Tutorial: LLM Data Mixing

Mayee Chen, Stanford University
mfchen@stanford.edu

Curating good training data is critical to the performance of LLMs.

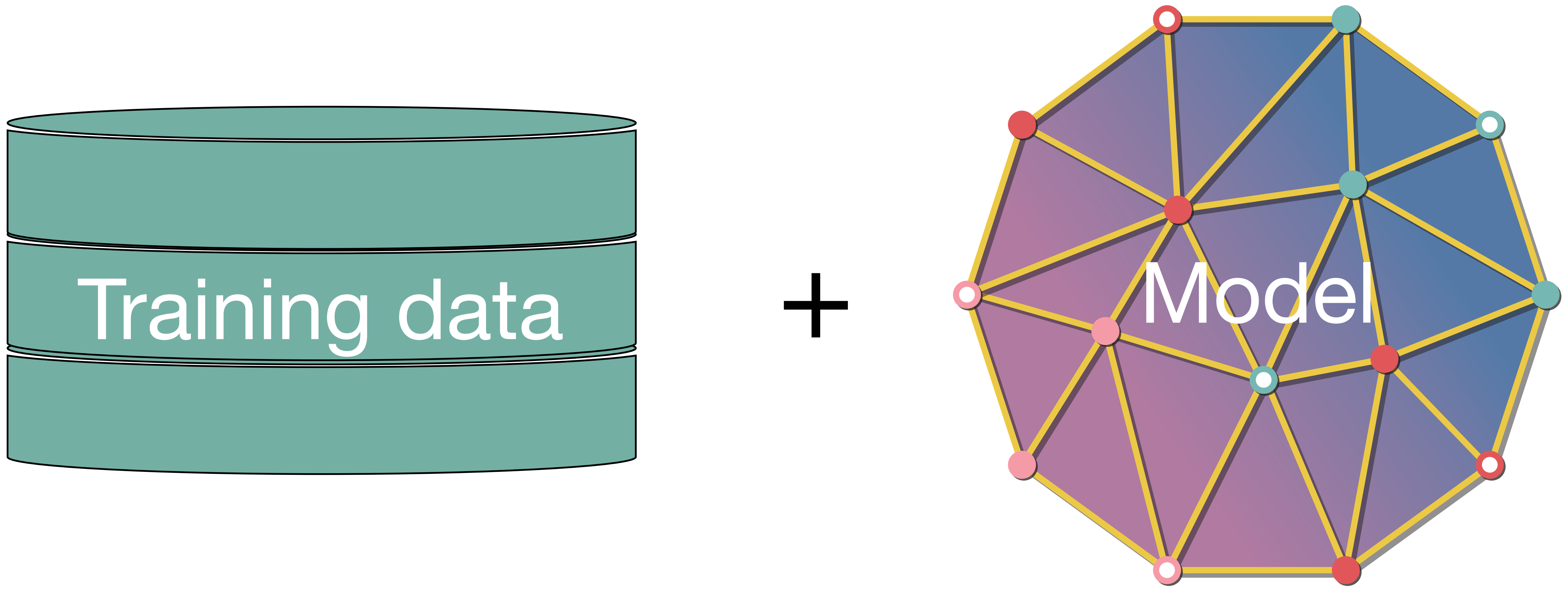
- Latest open-source LLMs are trained on 30+ trillion tokens of data (Qwen 3)

- Every frontier lab has data teams constantly working on designing new training datasets

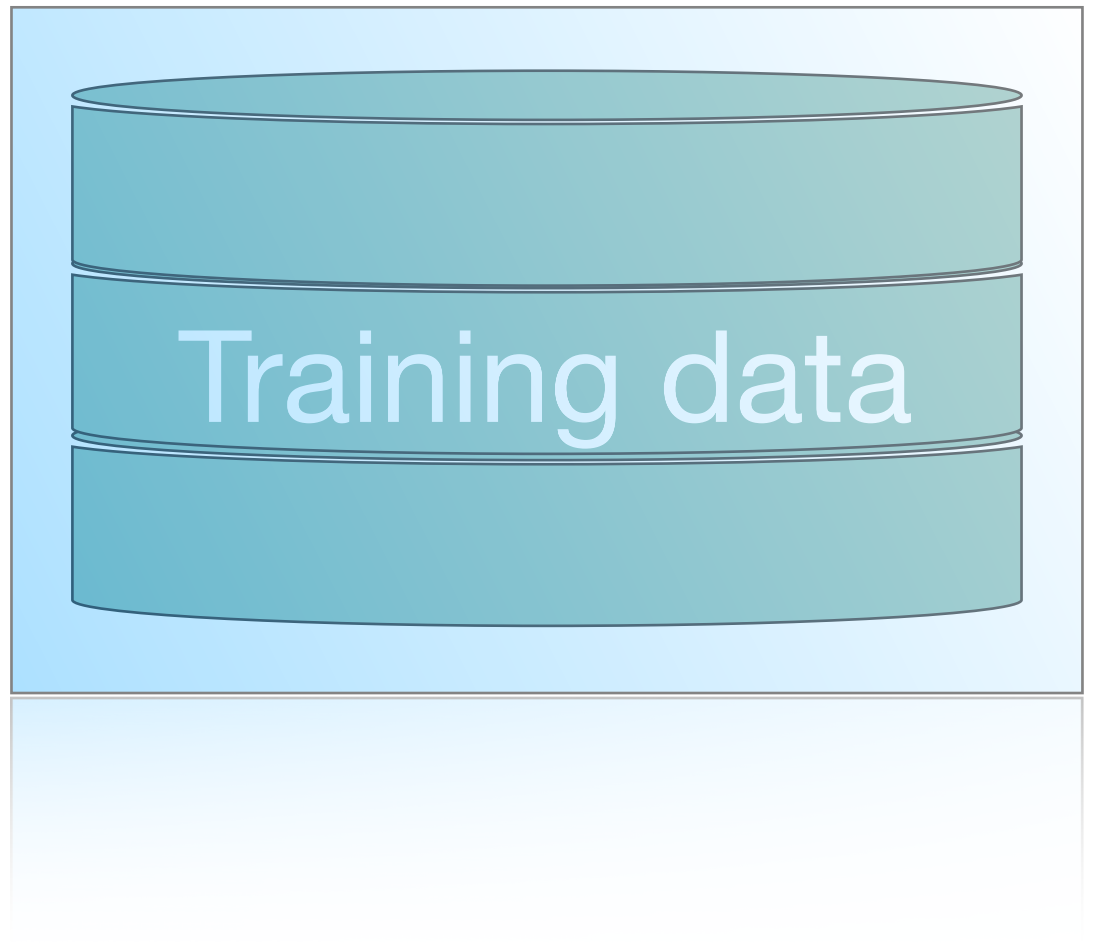
- “To train the best language model, the curation of a large, high-quality training dataset is paramount. In line with our design principles, we invested heavily in pretraining data.” - Llama3 blog

- How did we get here?

