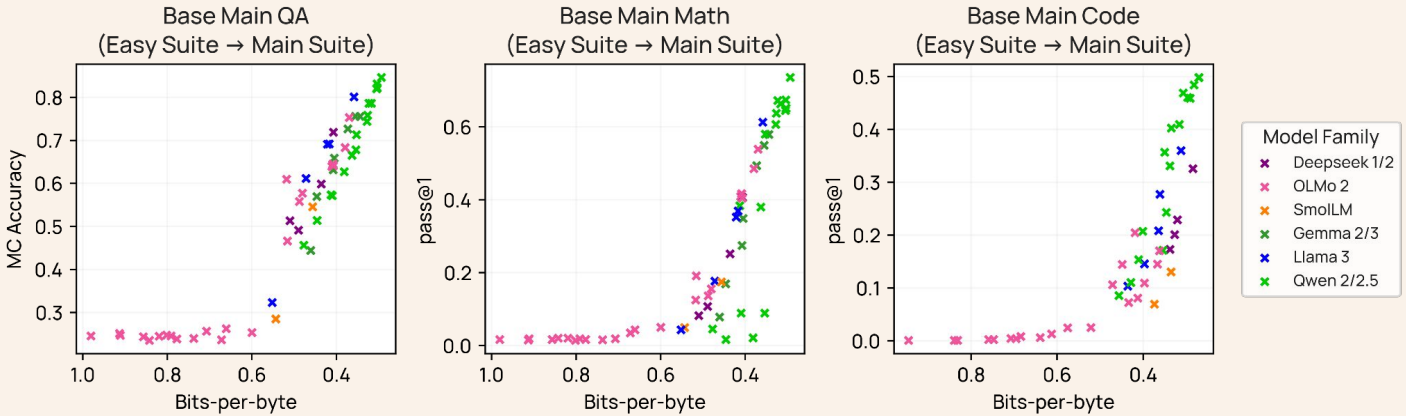
C4 Loss



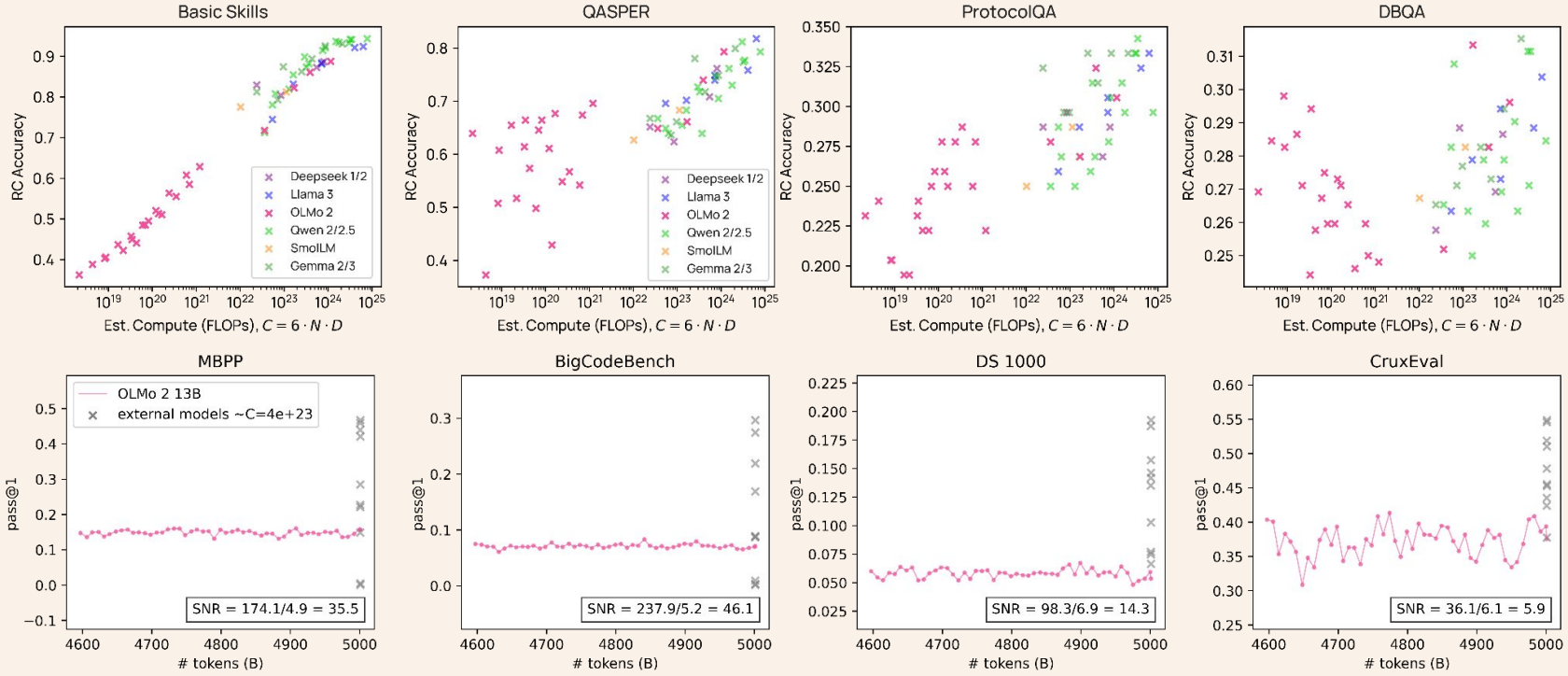


Ai2

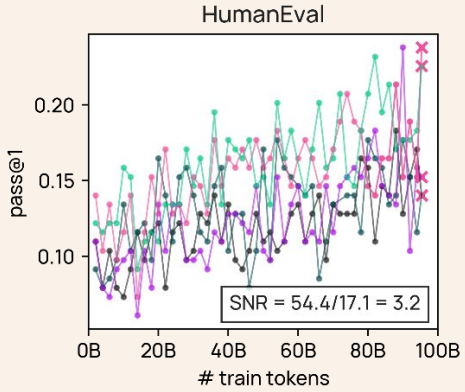
2 Small-scale metrics:

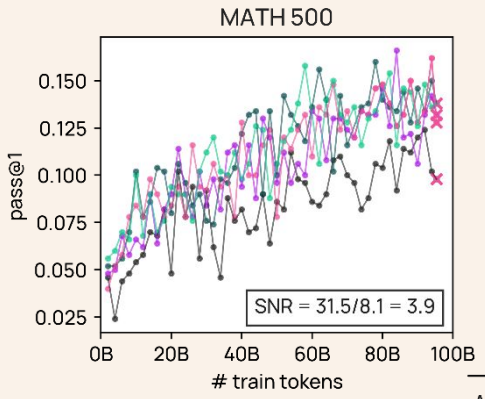


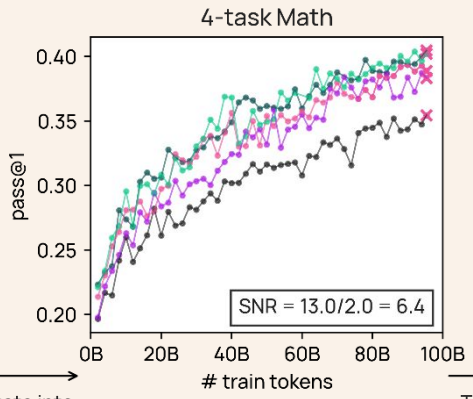
3 SNR in Olmo 3 Eval

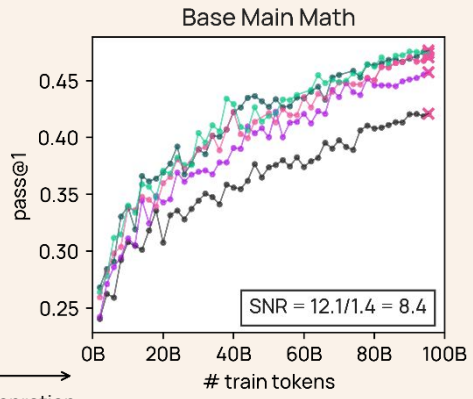


3 SNR in Olmo 3 Eval





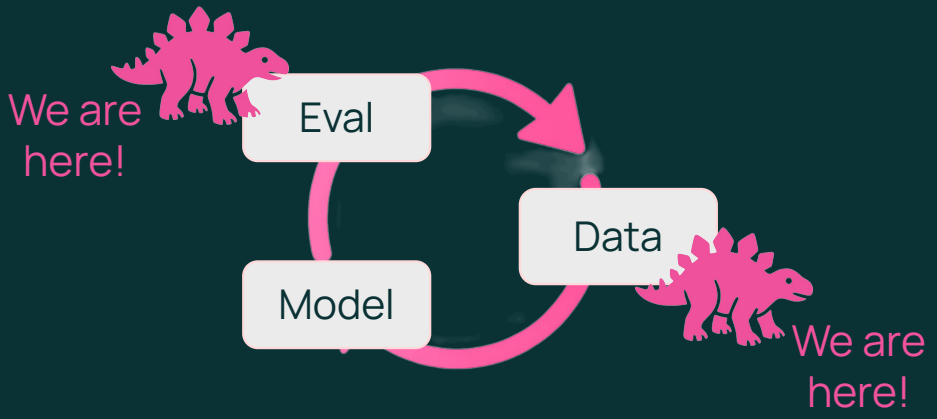




Aggregate into multi-task average, filter noisy tasks

Tune genration configuration

Ai2



Olmix: A Framework for Data Mixing Throughout LM Development

Mayee F. Chen[1,2] Tyler Murray[1] David Heineman[1] Matt Jordan[1] Hannaneh Hajishirzi[1,3]
Christopher Ré[2] Luca Soldaini[1] Kyle Lo[1,3]

[1]Allen Institute for AI [2]Stanford University [3]University of Washington

Code: Olmix Data: Olmix Contact: mfchen@cs.stanford.edu {lucas,kylel}@allenai.org

Abstract

Ai2

Data mixing—determining the ratios of data from different domains—is a first-order concern for training language models (LMs). While existing mixing methods show promise, they fall short when applied during real-world LM development. We present Olmix, a framework that addresses two such challenges. First, the configuration space for developing a mixing method is not well understood—design choices across existing methods lack justification or consensus and overlook practical issues like data constraints. We conduct a comprehensive empirical study of this space, identifying which design choices lead to a strong mixing method. Second, in practice, the domain set evolves throughout LM development as datasets are added, removed, partitioned, and revised—a problem setting largely unaddressed by existing works, which assume fixed domains. We study how to efficiently recompute the mixture after the domain set is updated, leveraging information from past mixtures. We introduce mixture reuse, a mechanism that reuses existing ratios and recomputes ratios only for domains affected by the update. Over a sequence of five domain-set updates mirroring real-world LM development, mixture reuse matches the performance of fully recomputing the mix after each update with 74% less compute and improves over training without mixing by 11.6% on downstream tasks.

1 Introduction

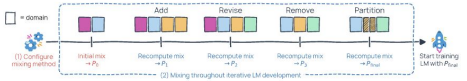


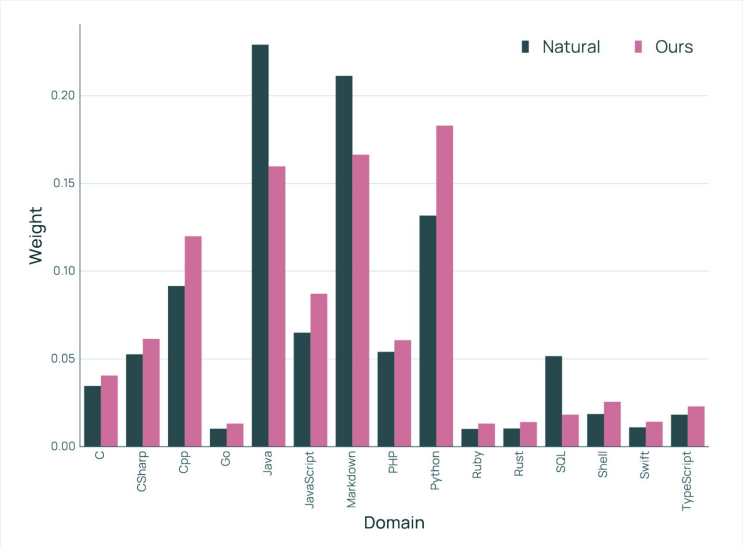
Figure 1 Two problems with data mixing encountered during LM development: (1) How to best configure your mixing method? (2) How to efficiently mix under evolving domain sets?

Modern language models (LMs) are trained on datasets composed of many domains, such as web text, code, and PDFs. The composition of these domains is crucial for strong downstream performance, making data mixing a first-order component of LM development (Grattafiori et al., 2024; Qwen et al., 2024; Olmo et al., 2025). However, finding a good mix is non-trivial: practitioners often resort to manual weight tuning or exhaustive search, which can require many training runs—possibly thousands of GPU hours—to assess performance. This has resulted in a growing literature on data mixing methods that aim to find strong mixtures systematically with less compute (Xie et al., 2023; Fan et al., 2024; Chen et al., 2025a). Many mixing methods that achieve promising results follow a common *offline mixing schema* (Liu et al., 2025b,a; Ye et al., 2025) that consists of three steps: 1. train a set of smaller proxy models on different mixtures (a "swarm"), 2.

Ai2

What data mixing?

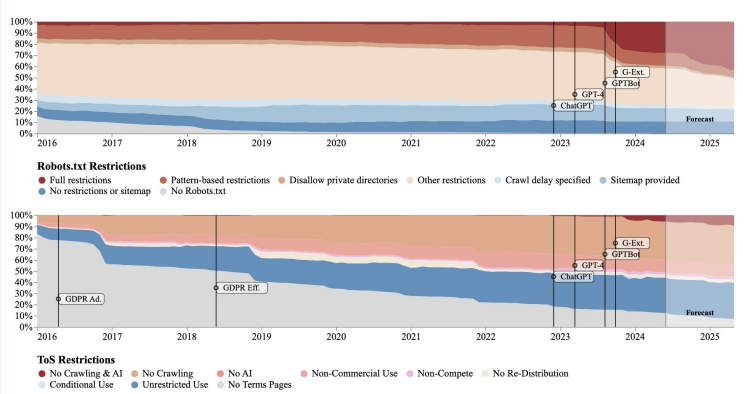
Olmo 3 (2025)



Why data mixing?

Token scale

Consent in Crisis (Neurips 2024)

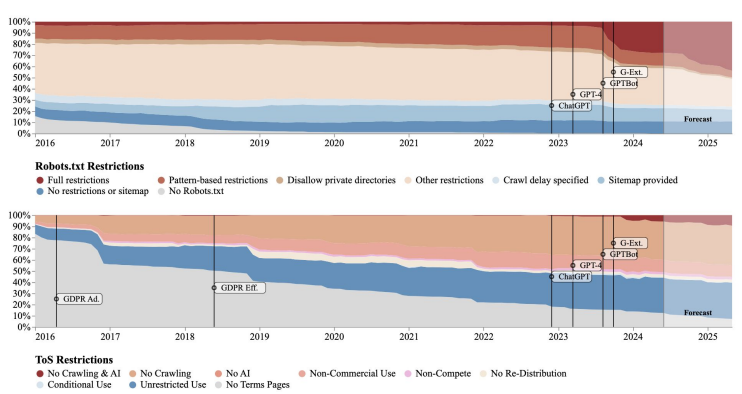


Ai2

Why data mixing?

Token scale

Consent in Crisis (Neurips 2024)



Proportion / Ratios

Gopher (2021)

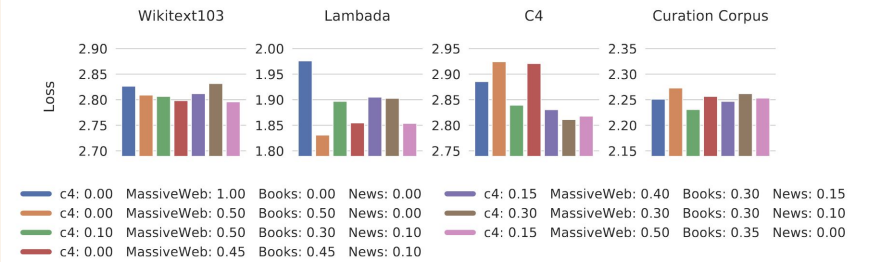
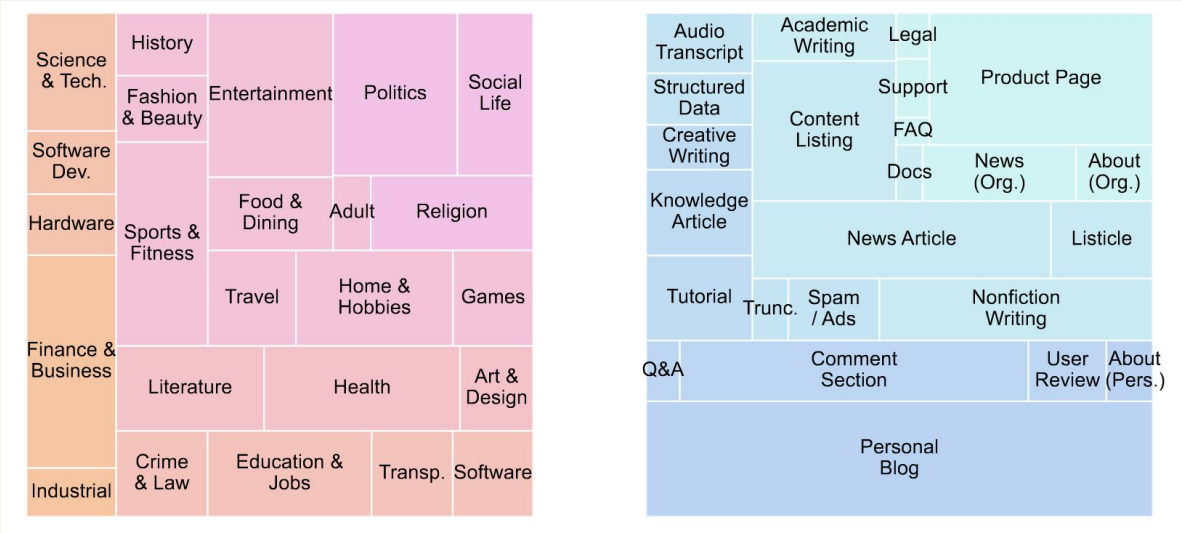


Figure A4 | **Downstream performance for different *MassiveText* subset sampling weights.** The configuration (in green) with 10% C4, 50% *MassiveWeb*, 30% Books, and 10% News performs well across all tasks and achieves the best performance on Curation Corpus—we therefore choose those sampling weights in our main *Gopher* training experiments.

Problem: Search over data mixtures is combinatorial

WebOrganizer (ICML 2024)



How data mixing?

REGMIX: DATA MIXTURE AS REGRESSION FOR LANGUAGE MODEL PRE-TRAINING

Qian Liu[1*] **Xiaosen Zheng**[2*] **Niklas Muennighoff**[3,4] **Guangtao Zeng**[5]
Longxu Dou[1] **Tianyu Pang**[1] **Jing Jiang**[2] **Min Lin**[1]
[1]Sea AI Lab [2]SMU [3]Contextual AI [4]Stanford University [5]SUTD
liuqian.sea@gmail.com; xszheng.2020@phdcs.smu.edu.sg

Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Jiasheng Ye[1,*] **Peiju Liu**[1,*] **Tianxiang Sun**[1] **Jun Zhan**[1] **Yunhua Zhou**[2,†] **Xipeng Qiu**[1,†]
{jsye23,pjliu23}@m.fudan.edu.cn zhouyunhua@pjlab.org.cn xpqiu@fudan.edu.cn
[1]Fudan University [2]Shanghai AI Laboratory

BIMIX: BIVARIATE DATA MIXING LAW FOR LANGUAGE MODEL PRETRAINING

Ce Ge & Zhijian Ma
Alibaba Group
Beijing, China
{gece.gc,zhijian.mzj}@alibaba-inc.com

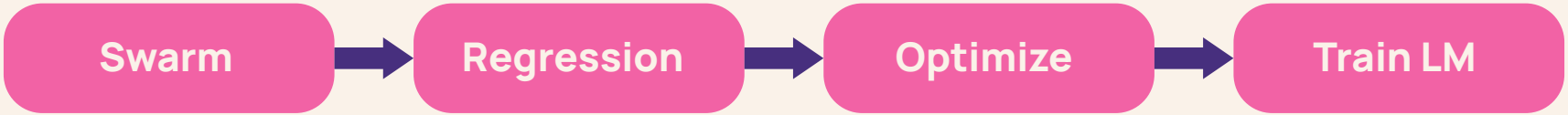
Dayuan Chen
Alibaba Group
Hangzhou, China
daoyuanchen.cdy@alibaba-inc.com

Yaliang Li* **& Bolin Ding**
Alibaba Group
Bellevue, USA
{yaliang.li,bolin.ding}@alibaba-inc.com

Nemotron-CLIMB: CLustering-based Iterative Data Mixture Bootstrapping for Language Model Pre-training

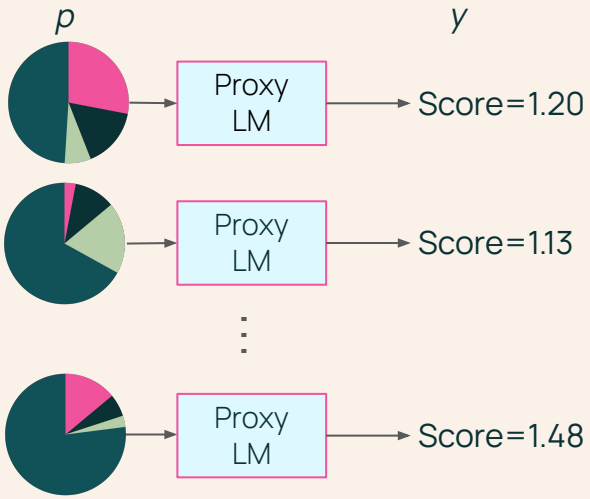
Shizhe Diao[1], **Yu Yang**[1], **Yonggan Fu**[1], **Xin Dong**[1], **Dan Su**[1], **Markus Kliegl**[1], **Zijia Chen**[1], **Peter Belcak**[1], **Yoshi Suhara**[1], **Hongxu Yin**[1], **Mostofa Patwary**[1], **Yingyan (Celine) Lin**[2], **Jan Kautz**[1], **Pavlo Molchanov**[1]
[1]NVIDIA [2]Georgia Institute of Technology

How data mixing?



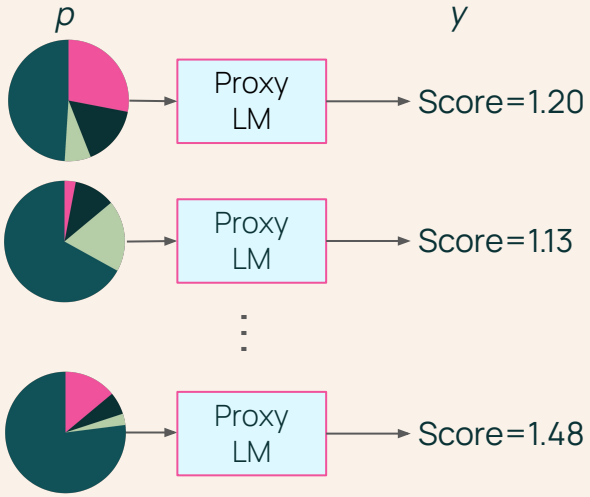
How data mixing?

1. **Swarm**: Train K small models with randomly sampled mixtures p



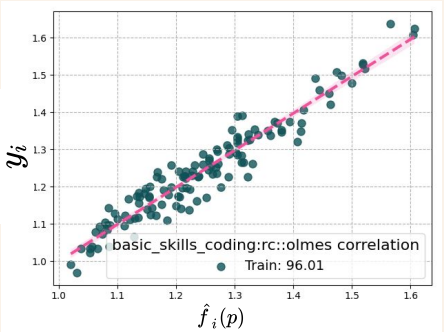
How data mixing?

1. **Swarm**: Train *K* small models with randomly sampled mixtures *p*



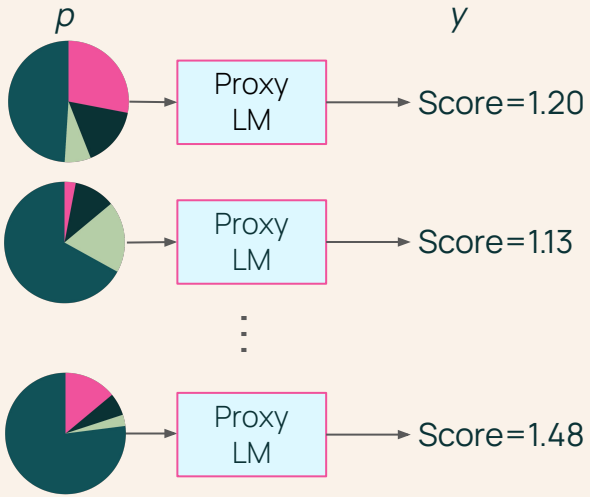
2. **Regression:** Fit function to predict LM performance given mixture *p*

$$\hat{f}_i(p) \approx y_i$$



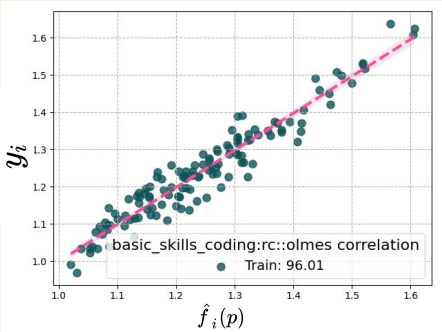
How data mixing?

1. **Swarm**: Train *K* small models with randomly sampled mixtures *p*



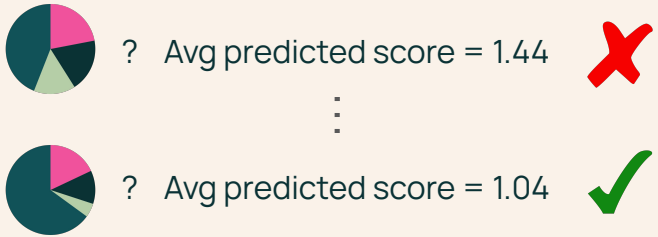
2. **Regression:** Fit function to predict LM performance given mixture *p*

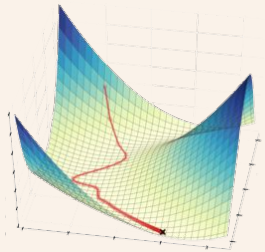
$$\hat{f}_i(p) \approx y_i$$



3. **Optimize:** Use fit function to solve for optimal mix p*

$$\underset{p \in \triangle^{m-1}}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} \hat{f}_i(p)$$





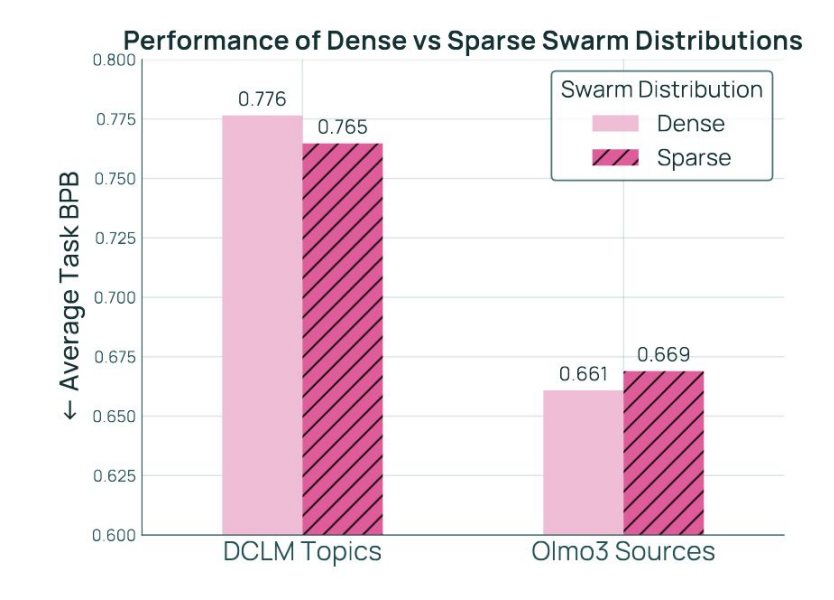
How data mixing?

Design Choice	RegMix (Liu et al., 2025a)	DML (Ye et al., 2025)	AutoScale (Kang et al., 2025)	BiMix (Ge et al., 2025b)	ADMIRE-BayesOpt (Chen et al., 2025b)	CLIMB (Diao et al., 2025)
Swarm Construction						
Proxy model size	1M	70, 160, 305, 410M	Target	280M	1M, 60M	350M
Swarm size (vs m domains)	512 ($m = 17$)	20 ($m = 7$)	$2m + 1$	4	101 ($m = 17$)	112 ($m = 21$)
Swarm distribution	Dirichlet with natural prior	Exponential grid	Exponential grid	Entropy-weighted	Dynamic	Dirichlet with natural prior
Regression Model						
Regression model family	LightGBM	Log-Linear	Power Law	Power Law	Gaussian Process	LightGBM
Regression granularity	Aggregated	Aggregated	Per-Task	Per-Task	Aggregated	Aggregated
Mixture Optimization						
Data repetition constraints	No	No	No	No	No	No
Optimization solver	Search	Search	Gradient Descent	Exact Solver	Search	Search

Problem 1: No “standard” config

Design Choice	RegMix (Liu et al., 2025a)	DML (Ye et al., 2025)	AutoScale (Kang et al., 2025)	BiMix (Ge et al., 2025b)	ADMIRE-BayesOpt (Chen et al., 2025b)	CLIMB (Diao et al., 2025)
Swarm Construction						
Proxy model size	1M	70, 160, 305, 410M	Target	280M	1M, 60M	350M
Swarm size (vs m domains)	512 ($m = 17$)	20 ($m = 7$)	$2m + 1$	4	101 ($m = 17$)	112 ($m = 21$)

Problem 1: No “standard” config



Problem 1: No “standard” config

Swarm construction:

RQ1 What is the smallest proxy model size (the number of parameters, S_{small}) such that decision-making generalizes to larger target models?

RQ2 How many proxy runs K do we need to learn a good mix on m domains?

RQ3 How should we specify the distribution $\mathcal{P}$ to sample the mixes for the proxy runs?

Regression model:

RQ4 Is there an optimal family of regression models ($\mathcal{F}$) for predicting mix performance?

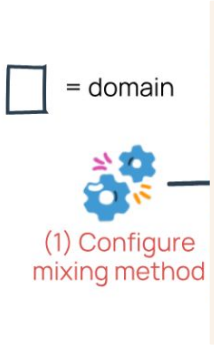
RQ5 At what granularity should we fit the regression models in order to construct $\hat{f}(p)$?

Mix optimization:

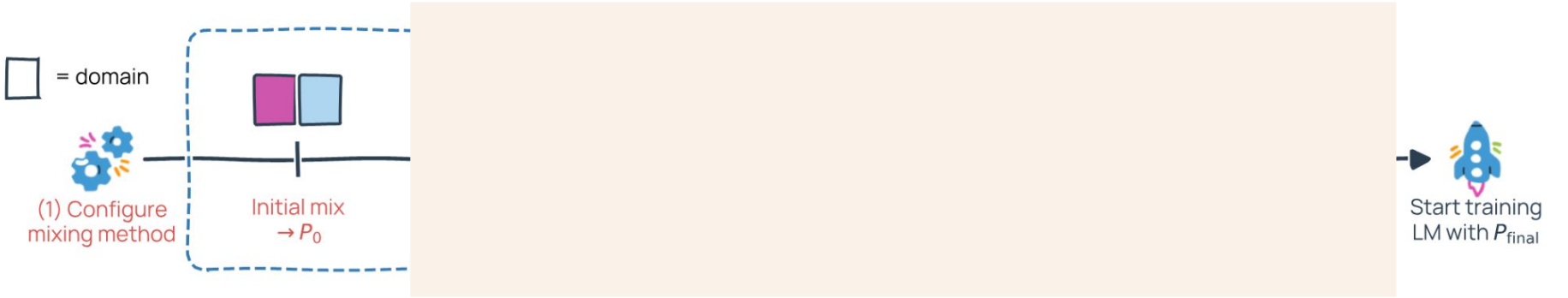
RQ6 How do we mix under finite data constraints?

RQ7 How do we solve the optimization problem?

Problem 2: Data changes during LM development

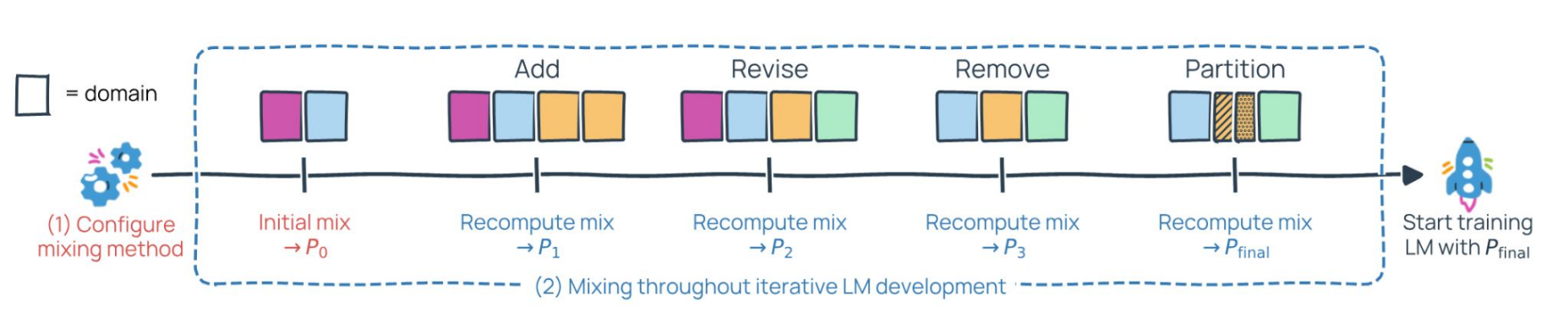


Problem 2: Data changes during LM development



Problem 2: Data changes during LM development

Problem 2: Data changes during LM development



Problem 2: Data changes during LM development

DCLM (Neurips 2025)

4.5 Dataset mixing

Often, Common Crawl (CC) is combined with other data sources that are considered high-quality [63, 70, 168, 170] (e.g., Wikipedia, StackExchange, and peS2o [156]). Since DCLM participants can include additional data sources in our mixing track, we examined the potential benefits of adding high-quality sources to training sets derived from Common Crawl only. We compare a model trained on 100% filtered CC data to models trained with the mixing ratios from Llama 1 and `RedPajama`: 67% CC, and 33% from Wikipedia, Books, Stack exchange, arXiv, and Github. For the CC component, we consider different variants: a subset of our DCLM-BASELINE, `RedPajama`'s CC portion, `RefinedWeb`, and `C4`. The results in Table 6 show that mixing improves performance for the lower-performing CC subsets (`C4`, `RedPajama`-CC, and `RefinedWeb`). In the case of DCLM-BASELINE however, mixing actually hurts performance on average, which suggests it can be counterproductive given performant filtering. For additional mixing results, see Appendix M.