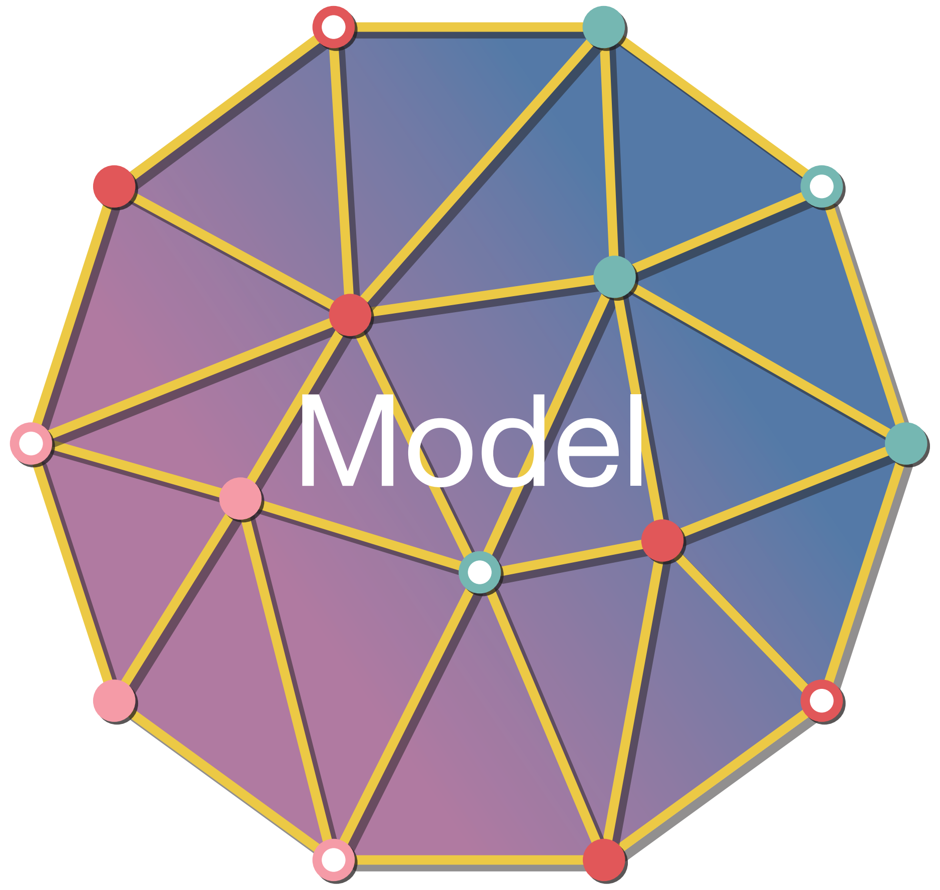
The role of Data in AI



The role of Data in AI



+

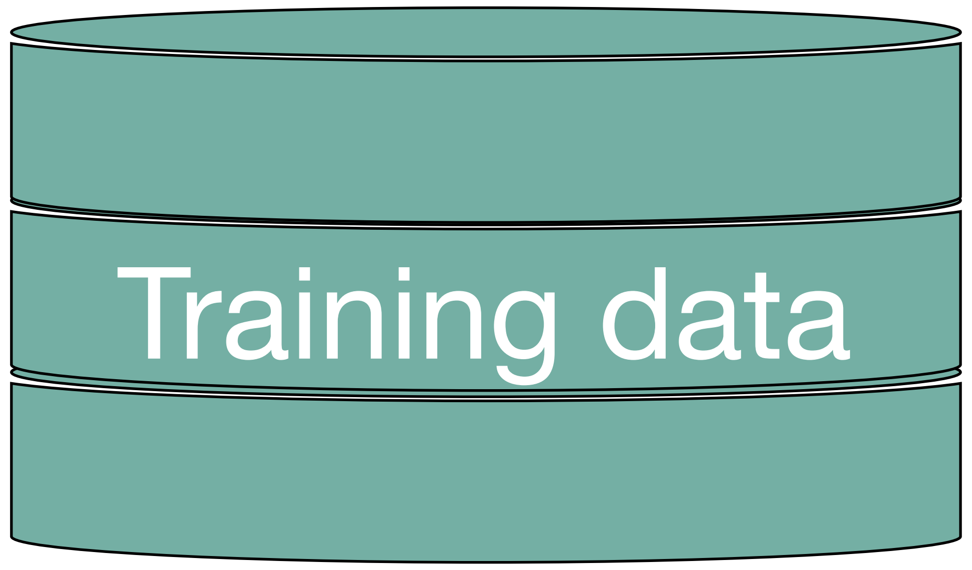


Model-centric AI:

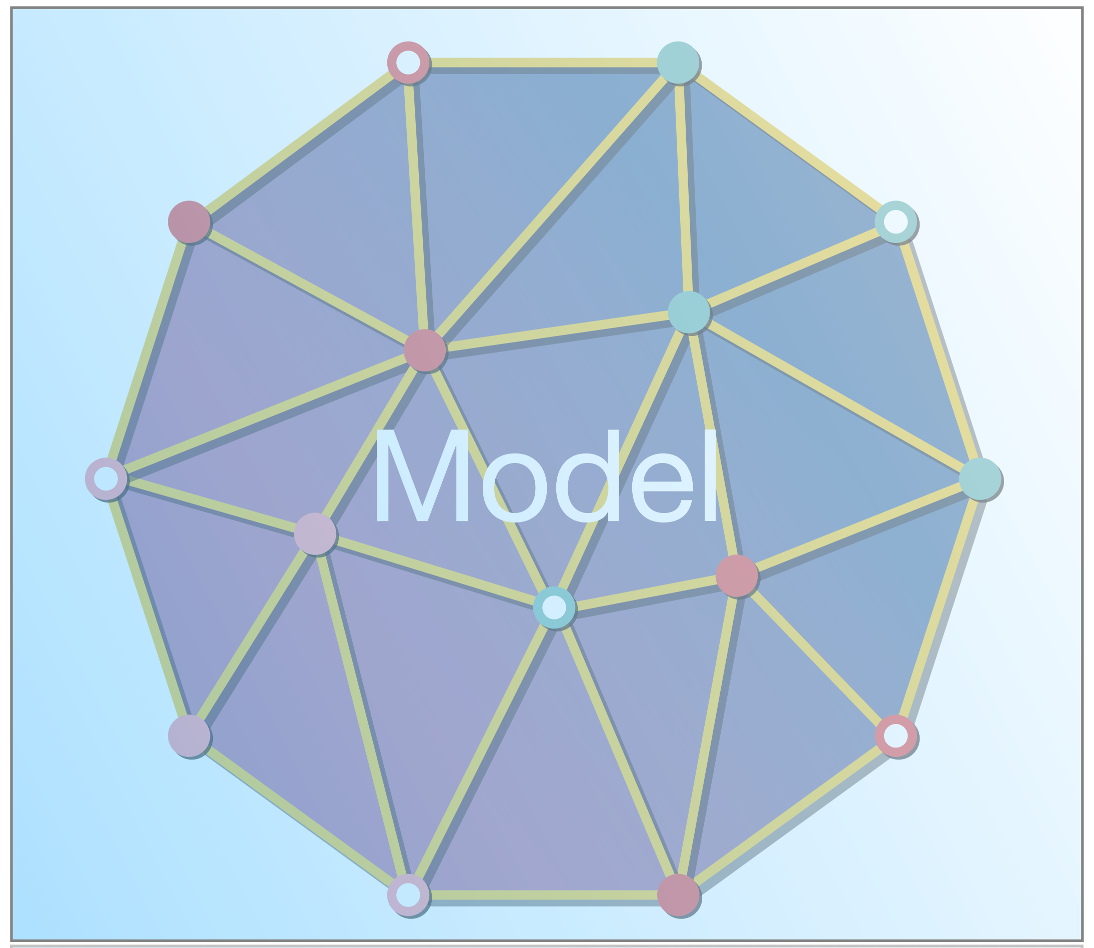
Improve architectures, optimisers, training