Problem 2: Data changes during LM development

Olmo 3 (2025)

Add + Remove
- DCLM → Olmo 3 Common Crawl
- Stack v2 → StackEdu
- OpenWebMath → FineMath

Transform
- DCLM quality filter → FineWebEdu quality filter

Partition
- S2ORC PDFs → Classify by field of study

Problem 2: Data changes during LM development

Olmo 3 (2025)

Add + Remove
- DCLM → Olmo 3 Common Crawl
- Stack v2 → StackEdu
- OpenWebMath → FineMath

Transform
- DCLM quality filter → FineWebEdu quality filter

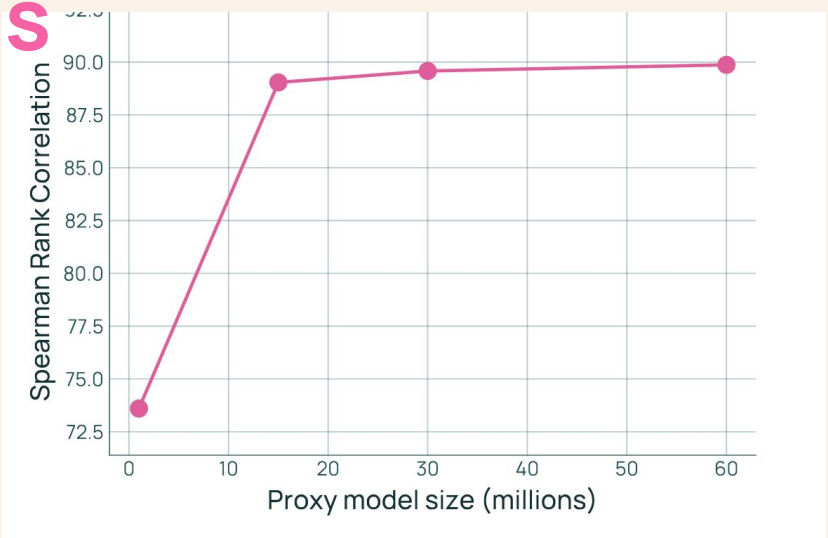
Partition
- S2ORC PDFs → Classify by field of study

Olmix: Data mixing over the LM development cycle

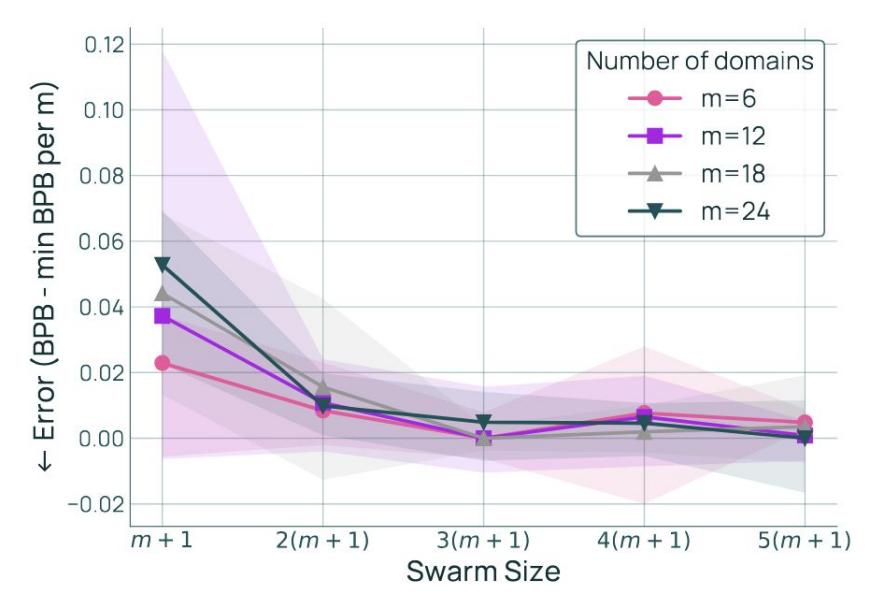
Ai2

Finding 1: Find smallest model size where performance ranking correlates well with larger models

Regmix paper

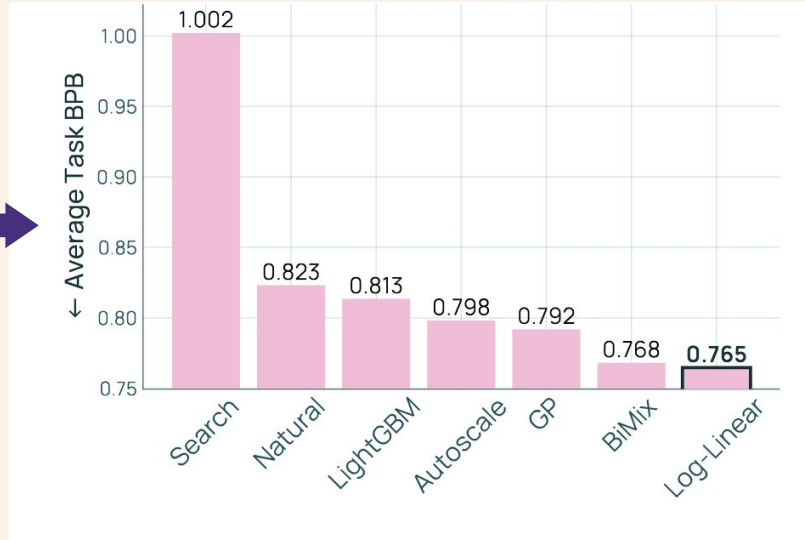


Finding 2: Swarm size scales linearly with number of domains

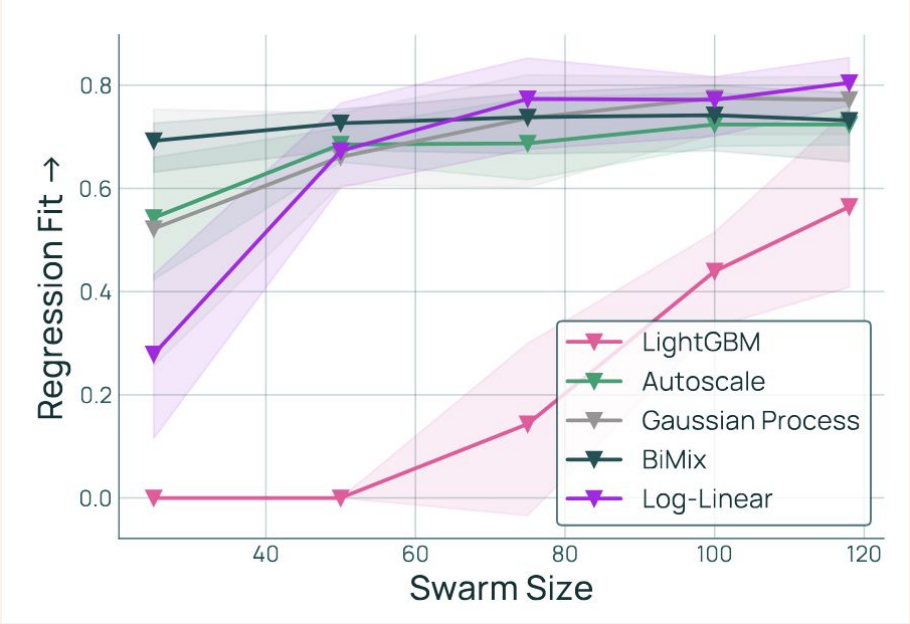


Finding 3: Regression fit is decently indicative of downstream performance

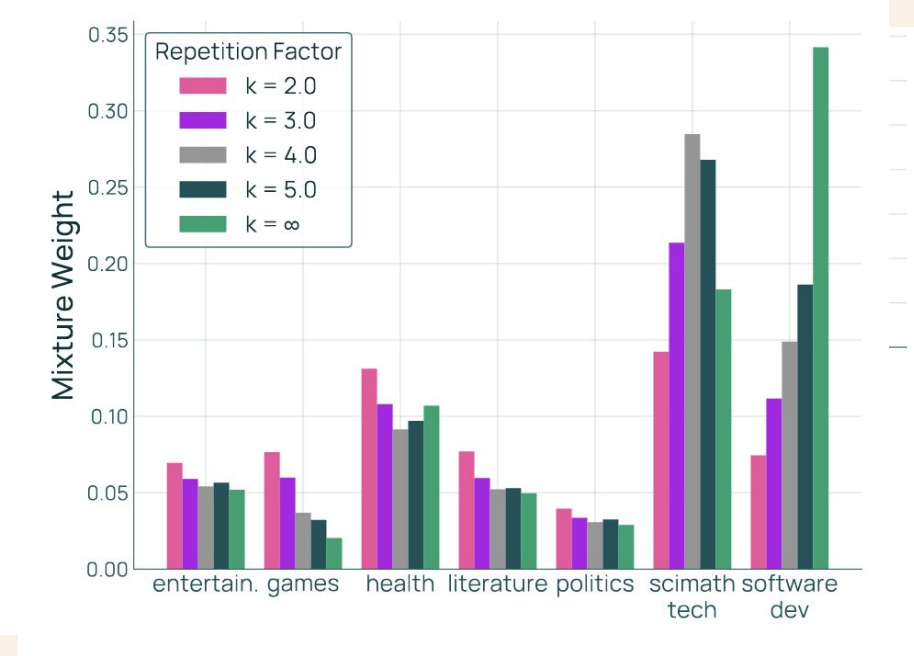
Finding 3: Regression fit is decently indicative of downstream performance



Finding 3b: Given sufficient compute, regression functional forms all similar



Finding 4: Data mixing dramatically sensitive if forced to repeat data



Finding 5: Historical data can be optimally reused to minimize experimental cost over time

Swarm

p1 = [0.3, 0.7]
p2 = [0.4, 0.6]
p3 = [0.8, 0.2]
...
pN = [0.1, 0.9]

→ p* = [0.66, 0.34]

Finding 5: Historical data can be optimally reused to minimize experimental cost over time

Swarm

p1 = [0.3, 0.7]
p2 = [0.4, 0.6]
p3 = [0.8, 0.2]
...
pN = [0.1, 0.9]

p* = [0.66, 0.34]

Add a domain

Finding 5: Historical data can be optimally reused to minimize experimental cost over time

Swarm

p1 = [0.3, 0.7]
p2 = [0.4, 0.6]
p3 = [0.8, 0.2]
...
pN = [0.1, 0.9]

p* = [0.66, 0.34]

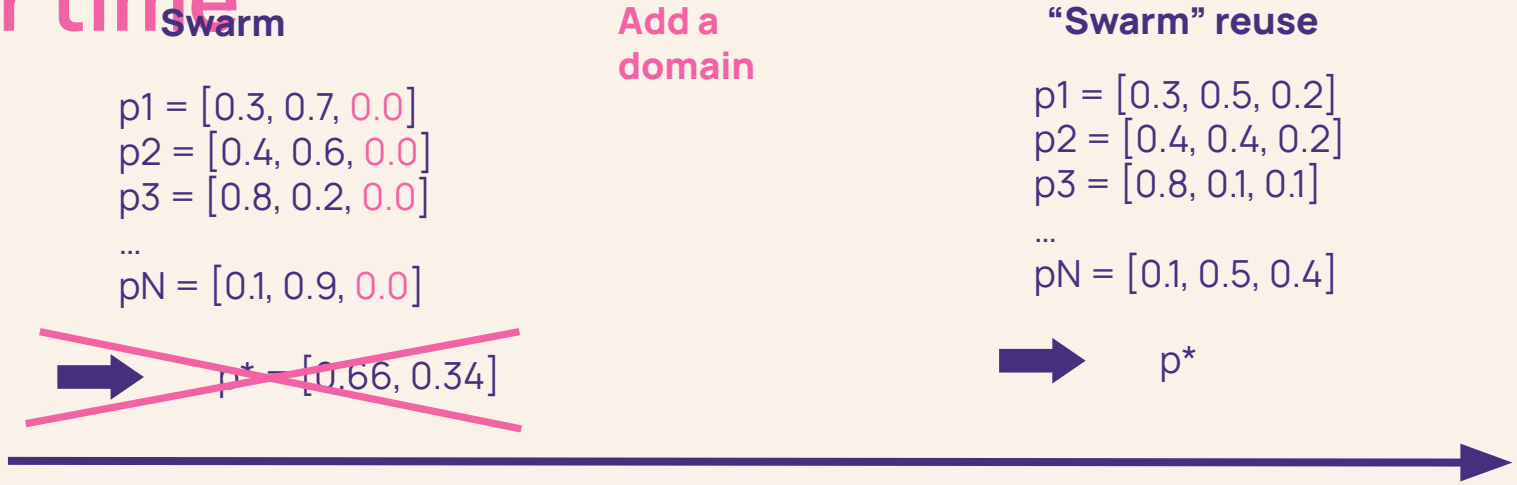
Add a domain

Full recomputation

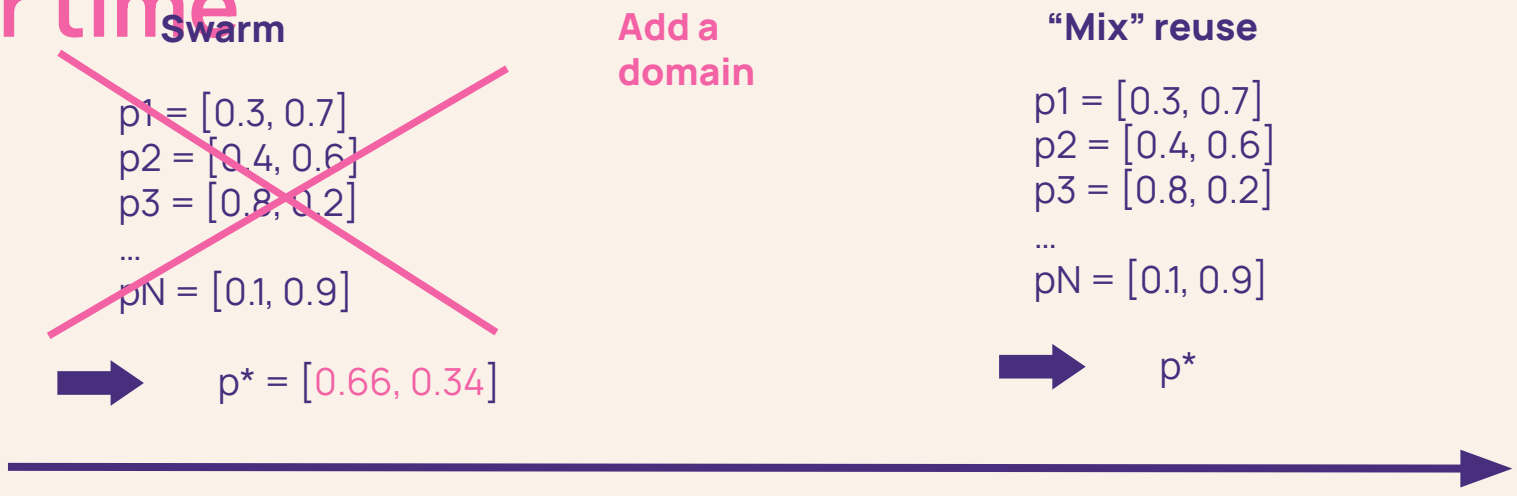
p1 = [0.3, 0.5, 0.2]
p2 = [0.4, 0.4, 0.2]
p3 = [0.8, 0.1, 0.1]
...
pN = [0.1, 0.5, 0.4]

p*

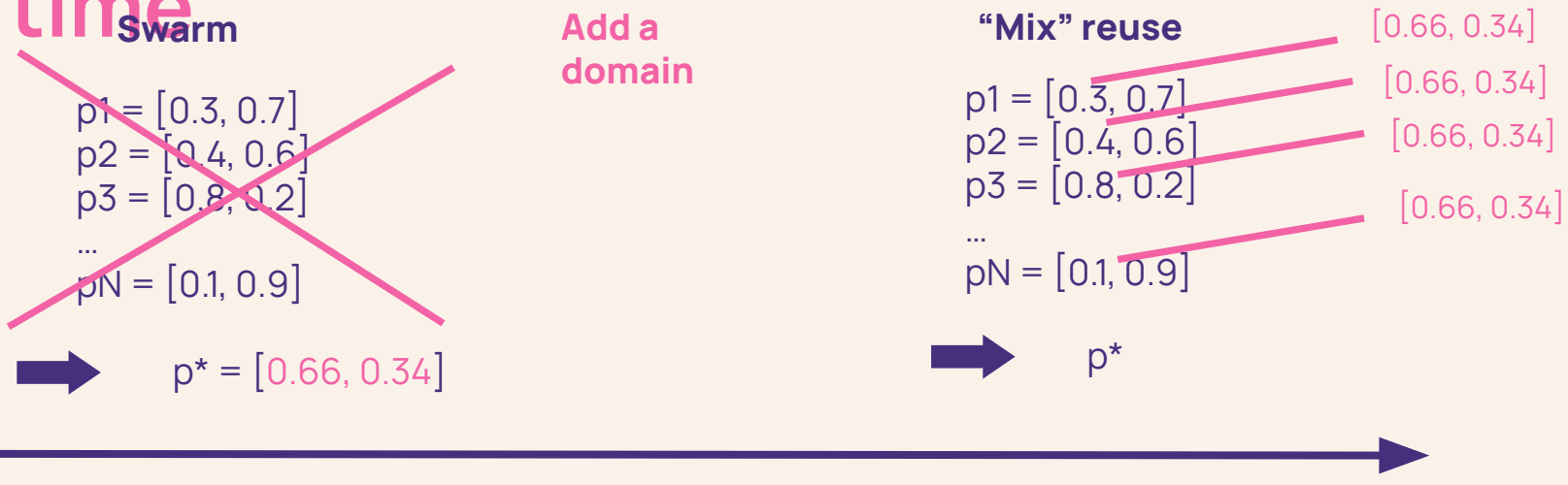
Finding 5: Historical data can be optimally reused to minimize experimental cost over time



Finding 5: Historical data can be optimally reused to minimize experimental cost over time



Finding 5: Historical data can be optimally reused to minimize experimental cost over time

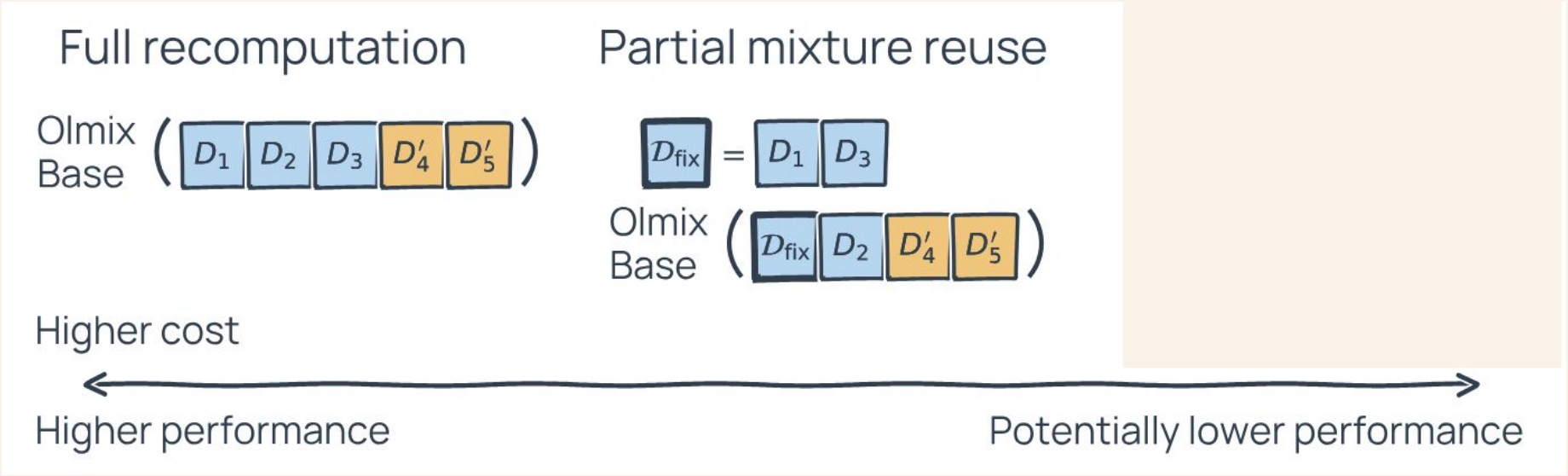


Finding 5: Mixture reuse

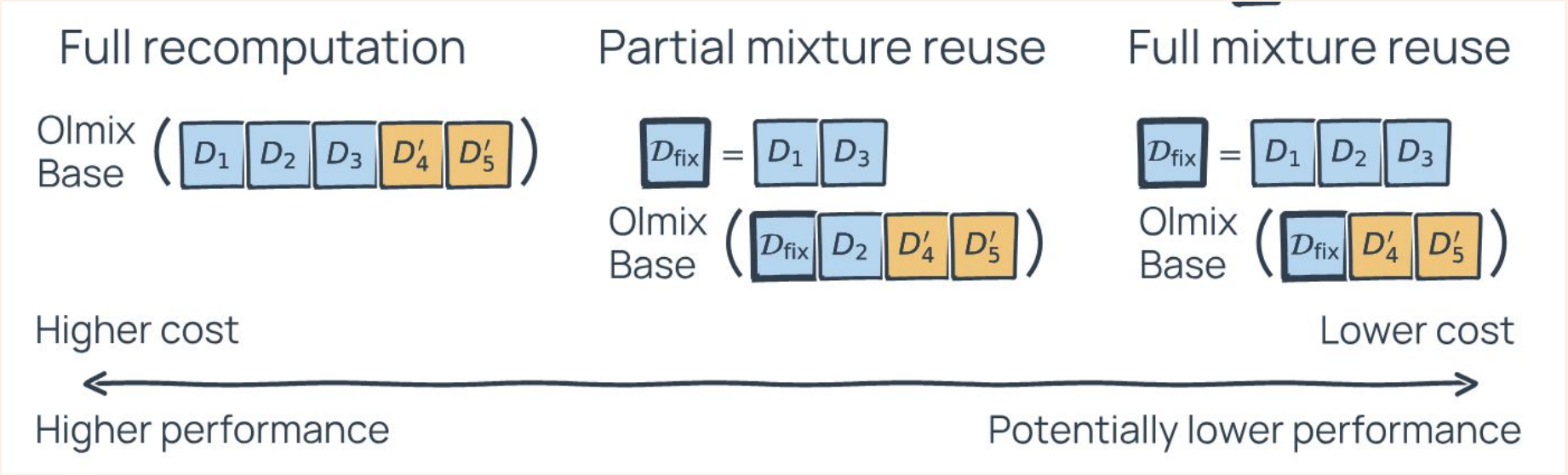
Finding 5: Mixture reuse

Finding 5: Mixture reuse

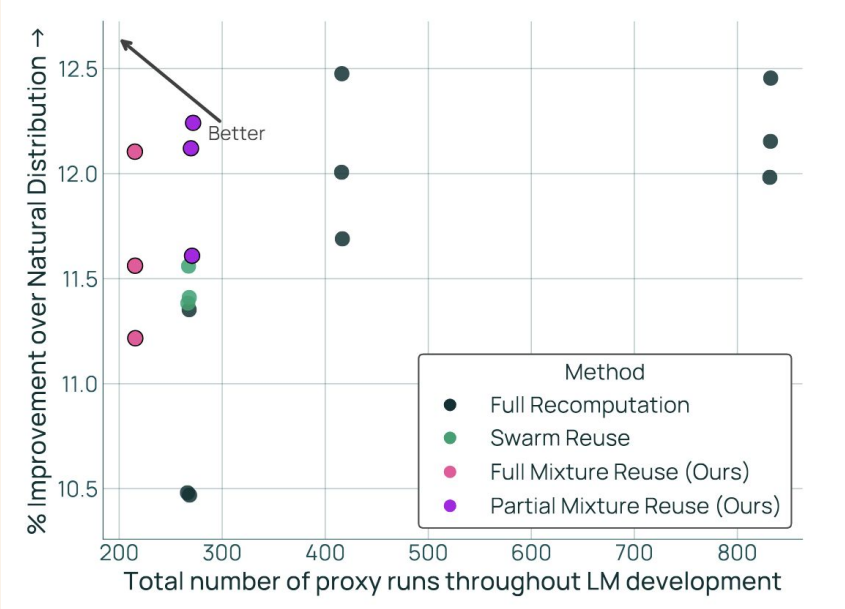
Finding 5: Mixture reuse



Finding 5: Mixture reuse



Finding 5: Historical data can be optimally reused to minimize experimental cost over time



Finding 5: Historical data can be optimally reused to minimize experimental cost over time

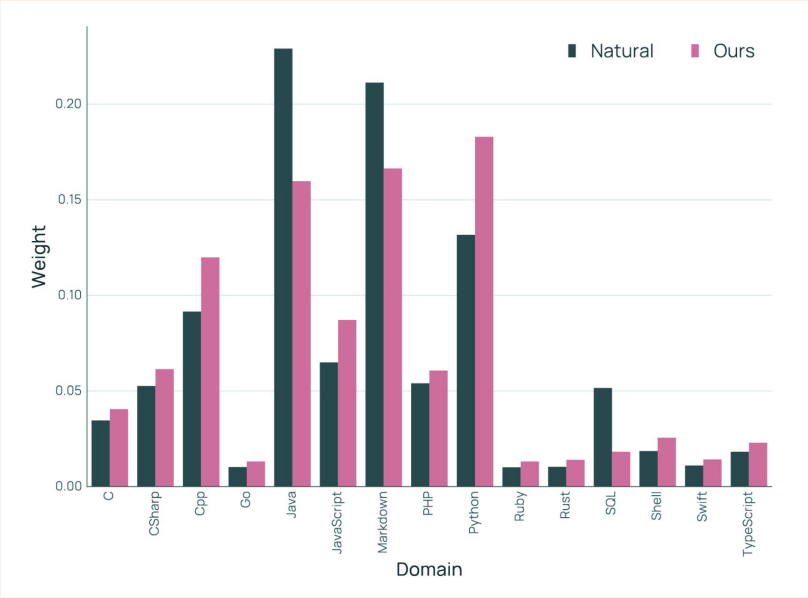
Theorem 4.1 (Performance gap bound). *The performance gap is bounded by*

$$F(q^\star(\tilde{p}_{\mathcal{D}_{fix}})) - F(q^\star) \leq C\|\tilde{p}_{\mathcal{D}_{fix}} - q^\star_{\mathcal{D}_{fix}}\|,$$
$$C = \mathcal{E}(\|\tilde{p}_{\mathcal{D}_{fix}} - q^\star_{\mathcal{D}_{fix}}\|)(\kappa(\alpha_{fix}, \alpha_{comp}) + \|\alpha_{fix}\|)$$

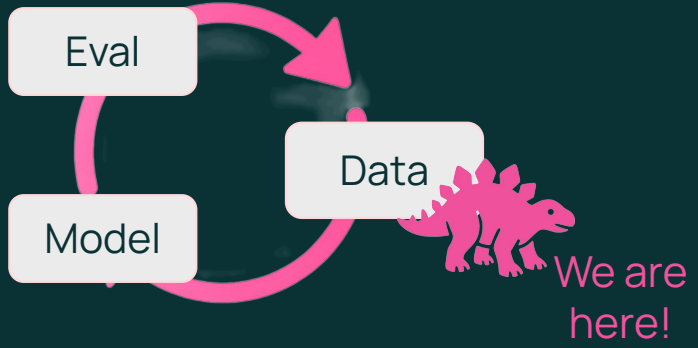
Theorem 4.2. *Assume that $\tilde{p}$ is the solution to (2) on $\mathcal{D}$. When new domains are added, the reuse gap is bounded by*

$$\|\tilde{p}_{\mathcal{D}_{fix}} - q^\star_{\mathcal{D}_{fix}}\| \leq \mathcal{E}(1-\rho^\star)\kappa(\alpha_{fix}, \alpha_{comp})(1-\rho^\star)$$

Data mixing



Ai2



Olmo 3

Olmo Team*

Allyson Ettinger[♥1] Amanda Bertsch[♥1,3] Bailey Kuehl[♥1] David Graham[♥1]
David Heineman[♥1] Dirk Groeneveld[♥1] Faeze Brahman[♥1] Finbarr Timbers[♥1]
Hamish Ivison[♥1,2] Jacob Morrison[♥1,2] Jake Poznanski[♥1] Kyle Lo[♥1,2] Luca Soldaini[♥1]
Matt Jordan[♥1] Mayee Chen[♥1,4] Michael Noukhovitch[♥1,5,6] Nathan Lambert[♥1]
Pete Walsh[♥1] Pradeep Dasigi[♥1] Robert Berry[♥1] Saumya Malik[♥1] Saurabh Shah[♥1]
Scott Geng[♥1,2] Shane Arora[♥1] Shashank Gupta[♥1] Taira Anderson[♥1] Teng Xiao[♥1]
Tyler Murray[♥1] Tyler Romero[♥1] Victoria Graf[♥1,2]

Akari Asai[1,3] Akshita Bhagia[1] Alexander Wettig[7] Alisa Liu[2] Aman Rangapur[1]
Chloe Anastasiades[1] Costa Huang[1] Dustin Schwenk[1] Harsh Trivedi[1] Ian Magnusson[1,2]
Jaron Lochner[1] Jiacheng Liu[1] Lester James V. Miranda[1] Maarten Sap[1,3] Malia Morgan[1]
Michael Schmitz[1] Michal Guerquin[1] Michael Wilson[1] Regan Huff[1] Ronan Le Bras[1]
Rui Xin[2] Rulin Shao[2] Sam Skjonsberg[1] Shannon Zejiang Shen[8] Shuyue Stella Li[2]
Tucker Wilde[1] Valentina Pyatkin[1] Will Merrill[1] Yapei Chang[3] Yuling Gu[1] Zhiyuan Zeng[1,2]

Ashish Sabharwal[1] Luke Zettlemoyer[2] Pang Wei Koh[1,2]
Ali Farhadi[1,2] Noah A. Smith[♥1,2] Hannaneh Hajishirzi[♥1,2]

[1]Allen Institute for AI [2]University of Washington [3]Carnegie Mellon University [4]Stanford University [5]Mila
[6]Université de Montréal [7]Princeton University [8]Massachusetts Institute of Technology [9]University of Maryland

*Olmo 3 was a team effort; authors sorted alphabetically. ♥marks core contributors. See author contributions here.

- **Olmo 3 Base:** `Olmo-3-1025-7B` `Olmo-3-1125-32B`
- **Olmo 3 Think:** `Olmo-3-7B-Think` `Olmo-{3|3.1}-32B-Think`
- **Olmo 3 Instruct:** `Olmo-3-7B-Instruct` `Olmo-3.1-32B-Instruct`
- **Olmo 3 RL Zero:** `Olmo-3-7B-RL-Zero-{Math|Code|IF|General|Mix}` `Olmo-3.1-7B-RL-Zero-{Math|Code}`
- **Base Data:** `Pretrain: Dolma 3 Mix` `Midtrain: Dolma 3 Dolmino Mix` `Long-ctx: Dolma 3 Longmino Mix`
- **Think Data:** `Dolci-Think-{SFT|DPO|RL}-7B` `Dolci-Think-{SFT|DPO|RL}-32B`

Contact: olmo@allenai.org

3.4 Stage 1: Pretraining

We first train Olmo 3 Base on Dolma 3 Mix, our 6T token pretraining data mix. While Dolma 3 Mix is comprised of largely the same types of data sources used in other open pretraining recipes (Soldaini et al., 2024; Bakouch et al., 2025; OLMo et al., 2024), we demonstrate three key novelties:

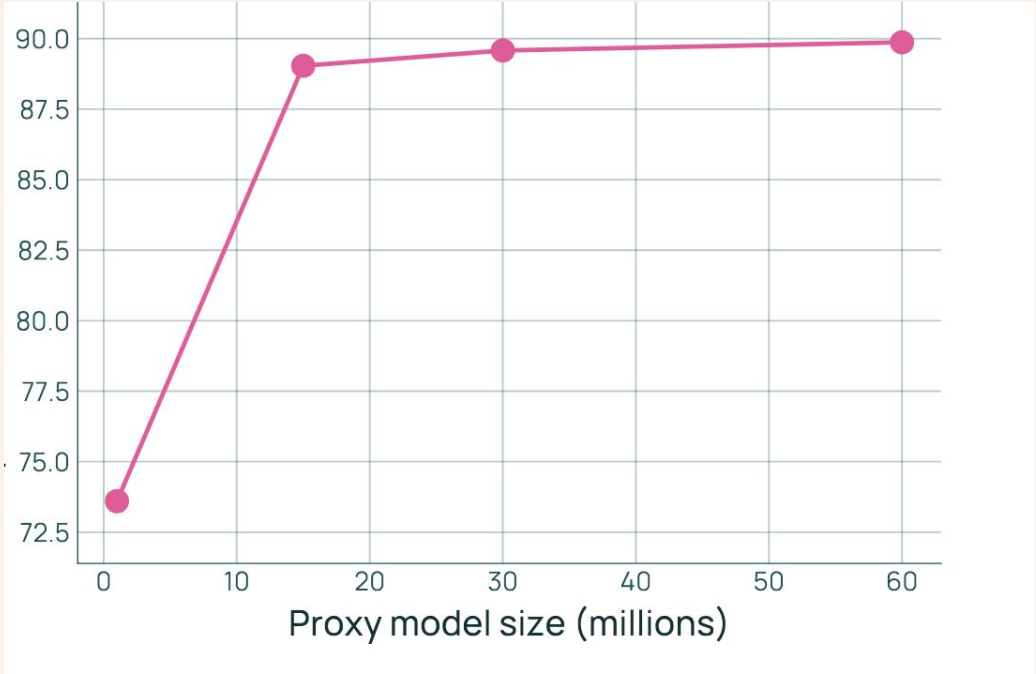
Abstract

We introduce **Olmo 3**, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. Olmo 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire **model flow**, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, **Olmo 3.1 Think 32B**, is the strongest fully-open thinking model released to-date.

Ai2

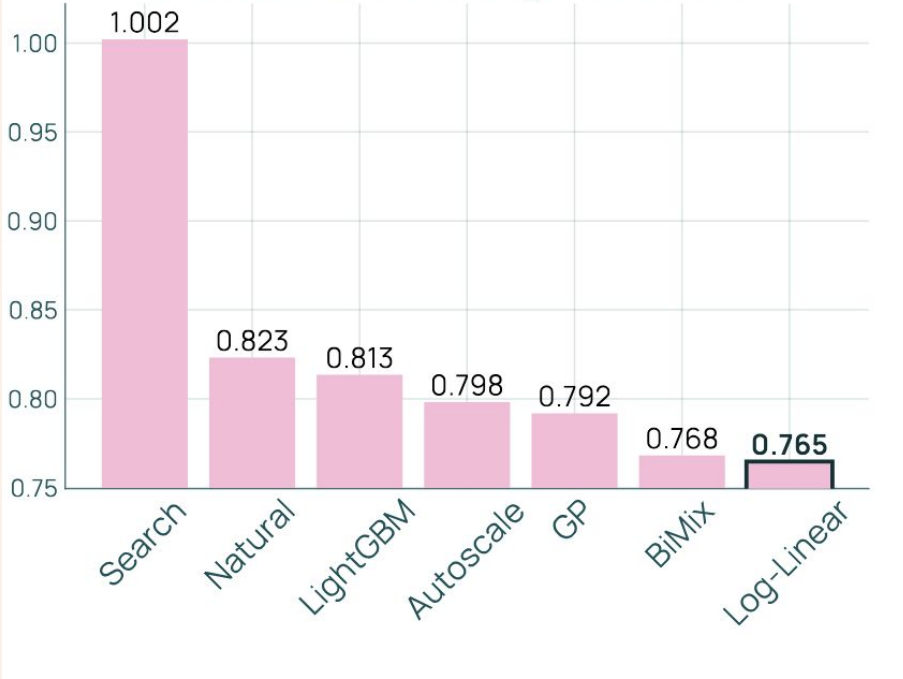
Data mixing: Infrastructure

Correlation between proxy and large model performances

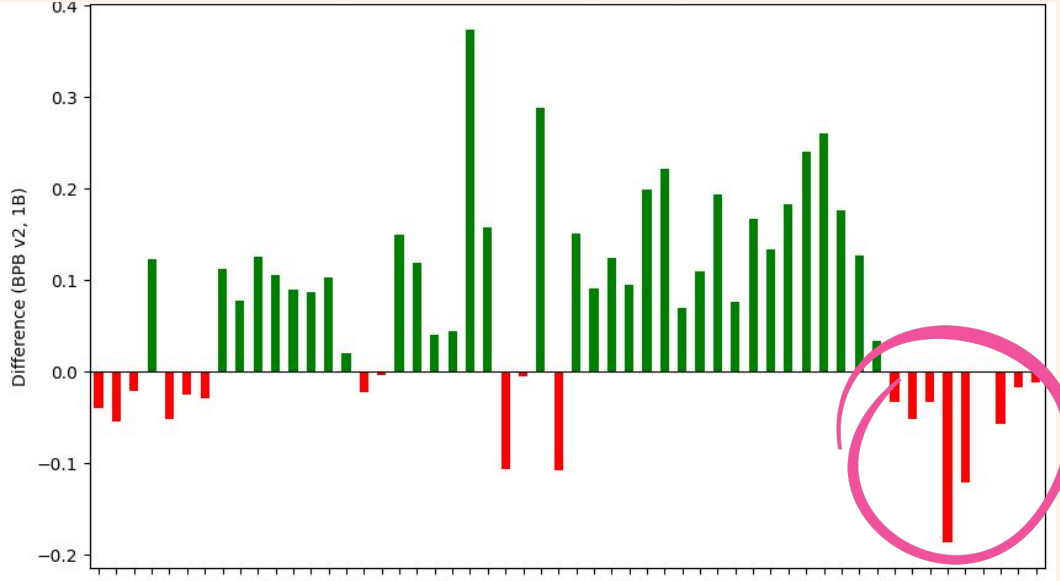


Data mixing: Infrastructure

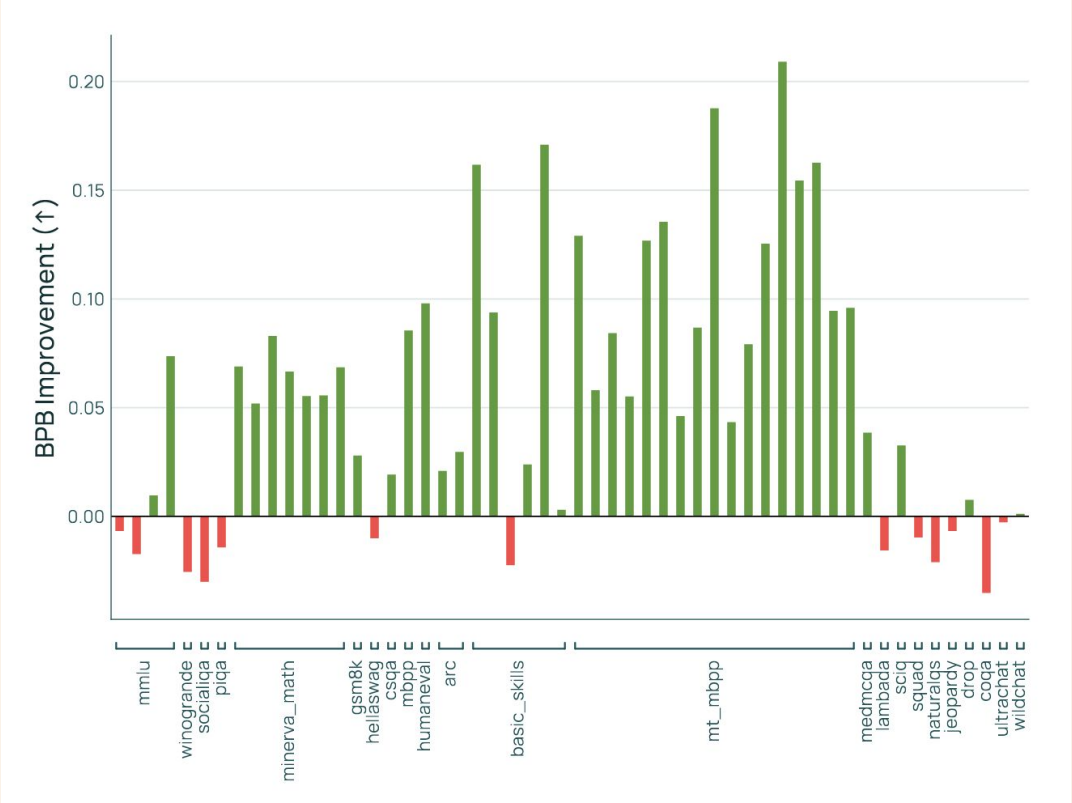
Performance
(BPB, lower is better)