Hold data constant

The role of Data in AI



+



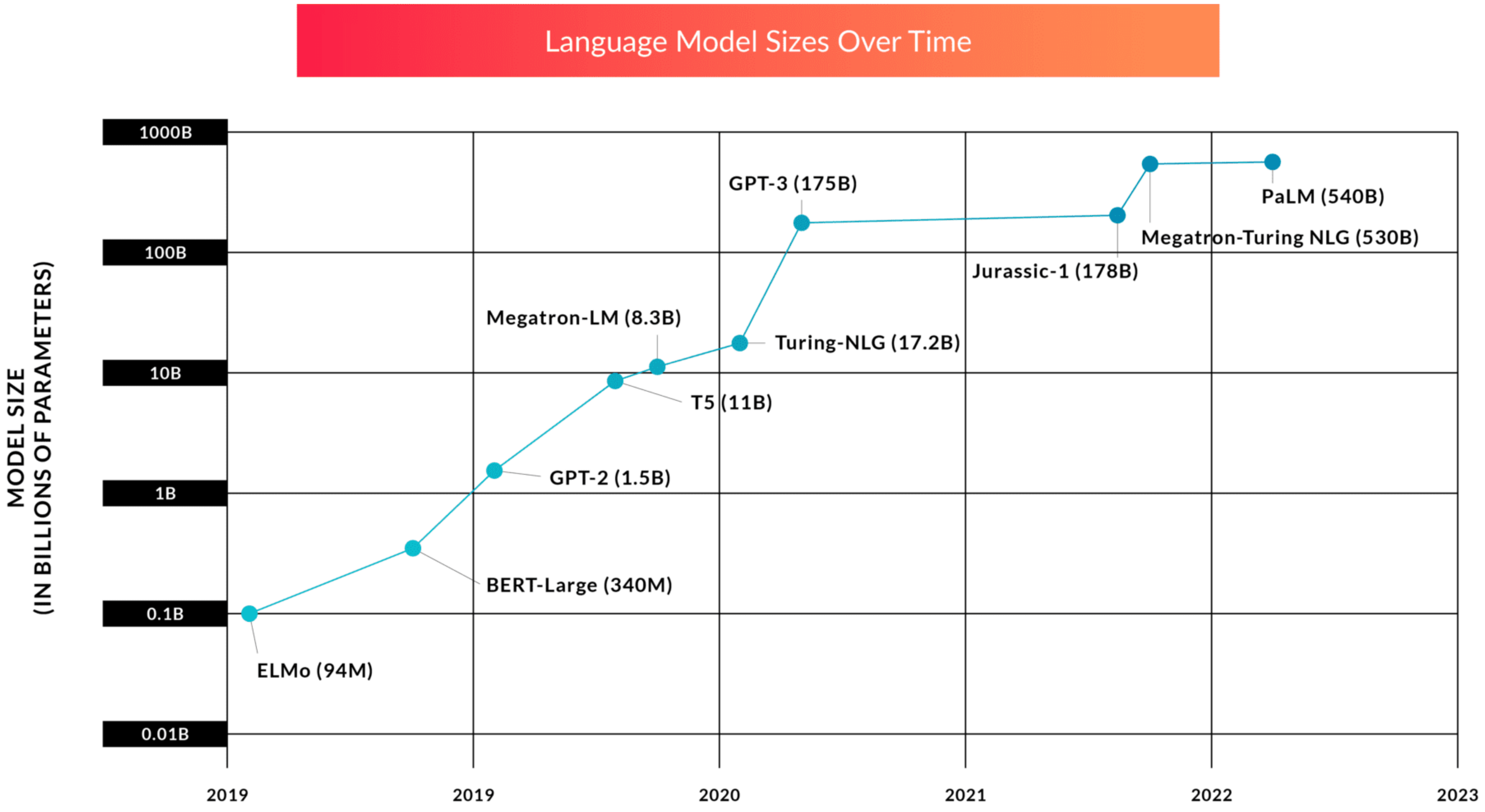
Data-Centric AI:

Improve data quality

Hold model/training constant

Late 2010s, early 2020s

Early LLMs: scaling



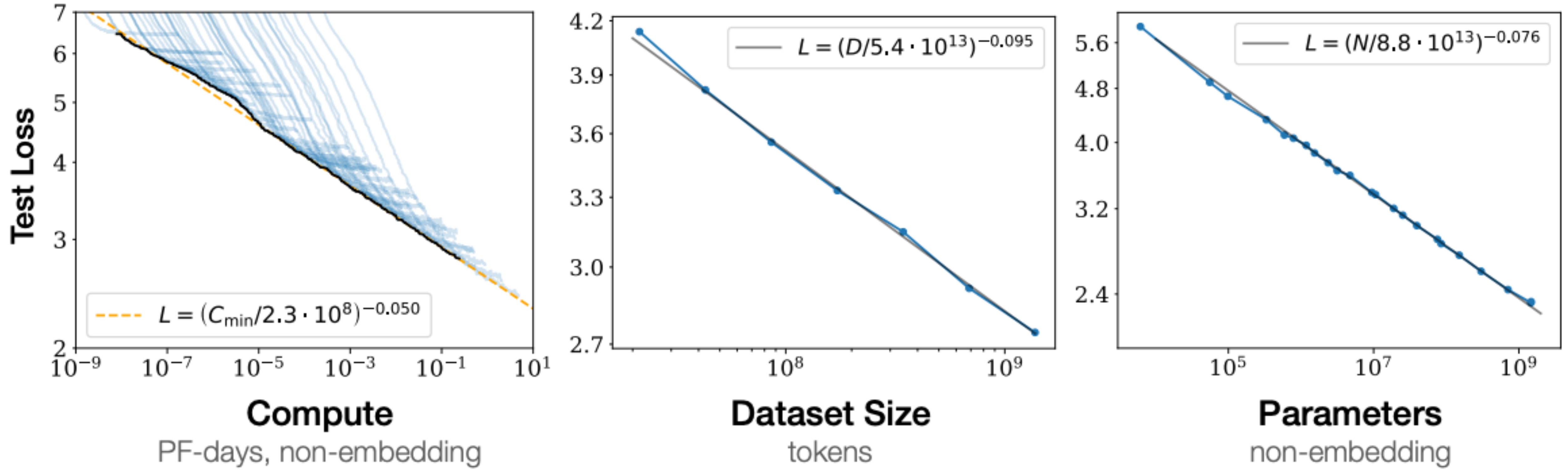


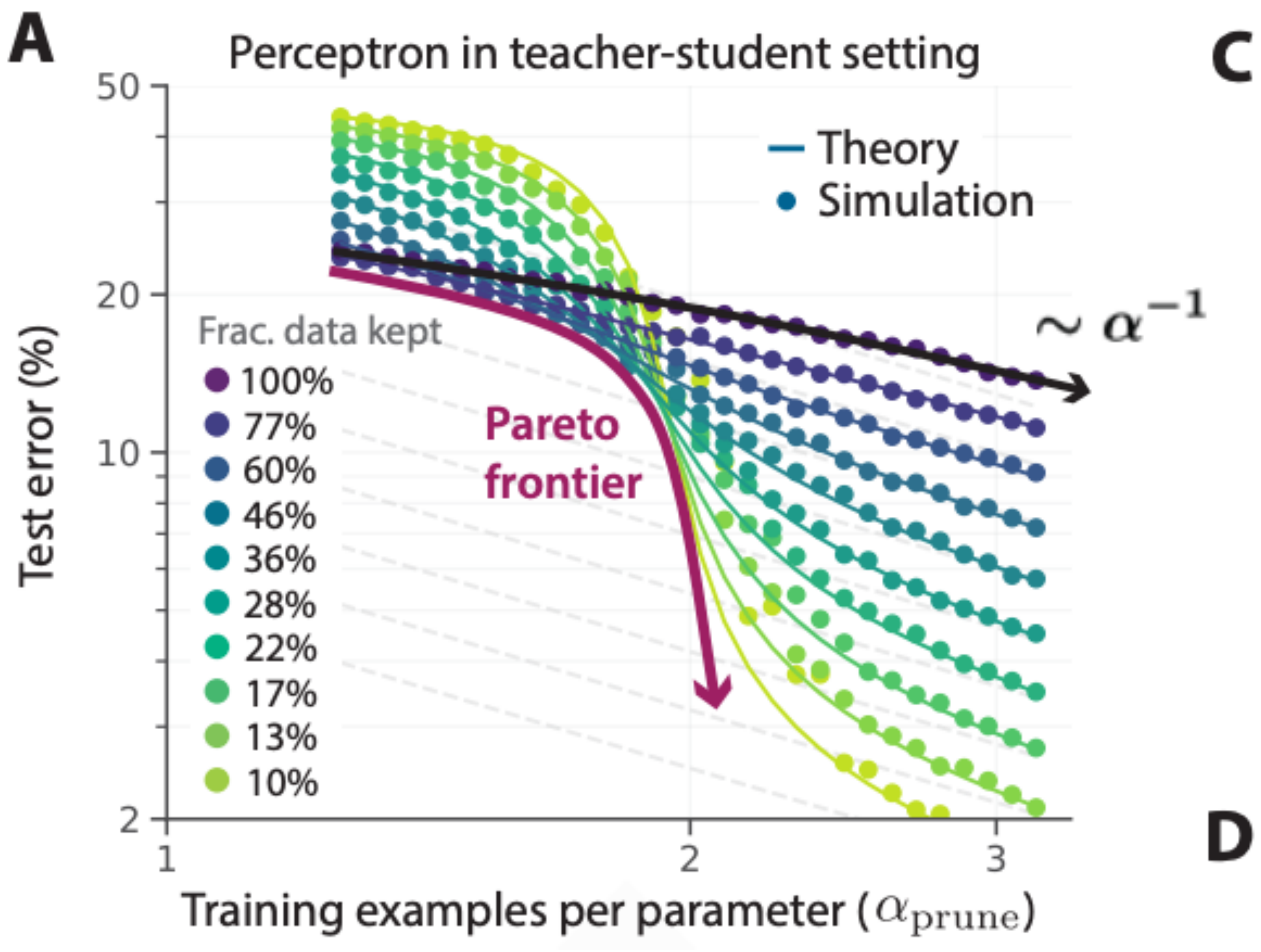
Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