Data mixing: Operations

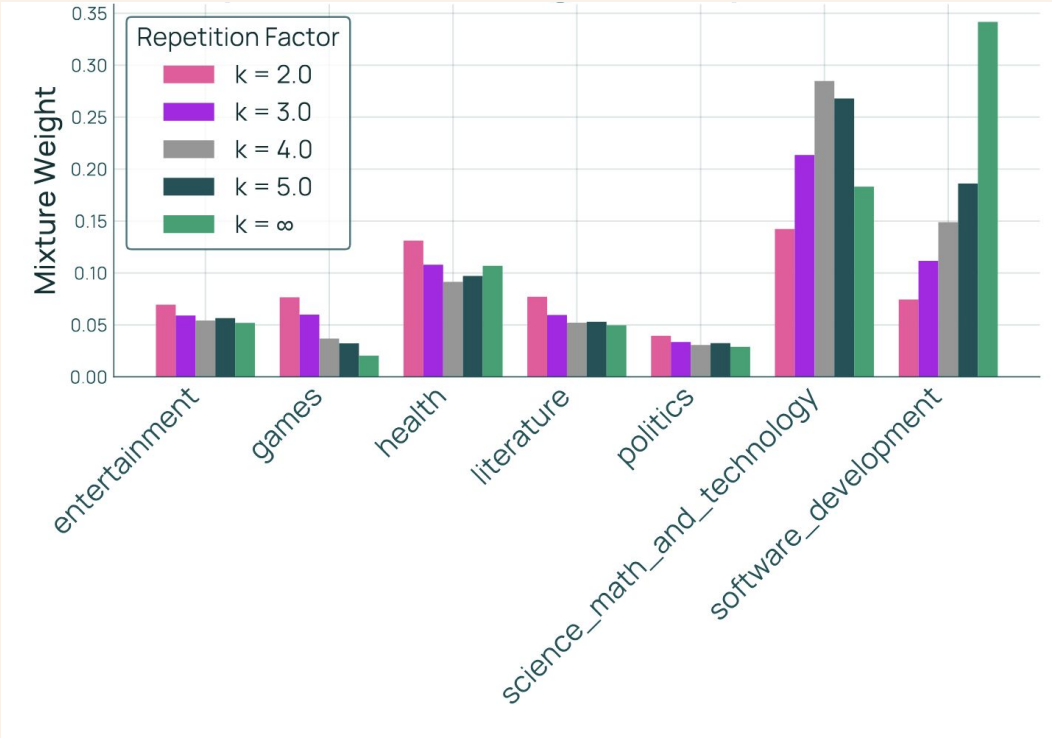


Data mixing: Operations

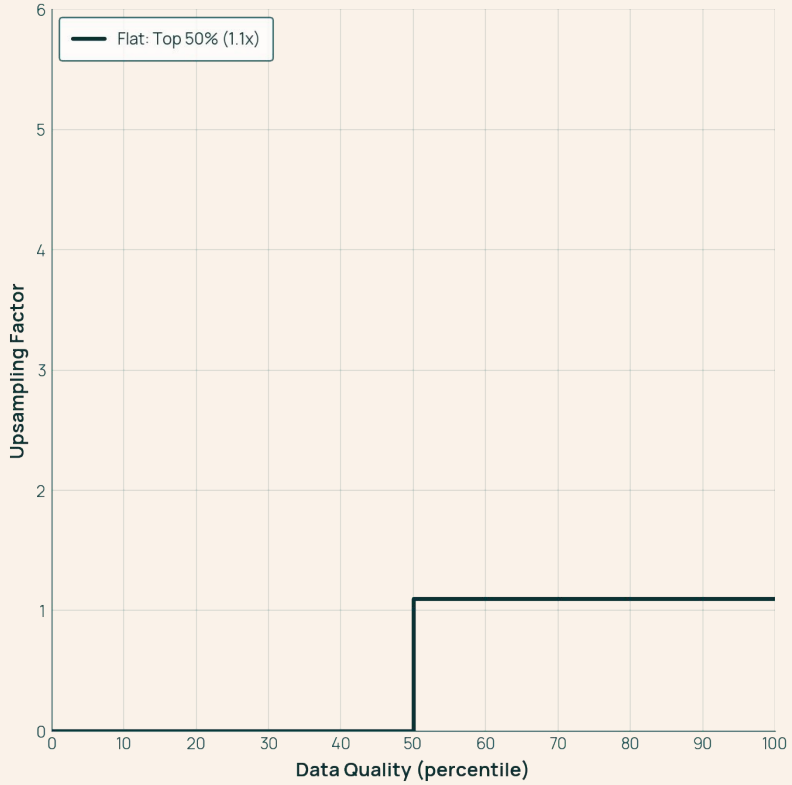


Ai2

Data mixing under data constraints

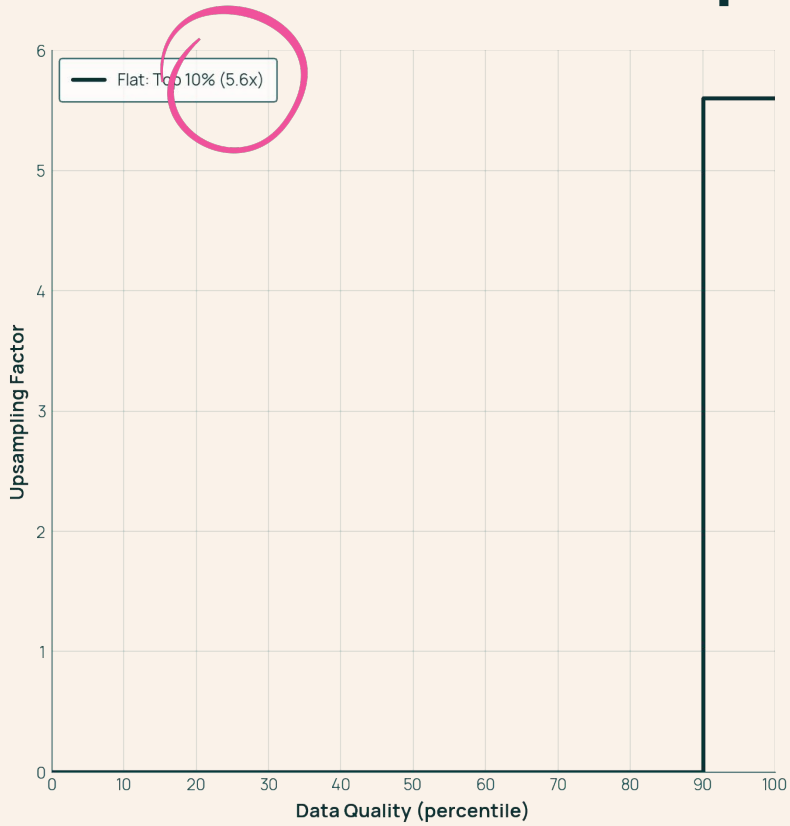


Data constraints influence upsampling



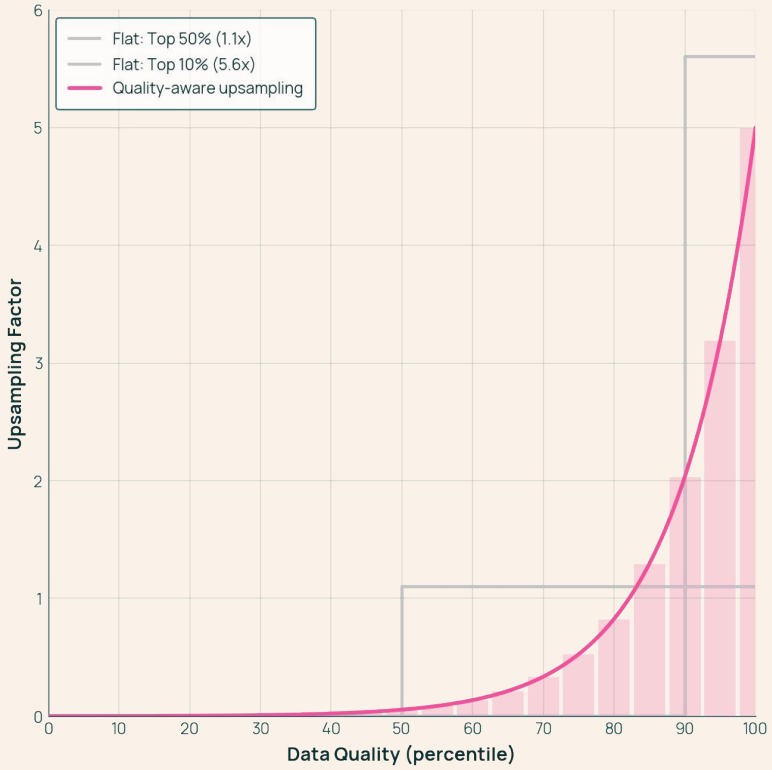
Ai2

Data constraints influence upsampling

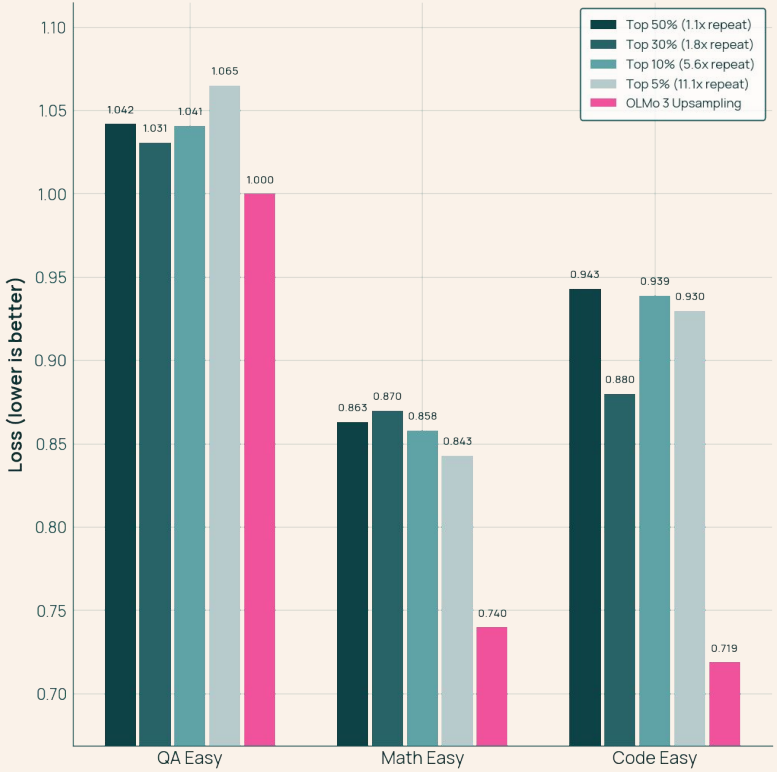


Ai2

Quality-aware upsampling



Quality aware upsampling

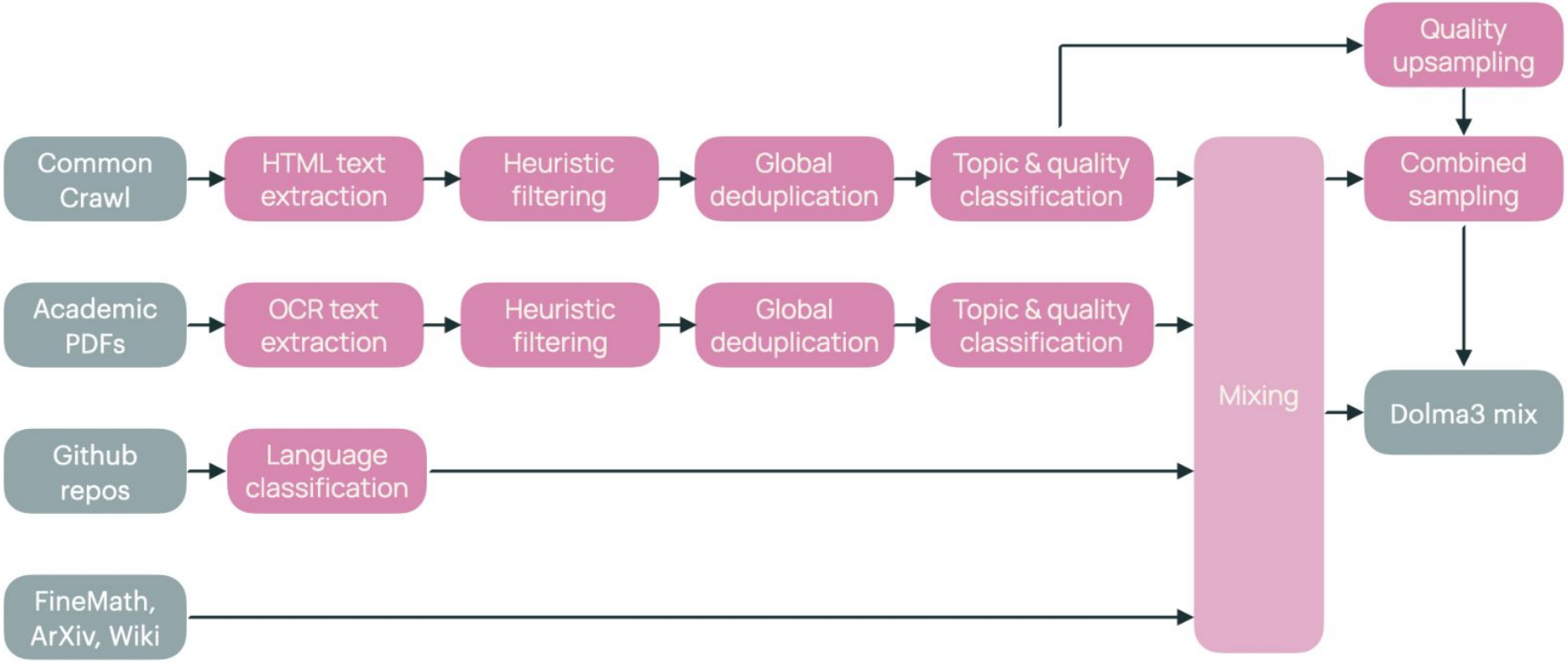


Ai2

Pretraining: Dolma 3

Source	Type	9T Pool		6T Mix	
		Tokens	Documents	Tokens	Ratio
Common Crawl	Web pages	8.14T	9.67B	4.51T	76.1%
olmOCR Science PDFs	Academic documents	972B	101M	804.9B	13.6%
StackEdu (Rebalanced)	GitHub code	137B	167M	408.9B	6.9%
arXiv	Papers with LaTeX	21.4B	3.95M	50.8B	0.9%
FineMath 3+	Math web pages	34.1B	21.4M	151.9B	2.6%
Wikipedia & Wikibooks	Encyclopedic	3.69B	6.67M	2.5B	0.04%
Total		**9.31T**	**9.97B**	**5.93T**	**100%**

Extraction, filtering, deduplication, quality classification


Common Crawl
HTML text extraction
Heuristic filtering
Global deduplication
Topic & quality classification
Quality upsampling
Combined sampling
Academic PDFs
OCR text extraction
Heuristic filtering
Global deduplication
Topic & quality classification
Mixing
Dolma3 mix
Github repos
Language classification
FineMath, ArXiv, Wiki

Filtering, Deduplication, Fine-Grained Organization

Common Crawl

255B Docs

Heuristic Filtering

- PII filtering
- Language filtering
- URL removal

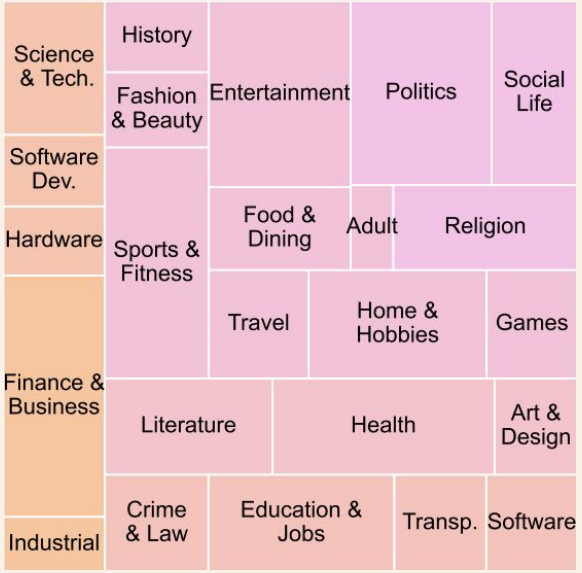
39B Docs

Deduplication

- Exact, global deduplication (new tooling!)
- Min-hash (fuzzy)
- Suffix array for boilerplate

10 B Docs

Web Organizer



Ai2

Wettig et al., "Organize the Web: Constructing Domains Enhances Pre-Training Data Curation", ICML 2025

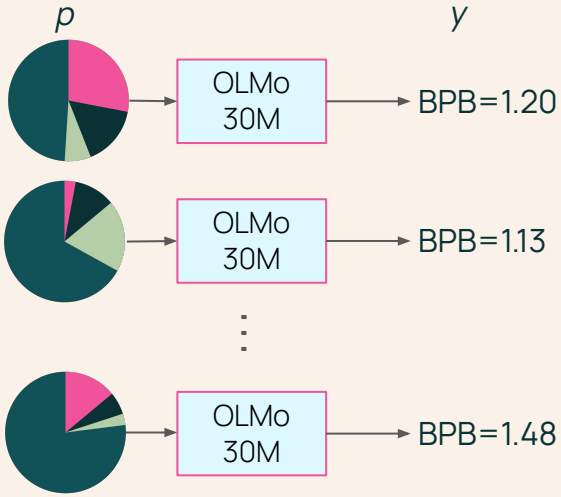
From Pools to a Mix

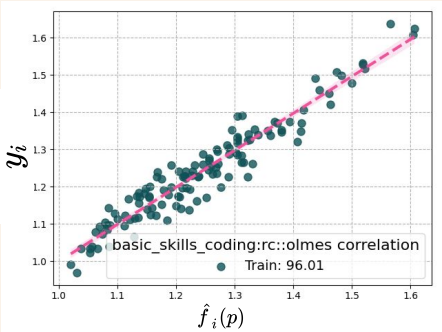
1. **Swarm runs**: train *K* 30M models with randomly sampled mixtures *p*