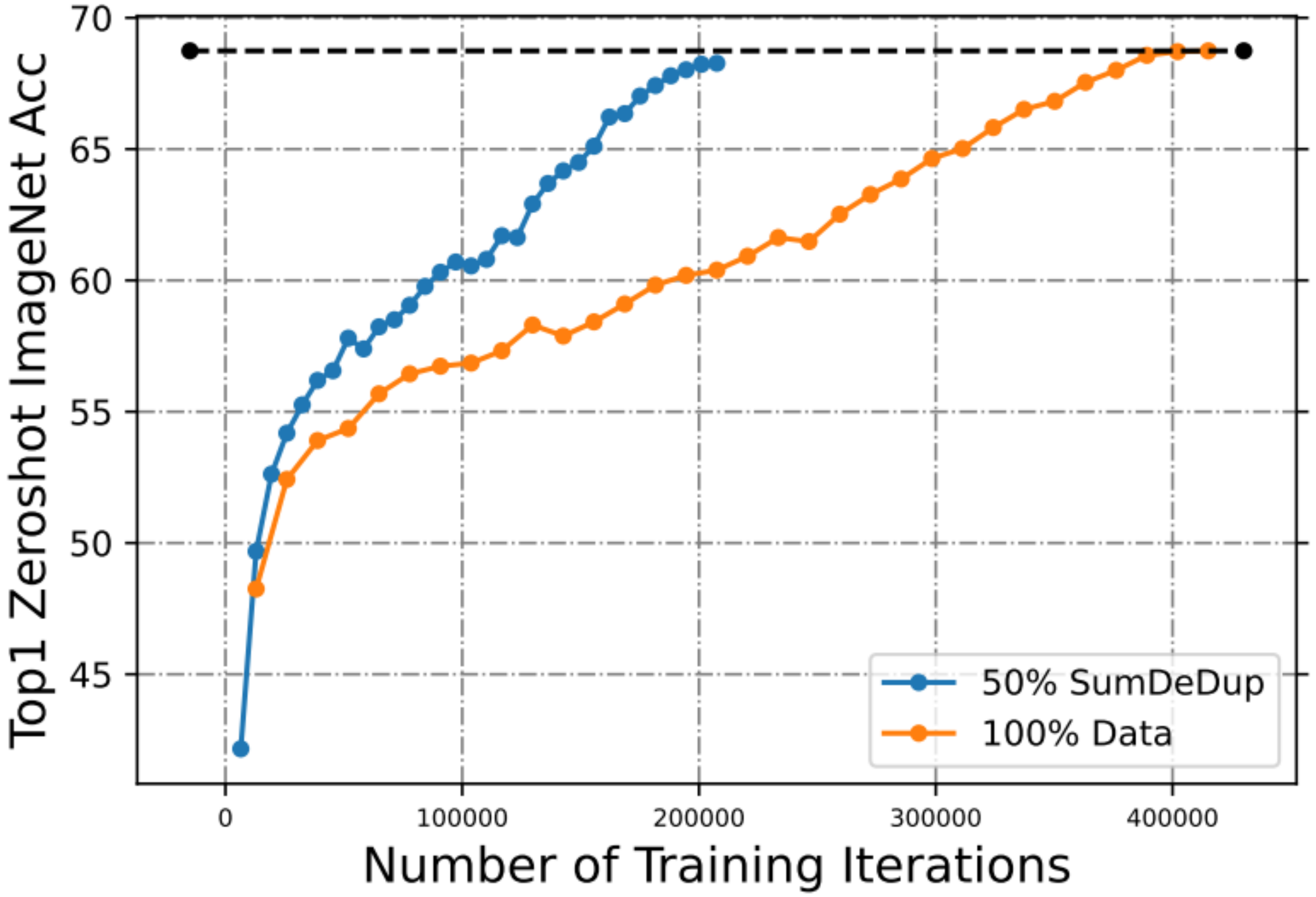
WIKIPEDIA
The Free Encyclopedia

COMMON CRAWL

https://futuredirections.ieee.org/2023/04/24/how-much-bigger-can-should-llms-become/
Kaplan et. al., 2020. Scaling Laws for Neural Language Models.

LLMs and Data: quantity is not everything





Takeaway: today, people widely accept that having good LLM training data is just as important as having a lot of it

Sorcher et al., 2022. Beyond neural scaling laws: beating power law scaling via data pruning.
Abbas et al., 2023. SemDeDup: Data-efficient learning at web-scale through semantic deduplication.

A bit about me

- LLM data researcher, final-year PhD student at Stanford advised by Christopher Re (I am not a roboticist :))
- Developed algorithms for data labelling, data curriculum, data mixing, synthetic data
- Partnered with Snorkel AI, Together AI, AI2, involved in creation of several LLMs and their training datasets (e.g., DCLM)

Outline

- **The LLM data development pipeline**
  - What makes good data?
  - How do you create a good dataset?
- **Deep dive into data mixing**
  - Key development: Mixing laws
  - Case study: two methods that utilise mixing laws
  - Implications of Mixing Laws: improving understanding

The LLM Data Development Pipeline

What makes a good LLM training dataset?

Quantity (# of tokens)

Quality (sample-level properties)

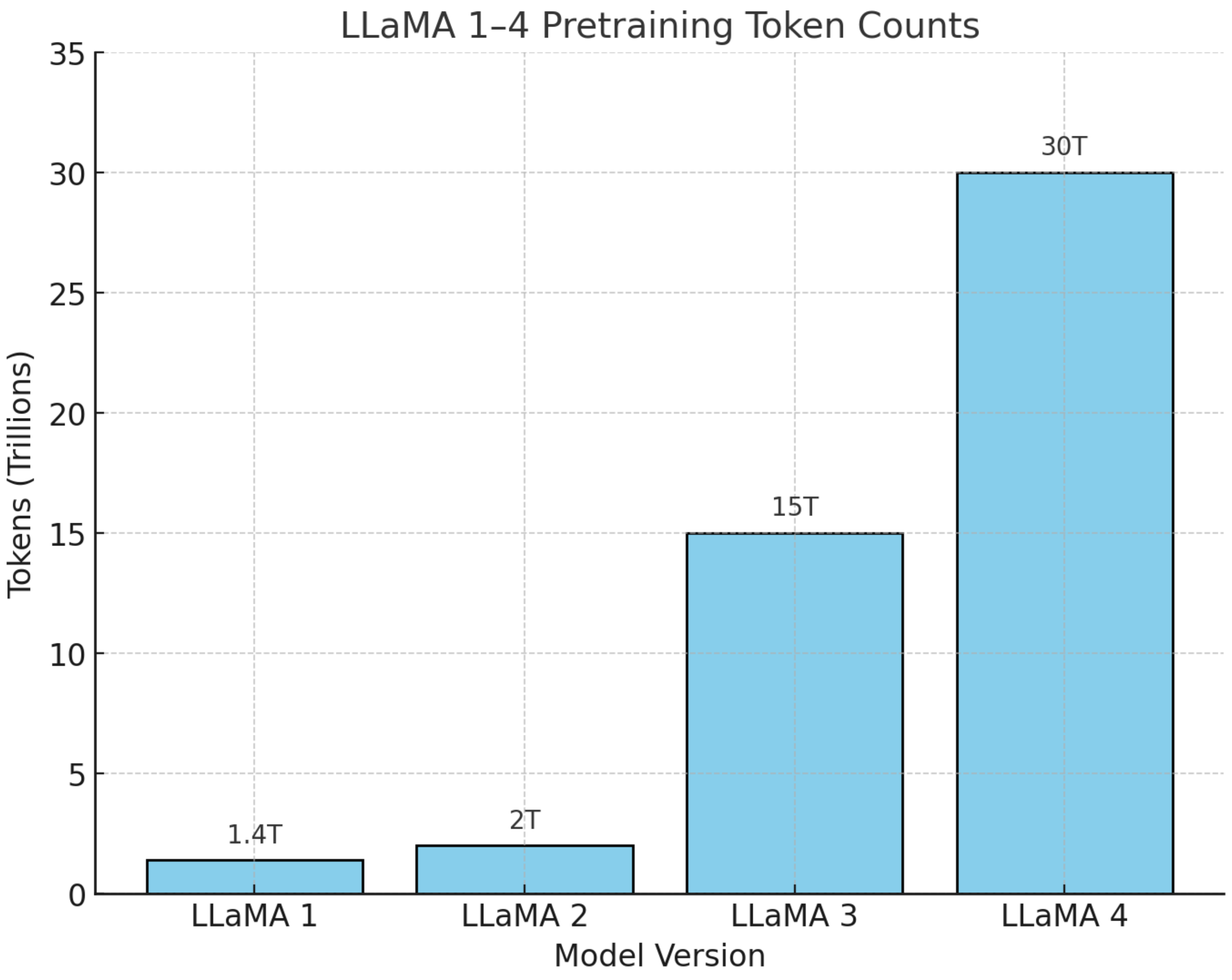
Composition (dataset-level properties)

What makes a good LLM training dataset?