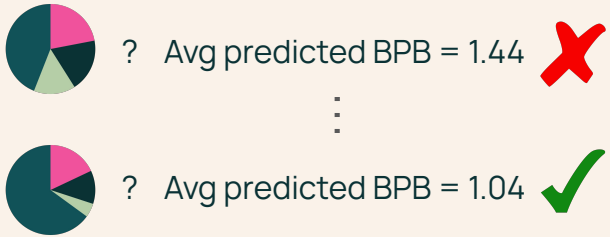
2. **Fit a regression model** for each benchmark task $\hat{f}_i(p) \approx y_i$

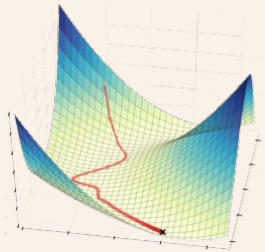
3. **Solve optimization problem** to get optimal mix p*

$$\underset{p \in \triangle^{m-1}}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} \hat{f}_i(p)$$



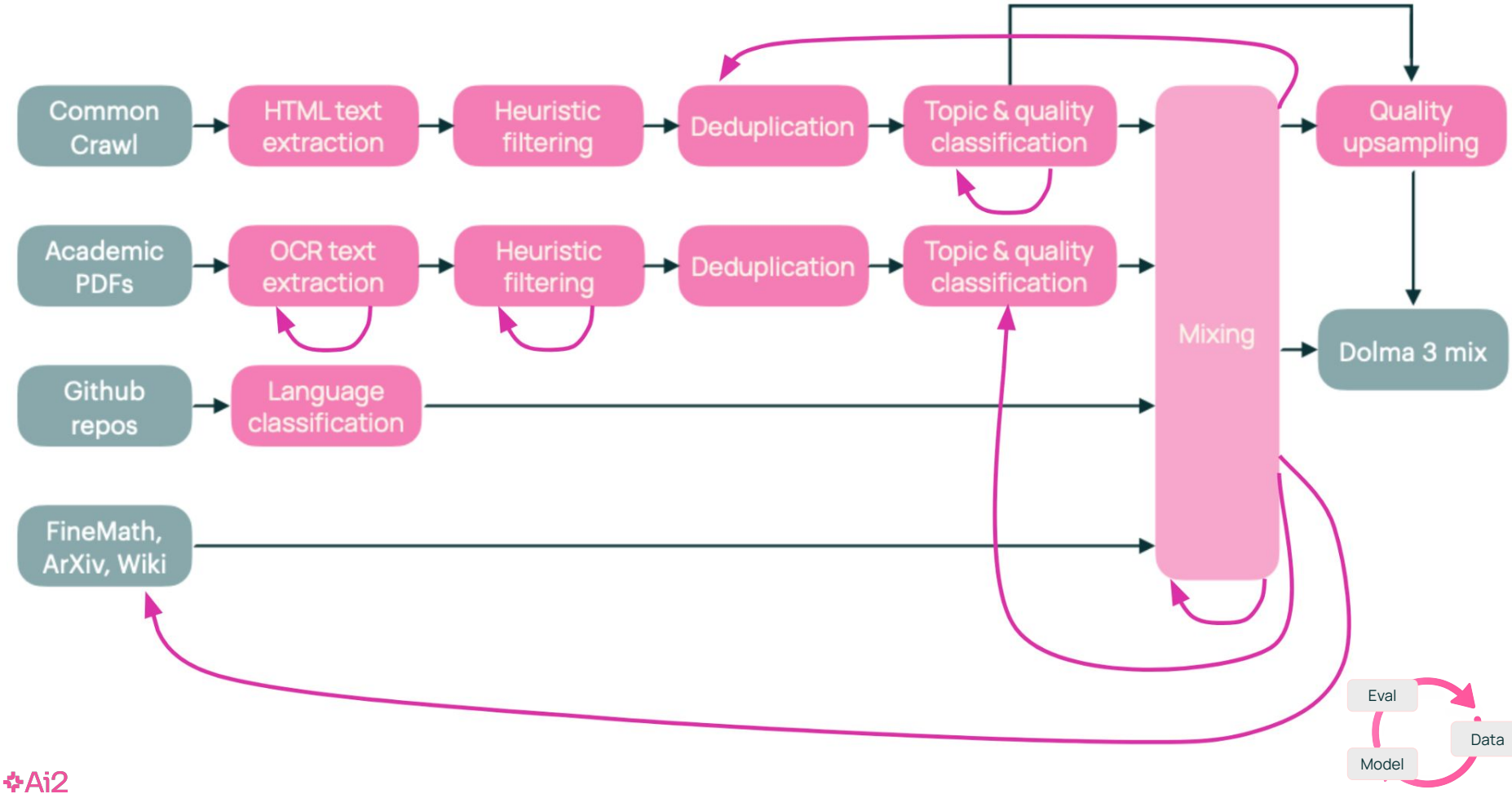






Ai2

Chen et al., "Olmix: Efficient Mixture Recomputation for Evolving LM Datasets", 2025

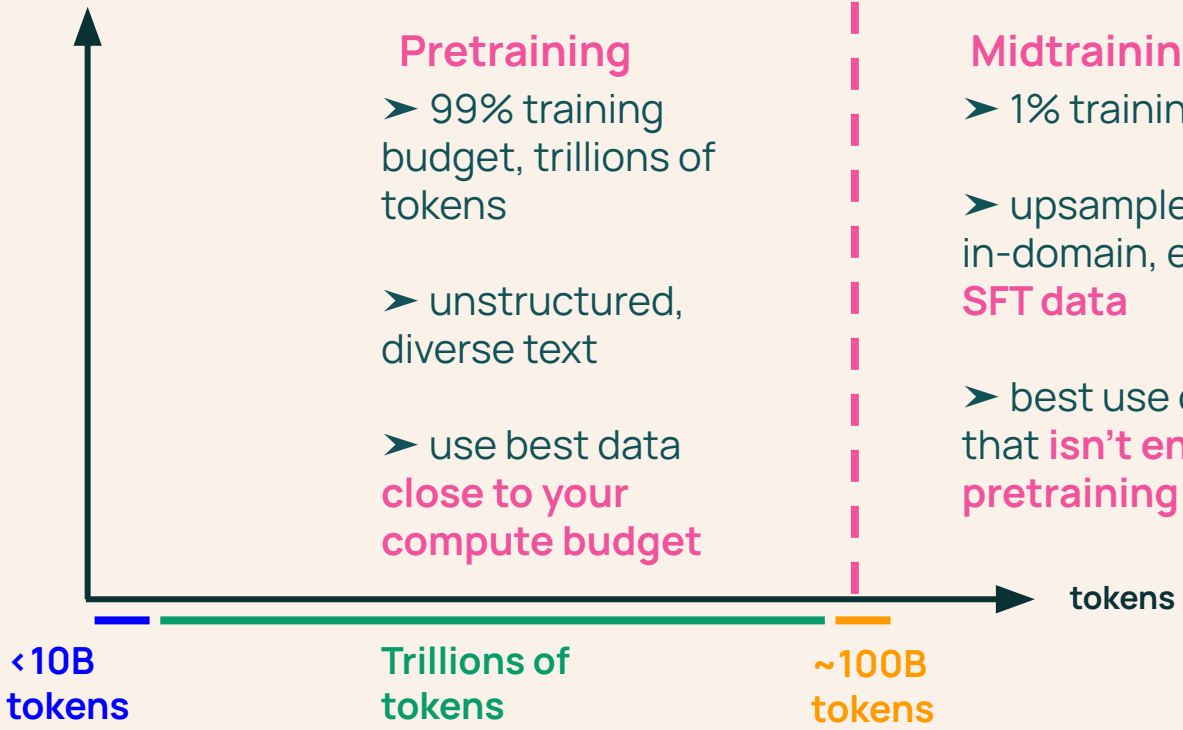

Common Crawl
HTML text extraction
Heuristic filtering
Deduplication
Topic & quality classification
Quality upsampling
Academic PDFs
OCR text extraction
Heuristic filtering
Deduplication
Topic & quality classification
Mixing
Dolma 3 mix
Github repos
Language classification
FineMath, ArXiv, Wiki
Eval
Data
Model

Pretraining: Dolma 3

Source	Type	9T Pool		6T Mix	
		Tokens	Documents	Tokens	Ratio
Common Crawl	Web pages	8.14T	9.67B	4.51T	76.1%
olmOCR Science PDFs	Academic documents	972B	101M	804.9B	13.6%
StackEdu (Rebalanced)	GitHub code	137B	167M	408.9B	6.9%
arXiv	Papers with LaTeX	21.4B	3.95M	50.8B	0.9%
FineMath 3+	Math web pages	34.1B	21.4M	151.9B	2.6%
Wikipedia & Wikibooks	Encyclopedic	3.69B	6.67M	2.5B	0.04%
Total		**9.31T**	**9.97B**	**5.93T**	**100%**

Upsampled; higher quality increases probability of sampling a document

Tension between Scale and Quality



Dolma 3 Dolmino: Midtraining Mix

Math problem-solving through **code and/or discussion**

Instruction datasets

STEM PDFs and high-quality web pages

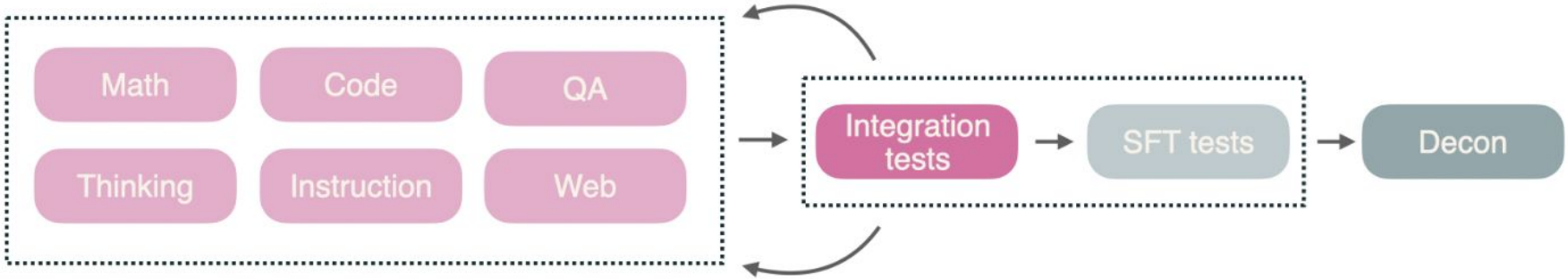
Type	Source	2T Pool		100B Mix	
		Tokens	Docs	Tokens	Docs
Math (synth)	TinyMATH Mind**	899M	1.42M	898M (0.9%)	1.52M
Math (synth)	TinyMATH PoT**	241M	729K	241M (0.24%)	758K
Math (synth)	CraneMath*	5.62B	6.55M	5.62B (5.63%)	7.24M
Math (synth)	MegaMatt*	3.88B	6.79M	1.73B (1.73%)	3.23M
Math (synth)	Dolmino Math^^	10.7B	21M	10.7B (10.7%)	22.3M
Code	StackEdu (FIM)^	21.4B	32M	10.0B (10.0%)	16.2M
Python (synth)	CraneCode*	18.8B	19.7M	10.0B (10.0%)	11.7M
QA (synth)	Reddit To Flashcards**	21.6B	370M	5.90B (5.9%)	101M
QA (synth)	Wiki To RCQA**	4.22B	22.3M	3.0B (3.0%)	16.3M
QA (synth)	Nemotron Synth QA^	487B	972M	5.0B (5.0%)	10.6M
Thinking (synth)	Math Meta-Reasoning**	1.05B	984K	381M (0.38%)	401K
Thinking (synth)	Code Meta-Reasoning**	1.27B	910K	459M (0.46%)	398K
Thinking (synth)	Program-Verifiable**	438M	384K	159M (0.16%)	158K
Thinking (synth)	OMR Rewrite FullThoughts^	850M	291K	850M (0.85%)	394K
Thinking (synth)	QWQ Reasoning Traces^	4.77B	438K	1.87B (1.87%)	401K
Thinking (synth)	General Reasoning Mix^	2.48B	668K	1.87B (1.87%)	732K
Thinking (synth)	Gemini Reasoning Traces^	246M	55.2K	246M (0.25%))	85.1K
Thinking (synth)	Llama Nemotron Reasoning Traces^	20.9B	3.91M	1.25B (1.25%)	368K
Thinking (synth)	OpenThoughts2 Reasoning Traces^	5.6B	1.11M	1.25B (1.25%)	402K
Instruction (synth)	Tulu 3 SFT^^	1.61B	1.95M	1.1B (1.1%)	1.45M
Instruction (synth)	Dolmino 1 Flan^^	16.8B	56.9M	5.0B (5.0%)	14.8M
PDFs	OLMOCR Science PDFs (High Q.)^	240B	28.7M	4.99B (5.0%)	1.20M
Web pages	STEM-Heavy Crawl^	5.21B	5.16M	4.99B (5.0%)	5.53M
Web pages	Common Crawl (High Q.)^	1.32T	965M	22.4B (22.5%)	18.3M
Total		**2.19T**	**2.52B**	**99.95B (100%)**	**236M**

Demonstration of **diverse question structures**, rewritten from natural knowledge-rich data

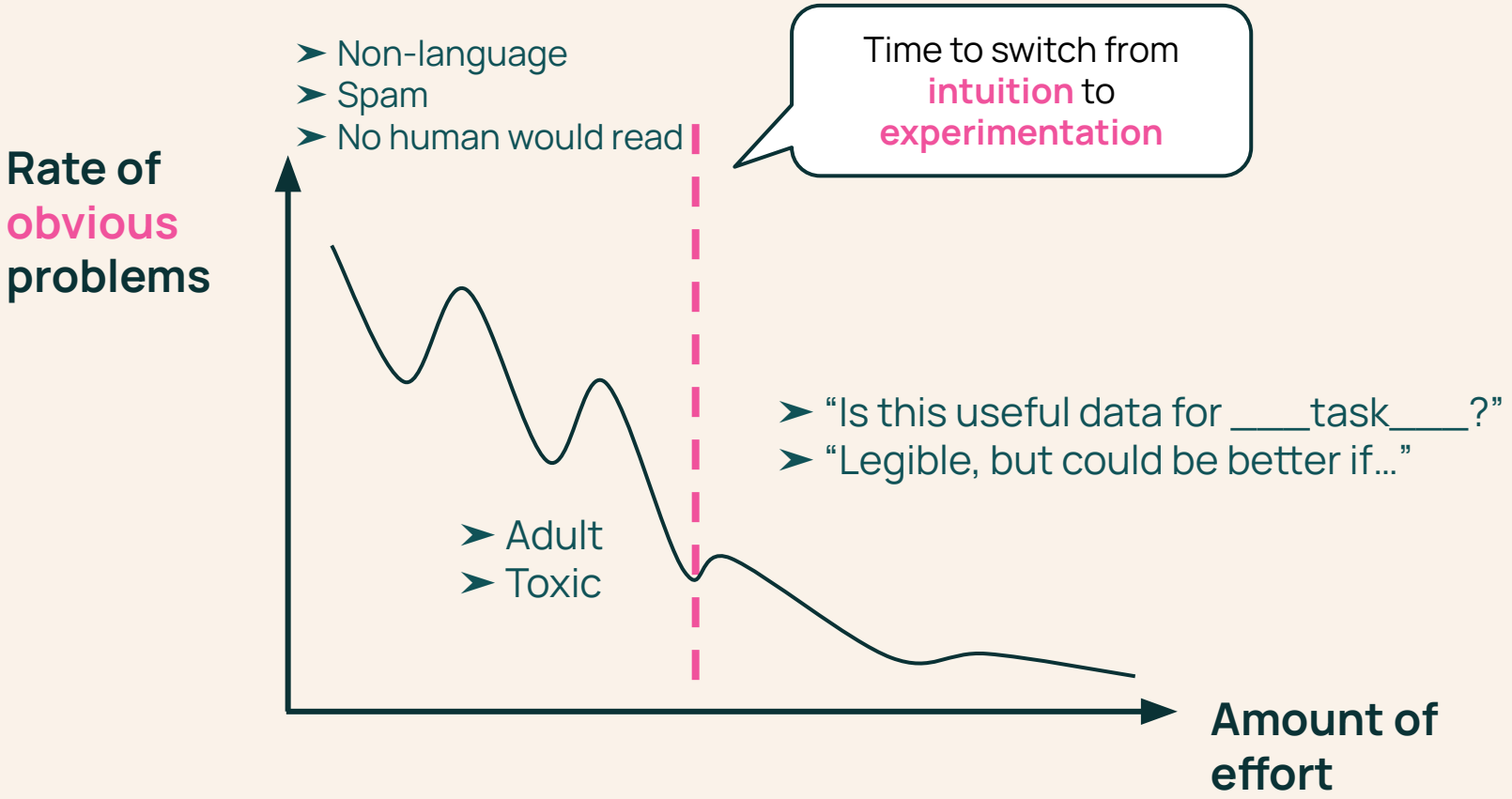
Math and code problem-solving using **human-inspired meta-reasoning strategies/traces**

Ai2

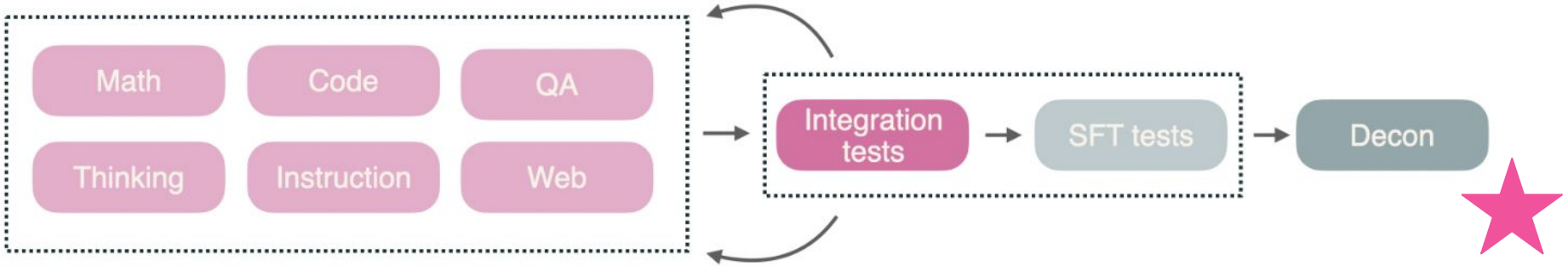
Distributed Exploration & Centralized Evaluation