Quantity (# of tokens)

Quality (sample-level properties)

Composition (dataset-level properties)



What makes a good LLM training dataset?

Quantity (# of tokens)

Quality (sample-level properties)

Composition (dataset-level properties)

FineWeb-Edu Score: 2/5

Well, these are still some difficult questions to answer with pin-point accuracy, and at this point I don't believe anyone has the exact answer to all 3 of these questions. What I offer below is a mix of what I Think, What I know and what Appears to be.... Anyone currently attempting to answer these questions with some type of

FineWeb-Edu Score: 4/5

A vaccine is a biological preparation that improves immunity to a particular disease. A vaccine typically contains an agent that resembles a disease-causing microorganism, and is often made from weakened or killed forms of the microbe, its toxins or one of its surface proteins.

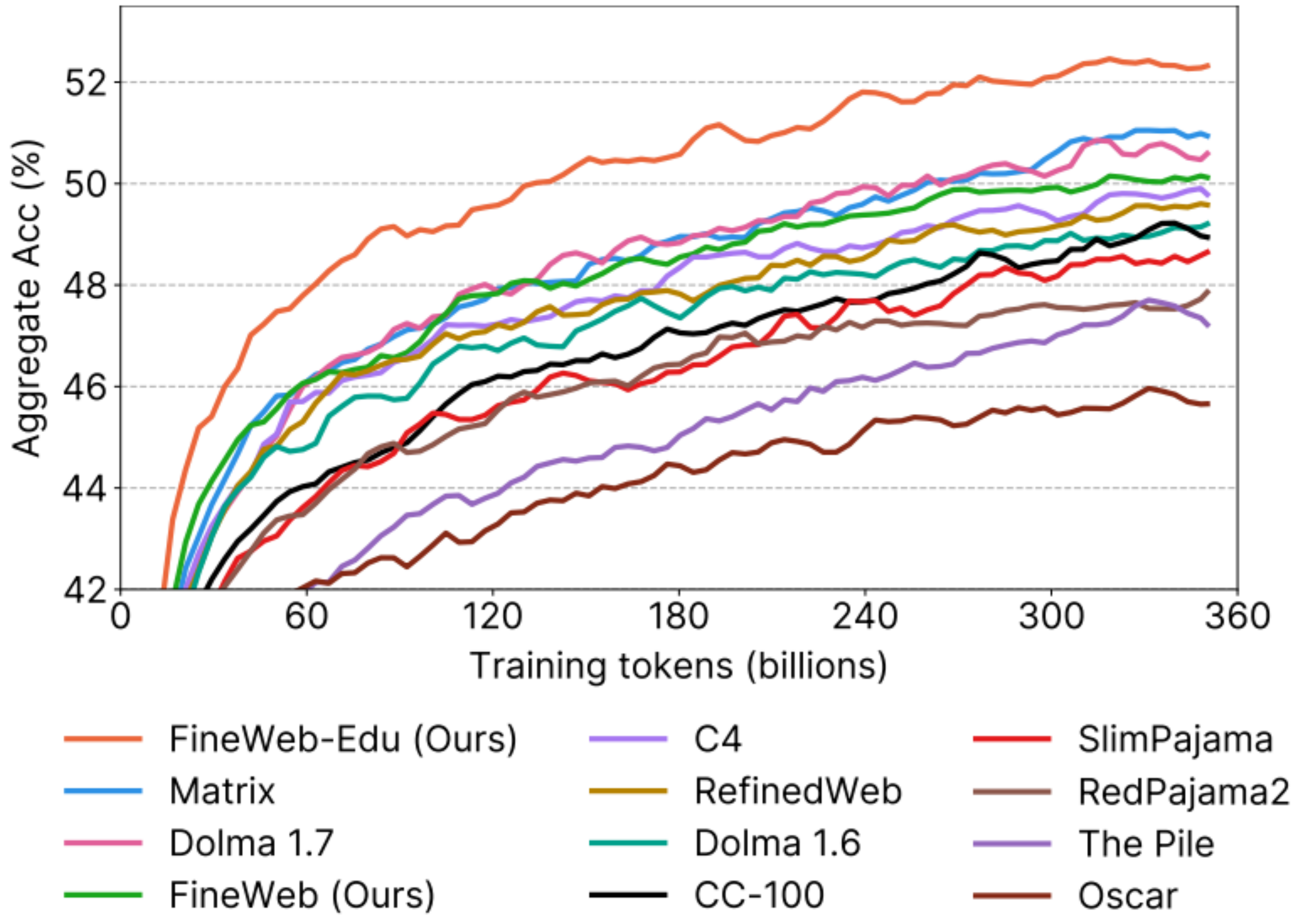


Figure 10: **Comparing FineWeb datasets to other public datasets.** Base FineWeb shows strong performance, with the educational subset (FineWeb-Edu) surpassing all other public datasets and further enhancing the aggregate score by approximately 2%.

Penedo et al., 2024. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale

What makes a good LLM training dataset?

Quantity (# of tokens)

Quality (sample-level properties)

Composition (dataset-level properties)