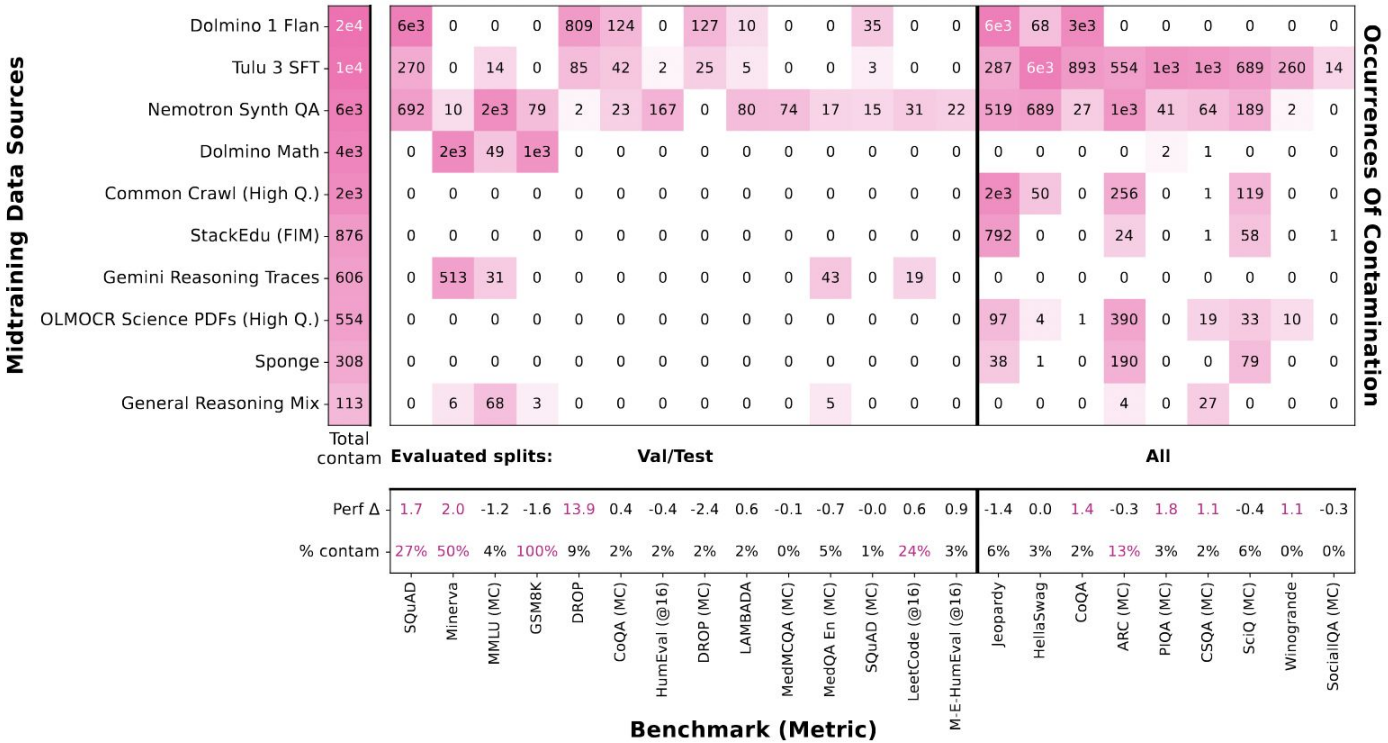
Inspect Data Often, Favor Fast, Scalable Improvements



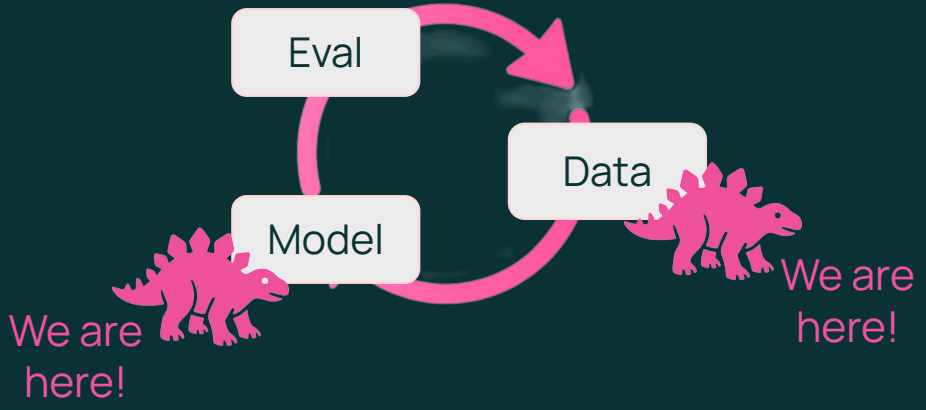
Distributed Exploration & Centralized Evaluation



Decontamination in Dolmino



Ai2



Olmo 3

Olmo Team*

Allyson Ettinger♥[1] Amanda Bertsch♥[1,3] Bailey Kuehl♥[1] David Graham♥[1]
David Heineman♥[1] Dirk Groeneveld♥[1] Faeze Brahman♥[1] Finbarr Timbers♥[1]
Hamish Ivison♥[1,2] Jacob Morrison♥[1,2] Jake Poznanski♥[1] Kyle Lo♥[1,2] Luca Soldaini♥[1]
Matt Jordan♥[1] Mayee Chen♥[1,4] Michael Noukhovitch♥[1,5,6] Nathan Lambert♥[1]
Pete Walsh♥[1] Pradeep Dasigi♥[1] Robert Berry♥[1] Saumya Malik♥[1] Saurabh Shah♥[1]
Scott Geng♥[1,2] Shane Arora♥[1] Shashank Gupta♥[1] Taira Anderson♥[1] Teng Xiao♥[1]
Tyler Murray♥[1] Tyler Romero♥[1] Victoria Graf♥[1,2]

Akari Asai[1,3] Akshita Bhagia[1] Alexander Wettig[7] Alisa Liu[2] Aman Rangapur[1]
Chloe Anastasiades[1] Costa Huang[1] Dustin Schwenk[1] Harsh Trivedi[1] Ian Magnusson[1,2]
Jaron Lochner[1] Jiacheng Liu[1] Lester James V. Miranda[1] Maarten Sap[1,3] Malia Morgan[1]
Michael Schmitz[1] Michal Guerquin[1] Michael Wilson[1] Regan Huff[1] Ronan Le Bras[1]
Rui Xin[2] Rulin Shao[2] Sam Skjonsberg[1] Shannon Zejiang Shen[8] Shuyue Stella Li[2]
Tucker Wilde[1] Valentina Pyatkin[1] Will Merrill[1] Yapei Chang[3] Yuling Gu[1] Zhiyuan Zeng[1,2]

Ashish Sabharwal[1] Luke Zettlemoyer[2] Pang Wei Koh[1,2]
Ali Farhadi[1,2] Noah A. Smith♥[1,2] Hannaneh Hajishirzi♥[1,2]

[1]Allen Institute for AI [2]University of Washington [3]Carnegie Mellon University [4]Stanford University [5]Mila
[6]Université de Montréal [7]Princeton University [8]Massachusetts Institute of Technology [9]University of Maryland

*Olmo 3 was a team effort; authors sorted alphabetically. ♥marks core contributors. See author contributions here.

🤗 **Olmo 3 Base:** `Olmo-3-1025-7B` `Olmo-3-1125-32B`
🤗 **Olmo 3 Think:** `Olmo-3-7B-Think` `Olmo-{3|3.1}-32B-Think`
🤗 **Olmo 3 Instruct:** `Olmo-3-7B-Instruct` `Olmo-3.1-32B-Instruct`
🤗 **Olmo 3 RL Zero:** `Olmo-3-7B-RL-Zero-{Math|Code|IF|General|Mix}` `Olmo-3.1-7B-RL-Zero-{Math|Code}`

🏭 **Base Data:** `Pretrain: Dolma 3 Mix` `Midtrain: Dolma 3 Dolmino Mix` `Long-ctx: Dolma 3 Longmino Mix`
🏭 **Think Data:** `Dolci-Think-{SFT|DPO|RL}-7B` `Dolci-Think-{SFT|DPO|RL}-32B`
🏭 **Instruct Data:** `Dolci-Instruct-{SFT|DPO|RL}`
🏭 **RL-Zero Data:** `Dolci-RL-Zero-{Math|Code|IF|General}-7B` `Dolci-RL-Zero-Mix-7B`

Training Code: `OLMo-core` (pretrain) `Open Instruct` (posttrain)
Data Code: `datamap-rs` (data processing) `duplodocus` (deduplication) `dolma3` (data recipes)
Eval Code: `OLMES` (eval suite) `decon` (eval decontamination)

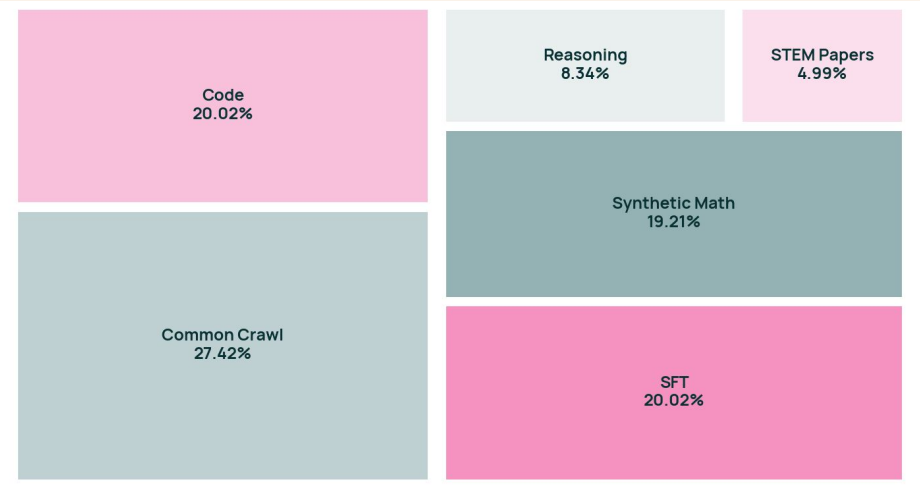
Training Logs: `Olmo-3-7B-{Base|Think|Instruct|RL-Zero}` `Olmo-3-32B-{Base|Think|Instruct}`
Demo: 32B Think 32B Instruct 7B Think 7B Instruct
Contact: `olmo@allenai.org`

Abstract

We introduce **Olmo 3**, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. Olmo 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire **model flow**, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, **Olmo 3.1 Think 32B**, is the strongest fully-open thinking model released to-date.

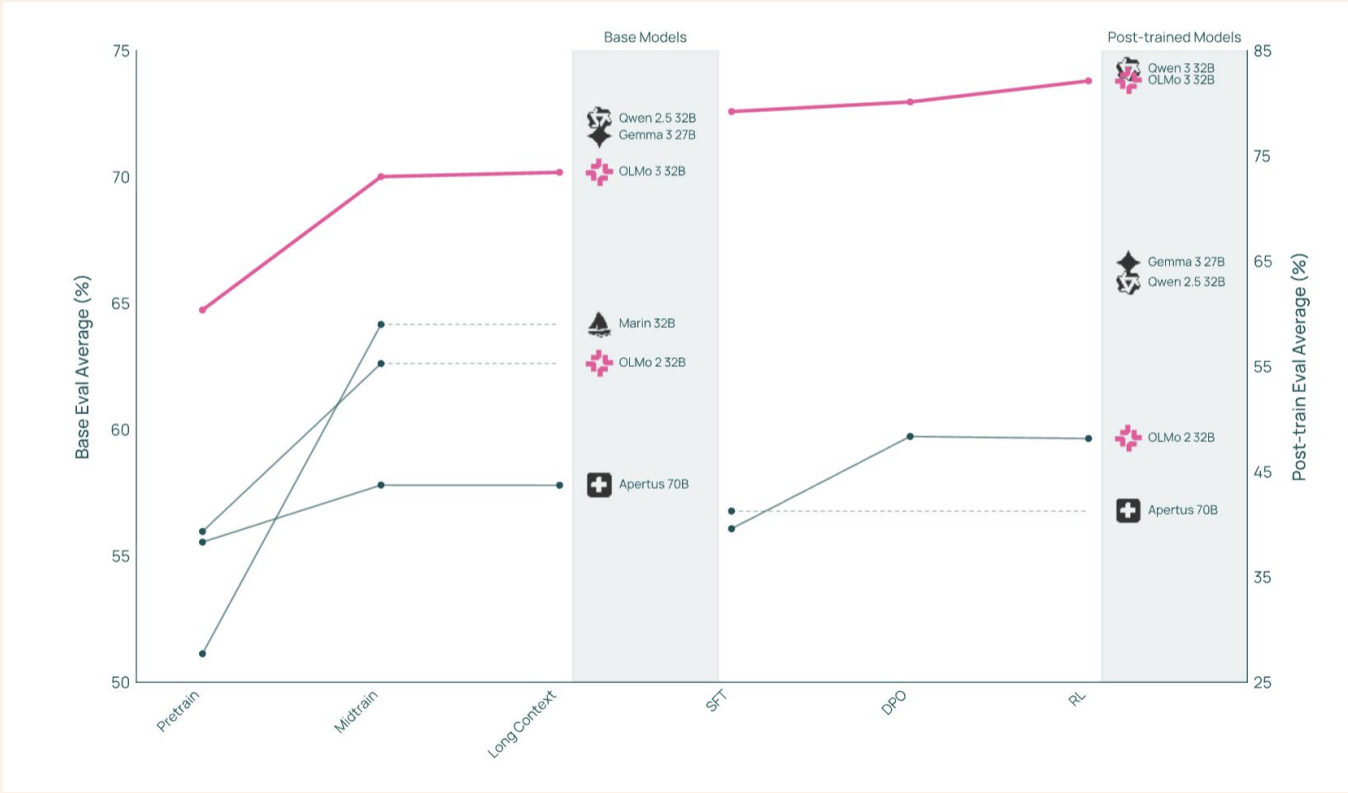
Ai2

Midtraining Olmo 3



Boost data that closely match post-train capability targets

Olmo 3 Model Flow



Pretraining → Midtraining → Long context → Thinking SFT → Thinking DPO → RLVR

Ai2

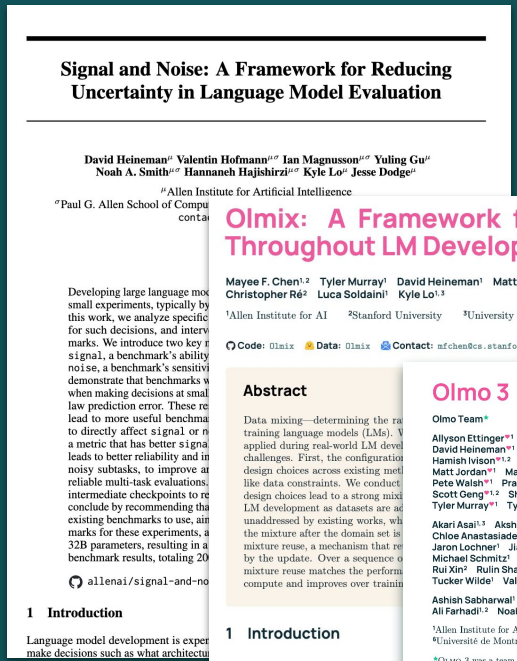
Thank you!

Pretraining → Midtraining → Long Ctx → SFT → Pref. Tuning → RLVR

Twitter	@heinemandavidj
Bluesky	@davidheineman.com
Email	davidh@allenai.org

... and many more (ordered arbitrarily)

Ai2 · PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING · UWNLP



Signal and Noise: A Framework for Reducing Uncertainty in Language Model Evaluation

David Heineman[μ] Valentin Hofmann[μσ] Ian Magnusson[μσ] Yuling Gu[μ]
Noah A. Smith[μσ] Hannaneh Hajishirzi[μσ] Kyle Lo[μ] Jesse Dodge[μ]

[μ]Allen Institute for Artificial Intelligence
[σ]Paul G. Allen School of Compu
conta

Developing large language mo
small experiments, typically by
this work, we analyze specific
for such decisions, and interv
marks. We introduce two key
signal, a benchmark's ability
noise, a benchmark's sensitivi
demonstrate that benchmarks w
when making decisions at smal
law prediction error. These re
lead to more useful benchma
to directly affect signal or n
a metric that has better signa
leads to better reliability and i
noisy subtasks, to improve a
reliable multi-task evaluations,
intermediate checkpoints to re
conclude by recommending tha
existing benchmarks to use, ai
marks for these experiments, a
32B parameters, resulting in a
benchmark results, totaling 20

allenai/signal-and-no

1 Introduction

Language model development is exper
make decisions such as what architectu

Olmix: A Framework for Data Mixing Throughout LM Development

Mayee F. Chen[1,2] Tyler Murray[1] David Heineman[1] Matt Jordan[1] Hannaneh Hajishirzi[1,3]
Christopher Ré[2] Luca Soldaini[1] Kyle Lo[1,3]

[1]Allen Institute for AI [2]Stanford University [3]University of Washington

Code: Olmix Data: Olmix Contact: mfchen@cs.stanford.edu {lucas,kylel}@allenai.org

Abstract

Data mixing—determining the ra
training language models (LMs).
applied during real-world LM deve
challenges. First, the configuratio
design choices across existing met
like data constraints. We conduct
design choices lead to a strong mix
LM development as datasets are ad
unaddressed by existing works, wh
the mixture after the domain set is
mixture reuse, a mechanism that re
by the update. Over a sequence o
mixture reuse matches the perform
compute and improves over trainin

1 Introduction

Figure 1 Two problems with data mixing
method? (2) How to efficiently mix under

Modern language models (LMs) are tr

Olmo 3

Olmo Team*

Allyson Ettinger[♥1] Amanda Bertsch[♥1,3] Bailey Kuehl[♥1] David Graham[♥1]
David Heineman[♥1] Dirk Groeneveld[♥1] Faeze Brahman[♥1] Finbarr Timbers[♥1]
Hamish Ivison[♥1,2] Jacob Morrison[♥1,2] Jake Poznanski[♥1] Kyle Lo[♥1,2] Luca Soldaini[♥1]
Matt Jordan[♥1] Mayee Chen[♥1,4] Michael Noukhovitch[♥1,5,6] Nathan Lambert[♥1]
Pete Walsh[♥1] Pradeep Dasigi[♥1] Robert Berry[♥1] Saumya Malik[♥1] Saurabh Shah[♥1]
Scott Geng[♥1,2] Shane Arora[♥1] Shashank Gupta[♥1] Taira Anderson[♥1] Teng Xiao[♥1]
Tyler Murray[♥1] Tyler Romero[♥1] Victoria Graf[♥1,2]

Akari Asai[1,3] Akshita Bhagia[1] Alexander Wettig[7] Alisa Liu[2] Aman Rangapur[1]
Chloe Anastasiades[1] Costa Huang[1] Dustin Schwenk[1] Harsh Trivedi[1] Ian Magnusson[1,2]
Jaron Lochner[1] Jiacheng Liu[1] Lester James V. Miranda[1] Maarten Sap[1,3] Malia Morgan[1]
Michael Schmitz[1] Michal Guerquin[1] Michael Wilson[1] Regan Huff[1] Ronan Le Bras[1]
Rui Xin[2] Rulin Shao[2] Sam Skjonsberg[1] Shannon Zejiang Shen[8] Shuyue Stella Li[2]
Tucker Wilde[1] Valentina Pyatkin[1] Will Merrill[1] Yapei Chang[9] Yuling Gu[1] Zhiyuan Zeng[1,2]

Ashish Sabharwal[1] Luke Zettlemoyer[2] Pang Wei Koh[1,2]
Ali Farhadi[1,2] Noah A. Smith[♥1,2] Hannaneh Hajishirzi[♥1,2]

[1]Allen Institute for AI [2]University of Washington [3]Carnegie Mellon University [4]Stanford University [5]Mila
[6]Université de Montréal [7]Princeton University [8]Massachusetts Institute of Technology [9]University of Maryland

*Olmo 3 was a team effort; authors sorted alphabetically. ♥marks core contributors. See author contributions here.