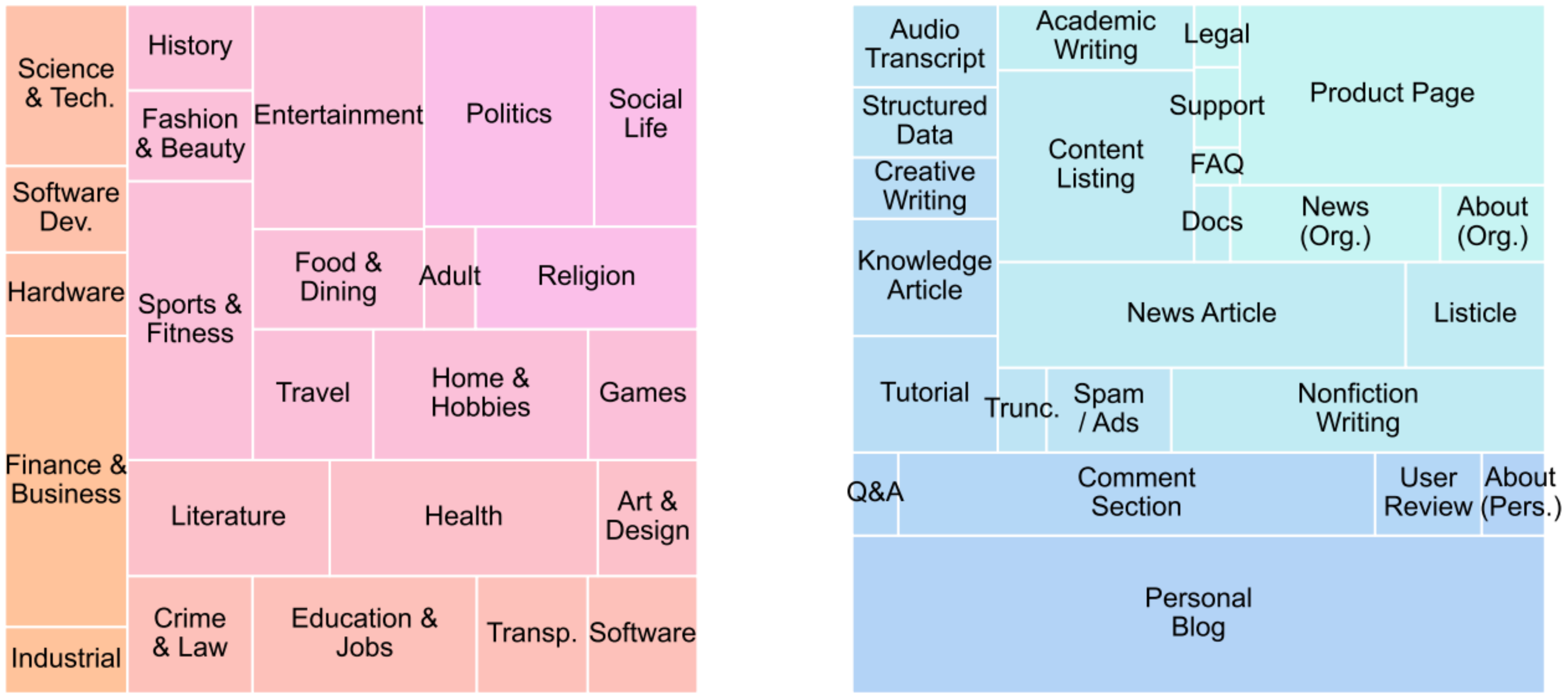


Figure 1: We construct **topic domains** (left) and **format domains** (right) to organize pre-training corpora. The areas visualize the number of tokens per domain in a cleaned pre-training corpus based on CommonCrawl. See Appendix A for detailed definitions of the categories. We provide an interactive explorer of the domains at weborganizer.allen.ai.

LLM that can do many things:

📝 summarise documents

🧑‍💻 write code

🔢 solve math problems

💬 chat with users in many languages

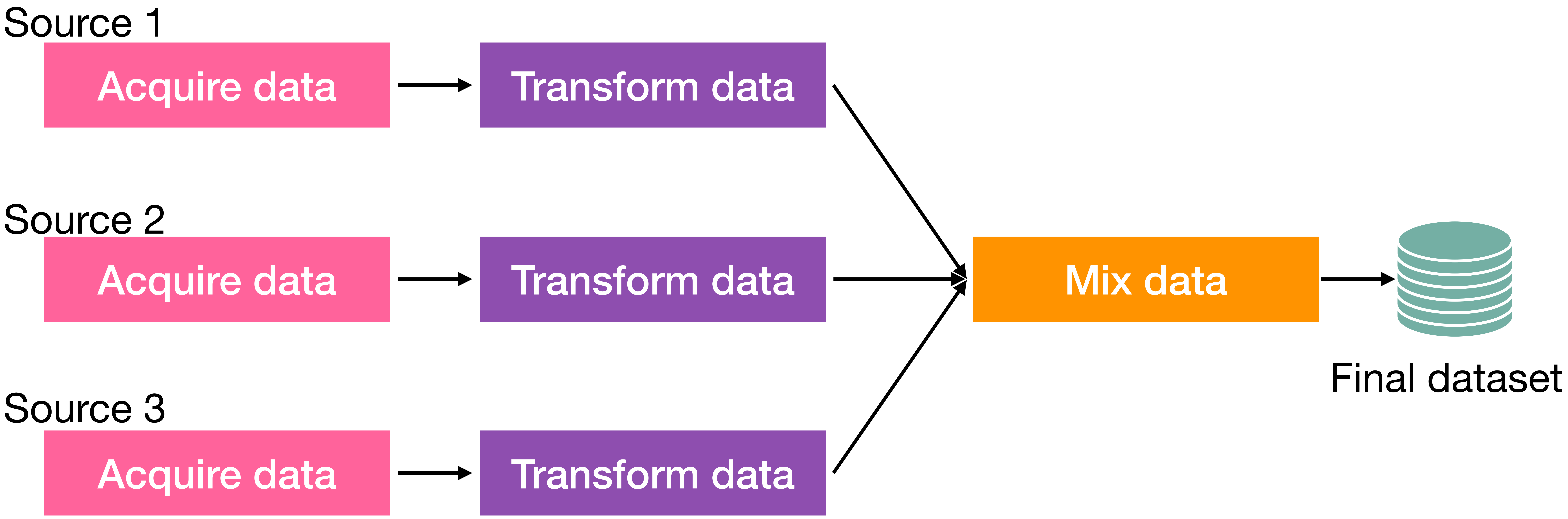
🧪 make scientific discoveries?

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

How to create a good LLM dataset

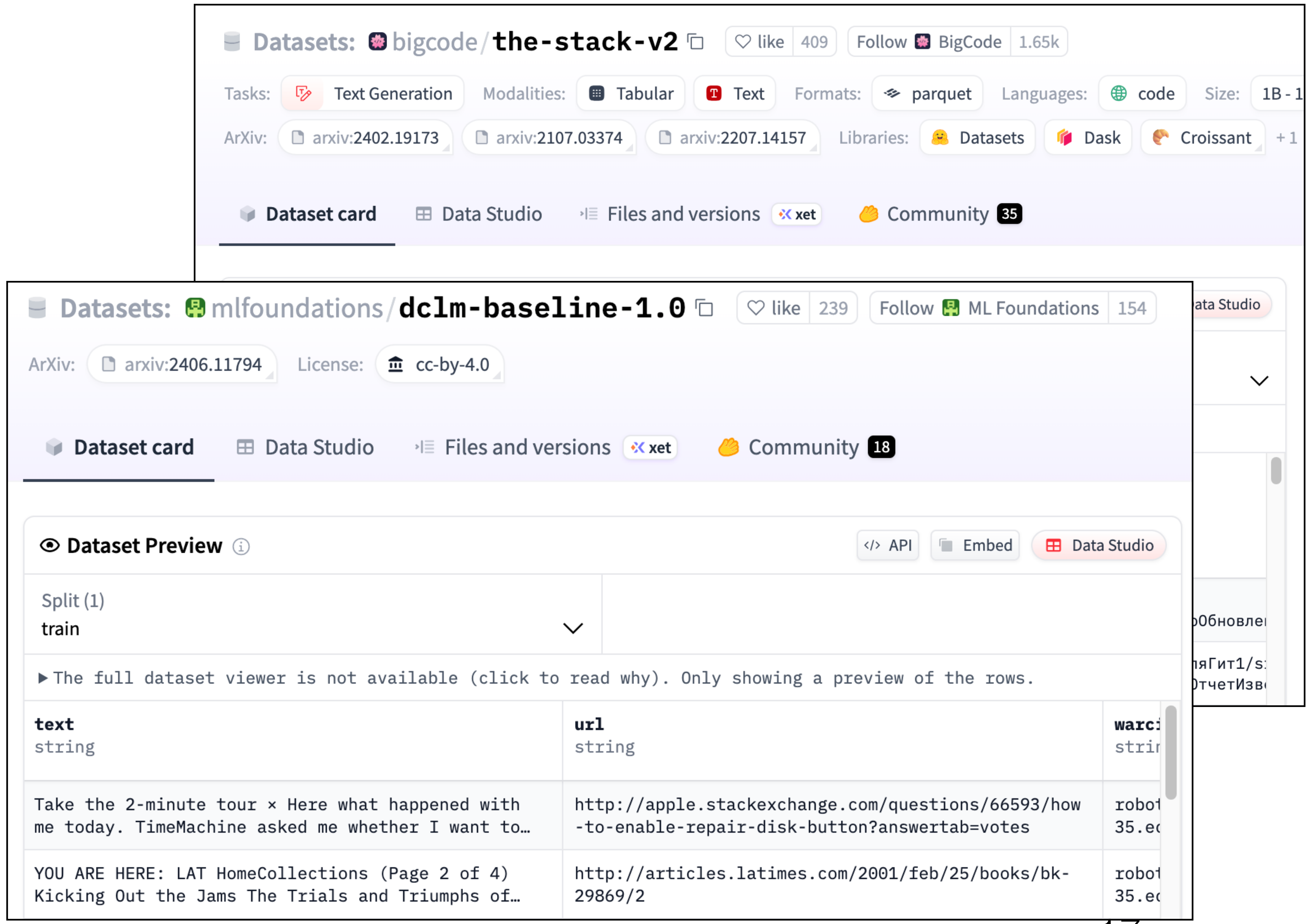
Acquire data	Transform data	Mix data
Quantity ↑	Quality ↑	Composition ↑