Olmo 3 Base: Olmo-3-1025-7B Olmo-3-1125-32B
Olmo 3 Think: Olmo-3-7B-Think Olmo-{3|3.1}-32B-Think
Olmo 3 Instruct: Olmo-3-7B-Instruct Olmo-3.1-32B-Instruct
Olmo 3 RL Zero: Olmo-3-7B-RL-Zero-{Math|Code|IF|General|Mix} Olmo-3.1-7B-RL-Zero-{Math|Code}

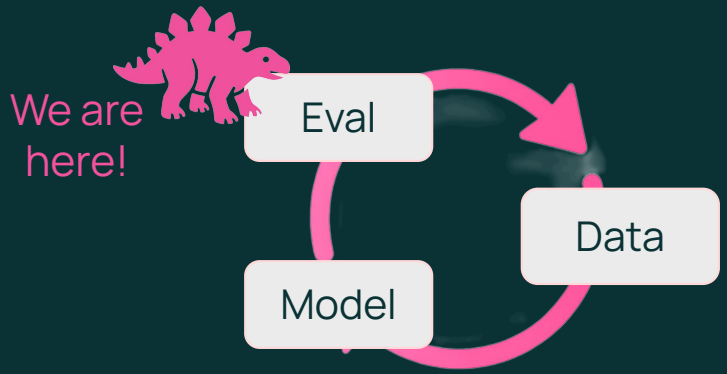
Base Data: Pretrain: Dolma 3 Mix Midtrain: Dolma 3 Dolmino Mix Long-ctx: Dolma 3 Longmino Mix
Think Data: Dolci-Think-{SFT|DPO|RL}-7B Dolci-Think-{SFT|DPO|RL}-32B
Instruct Data: Dolci-Instruct-{SFT|DPO|RL}
RL-Zero Data: Dolci-RL-Zero-{Math|Code|IF|General}-7B Dolci-RL-Zero-Mix-7B

Training Code: OLMo-core (pretrain) Open Instruct (posttrain)
Data Code: datamap-rs (data processing) duplodocus (deduplication) dolma3 (data recipes)
Eval Code: OLMES (eval suite) decon (eval decontamination)
Training Logs: Olmo-3-7B-{Base|Think|Instruct|RL-Zero} Olmo-3-32B-{Base|Think|Instruct}
Demo: 32B Think 32B Instruct 7B Think 7B Instruct
Contact: olmo@allenai.org

Abstract

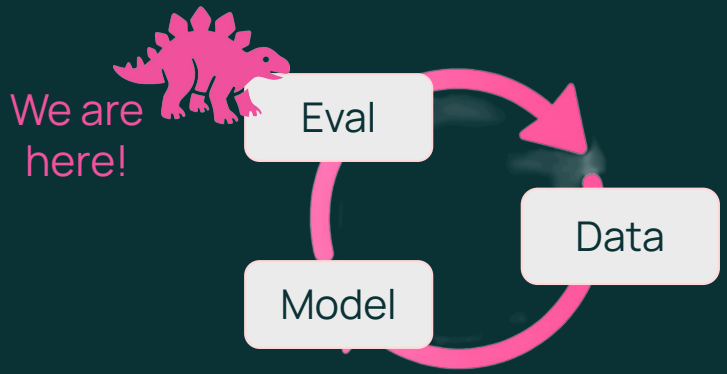
Ai2

We introduce **Olmo 3**, a family of state-of-the-art, fully-open language models at the 7B and 32B parameter scales. Olmo 3 model construction targets long-context reasoning, function calling, coding, instruction following, general chat, and knowledge recall. This release includes the entire **model flow**, i.e., the full lifecycle of the family of models, including every stage, checkpoint, data point, and dependency used to build it. Our flagship model, **Olmo 3.1 Think 32B**, is the strongest fully-open thinking model released to-date.



› Evaluation Methods

› Evaluation in Olmo 3



› Evaluation Methods

› Evaluation in Olmo 3

› The Olmo 3 Models

› Evaluation for LMs

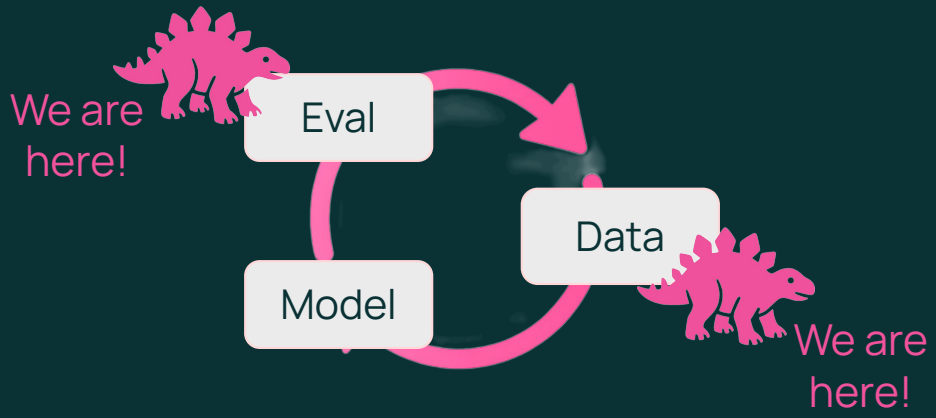
› Methods & Decision Making

› Model Selection in Olmo 3

› Data mixing with Olmix

- The Olmo 3 Models
- Evaluation for LMs
- **Methods & Decision Making**
- Model Selection in Olmo 3
- Data mixing with Olmix

- The Olmo 3 Models
- Evaluation for LMs
- Methods & Decision Making
- Model Selection in Olmo 3
- Data mixing with Olmix

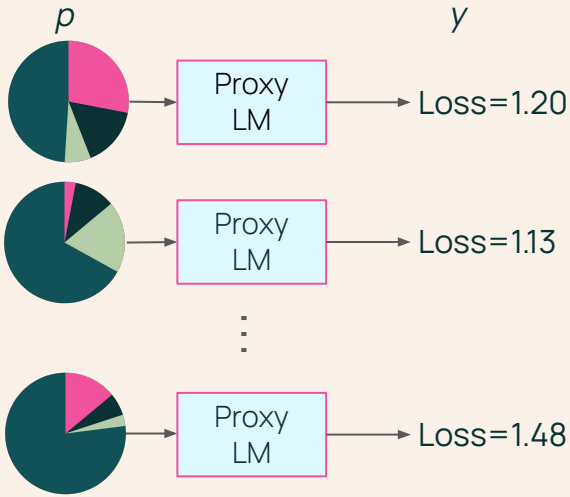


› Data Mixing with Olmix

› Data for Olmo 3

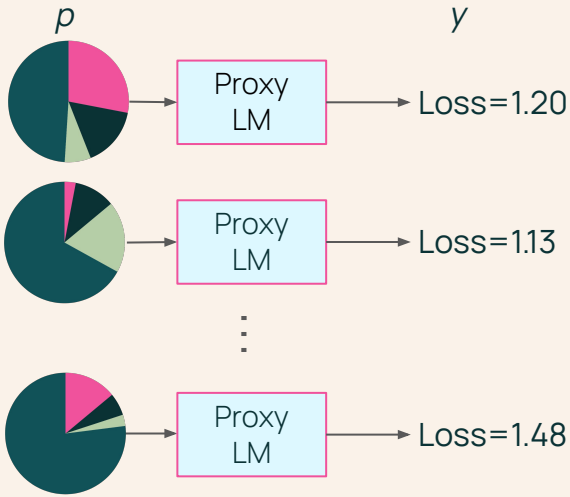
Data mixing

1. **Swarm**: Train K small models with randomly sampled mixtures p



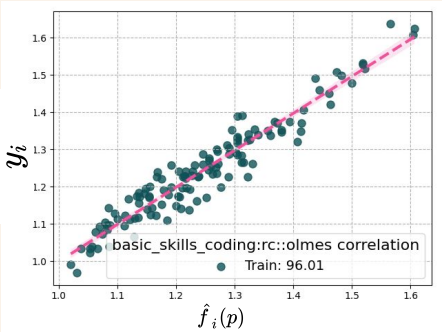
Data mixing

1. **Swarm**: Train *K* small models with randomly sampled mixtures *p*



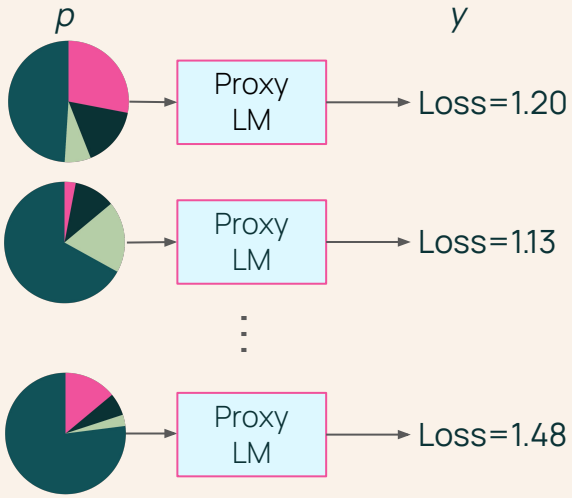
2. **Regression:** Fit function to predict LM performance given mixture *p*

$$\hat{f}_i(p) \approx y_i$$



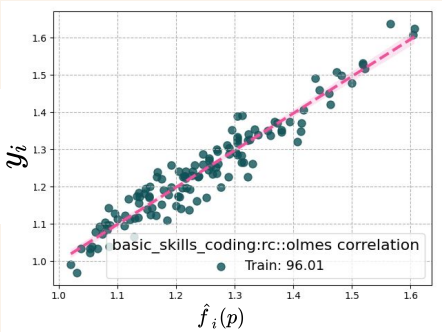
Data mixing

1. **Swarm**: Train *K* small models with randomly sampled mixtures *p*



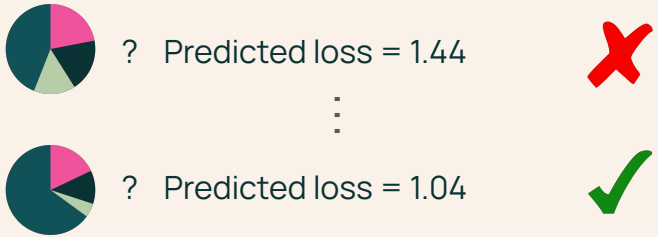
2. **Regression:** Fit function to predict LM performance given mixture *p*

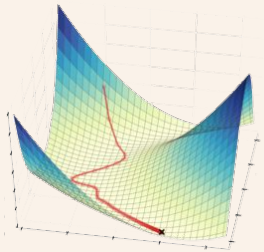
$$\hat{f}_i(p) \approx y_i$$



3. **Optimize:** Use fit function to solve for optimal mix p*

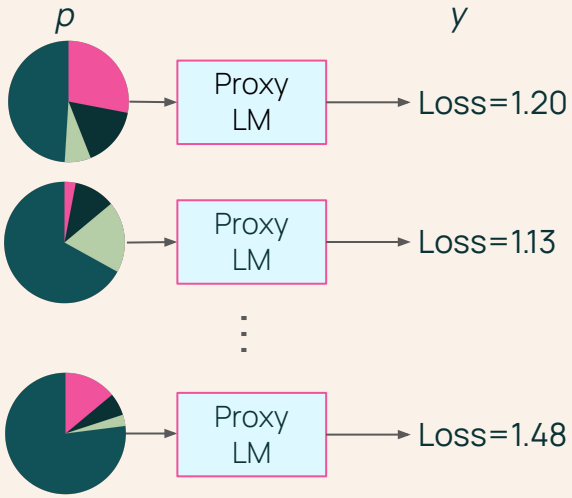
$$\underset{p \in \triangle^{m-1}}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} \hat{f}_i(p)$$





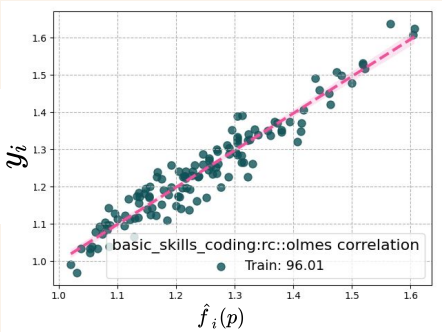
Data mixing

1. **Swarm**: Train K small models with randomly sampled mixtures p



2. **Regression:** Fit function to predict LM performance given mixture p

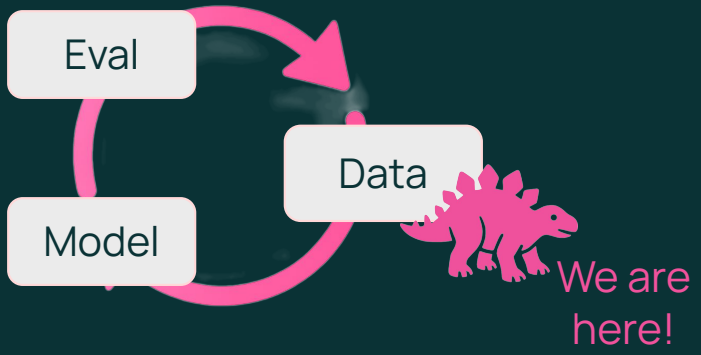
$$\hat{f}_i(p) \approx y_i$$



3. **Optimize:** Use fit function to solve for optimal mix p*

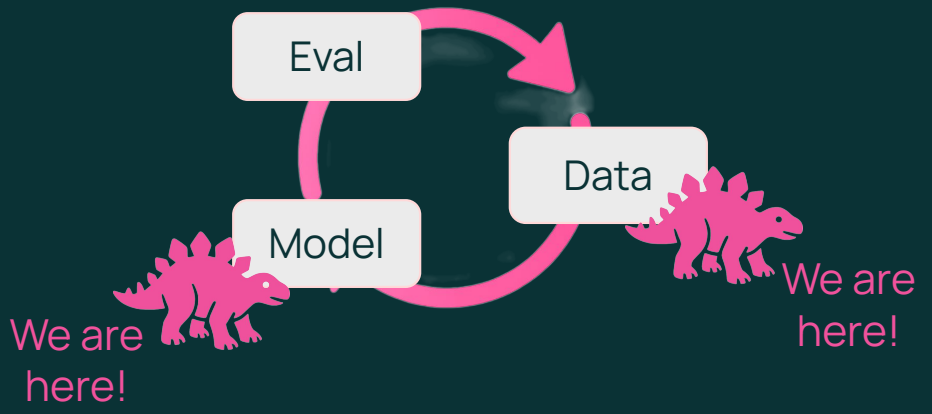
$$\underset{p \in \triangle^{m-1}}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} \hat{f}_i(p)$$

Ai2



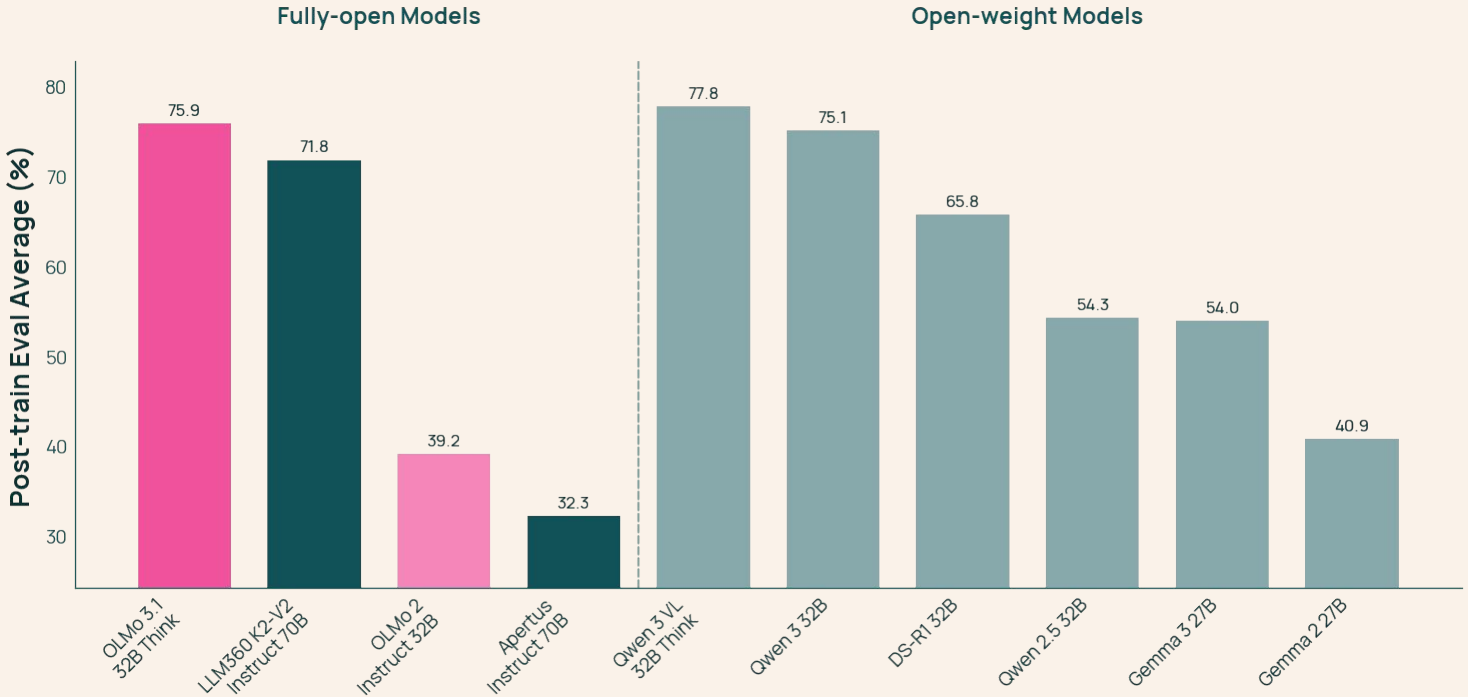
› Data Mixing with Olmix

› Data for Olmo 3



› Putting it together: Olmo 3

Olmo 3 Model Flow



Olmo 3 Model Flow