How to create a good LLM dataset



Data acquisition: $\varnothing \rightarrow X$

From the web


Datasets: bigcode/the-stack-v2 like 409 Follow BigCode 1.65k

Tasks: Text Generation Modalities: Tabular Text Formats: parquet Languages: code Size: 1B - 1

ArXiv: arxiv:2402.19173 arxiv:2107.03374 arxiv:2207.14157 Libraries: Datasets Dask Croissant +1

Dataset card Data Studio Files and versions xet Community 35

Datasets: mlfoundations/dclm-baseline-1.0 like 239 Follow ML Foundations 154

ArXiv: arxiv:2406.11794 License: cc-by-4.0

Dataset card Data Studio Files and versions xet Community 18

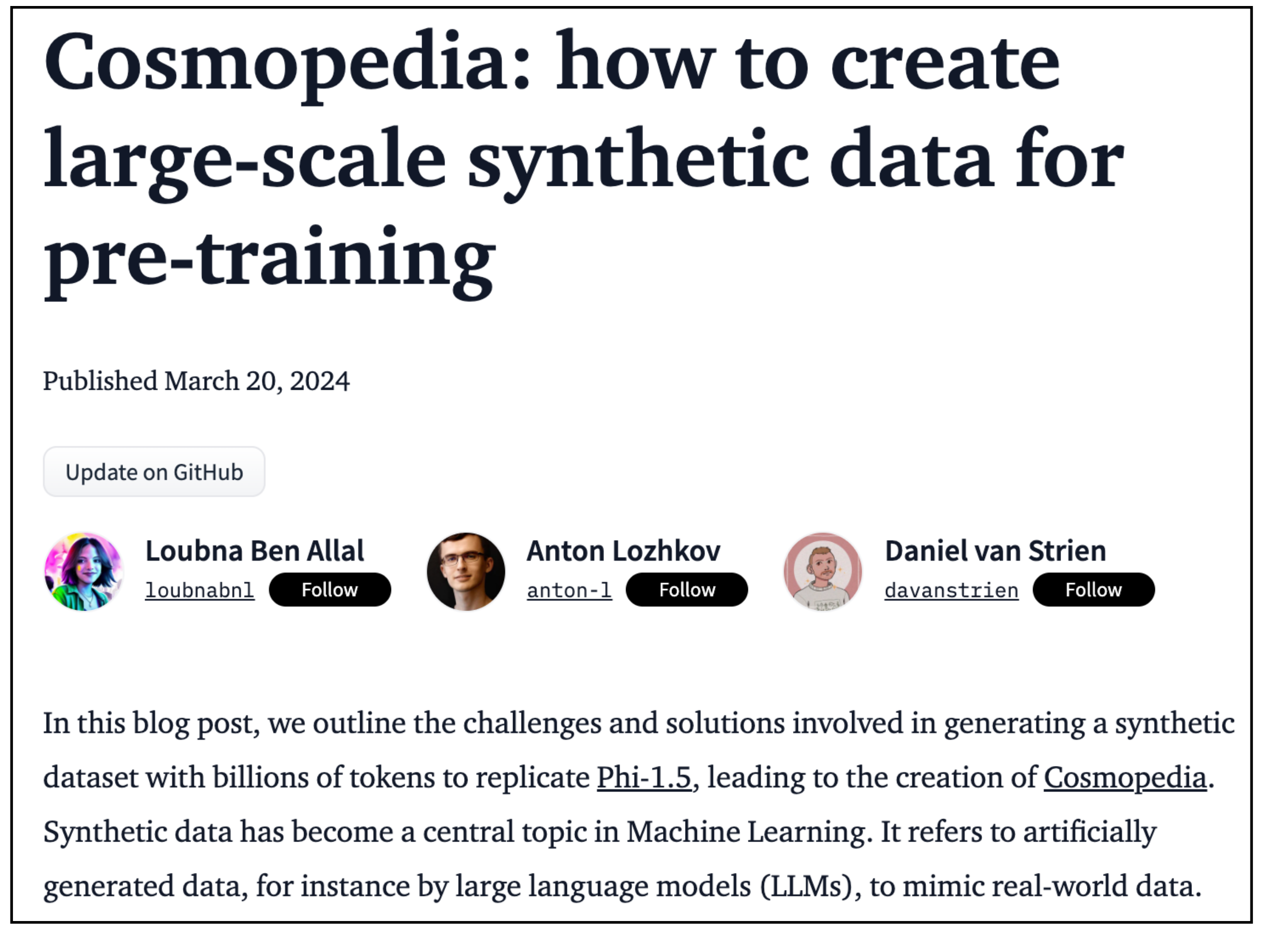
Dataset Preview API Embed Data Studio

Split (1)
train

▸ The full dataset viewer is not available (click to read why). Only showing a preview of the rows.

text string	url string
Take the 2-minute tour × Here what happened with me today. TimeMachine asked me whether I want to…	http://apple.stackexchange.com/questions/66593/how-to-enable-repair-disk-button?answertab=votes
YOU ARE HERE: LAT HomeCollections (Page 2 of 4) Kicking Out the Jams The Trials and Triumphs of…	http://articles.latimes.com/2001/feb/25/books/bk-29869/2

Synthetically Generated


Cosmopedia: how to create large-scale synthetic data for pre-training

Published March 20, 2024

Update on GitHub

Loubna Ben Allal loubnabnl Follow · Anton Lozhkov anton-l Follow · Daniel van Strien davanstrien Follow

In this blog post, we outline the challenges and solutions involved in generating a synthetic dataset with billions of tokens to replicate Phi-1.5, leading to the creation of Cosmopedia. Synthetic data has become a central topic in Machine Learning. It refers to artificially generated data, for instance by large language models (LLMs), to mimic real-world data.

Phi-4

Data transformation: $X \to X'$

Filtering

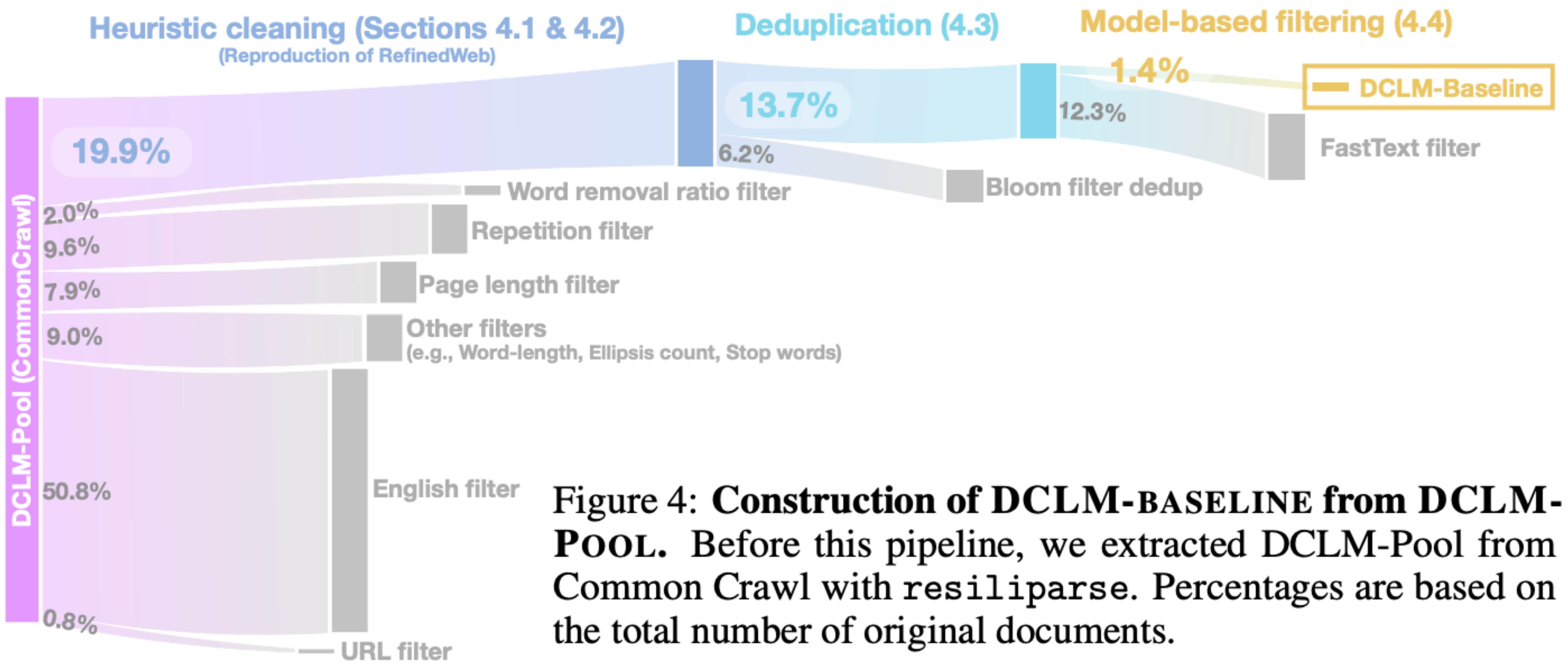


Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

Li et al., 2024. DataComp-LM: In search of the next generation of training sets for language models.
Maini et al., 2024. Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling.

Data transformation: $X \rightarrow X'$

Filtering

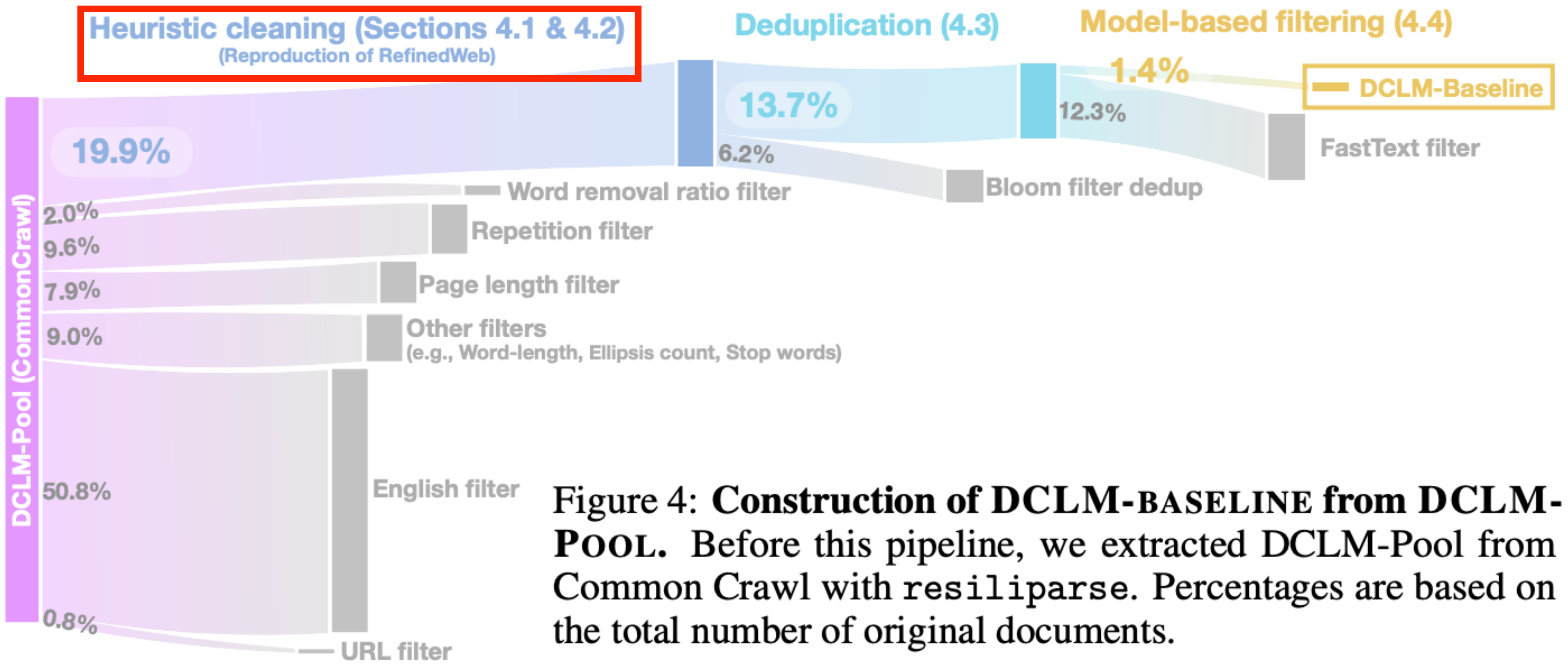


Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

Li et al., 2024. DataComp-LM: In search of the next generation of training sets for language models.
Maini et al., 2024. Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling.

Data transformation: $X \rightarrow X'$

Filtering

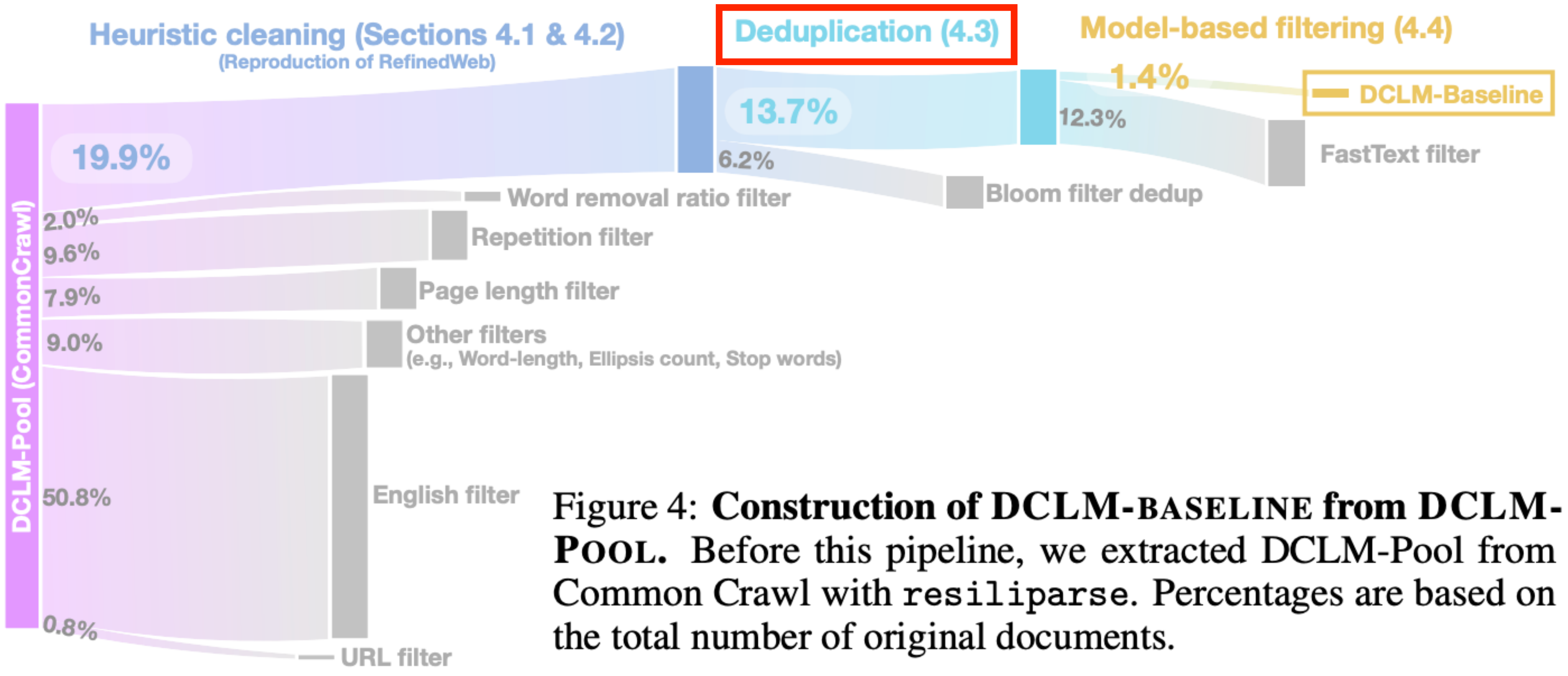


Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

Li et al., 2024. DataComp-LM: In search of the next generation of training sets for language models.
Maini et al., 2024. Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling.

Data transformation: $X \to X'$

Filtering

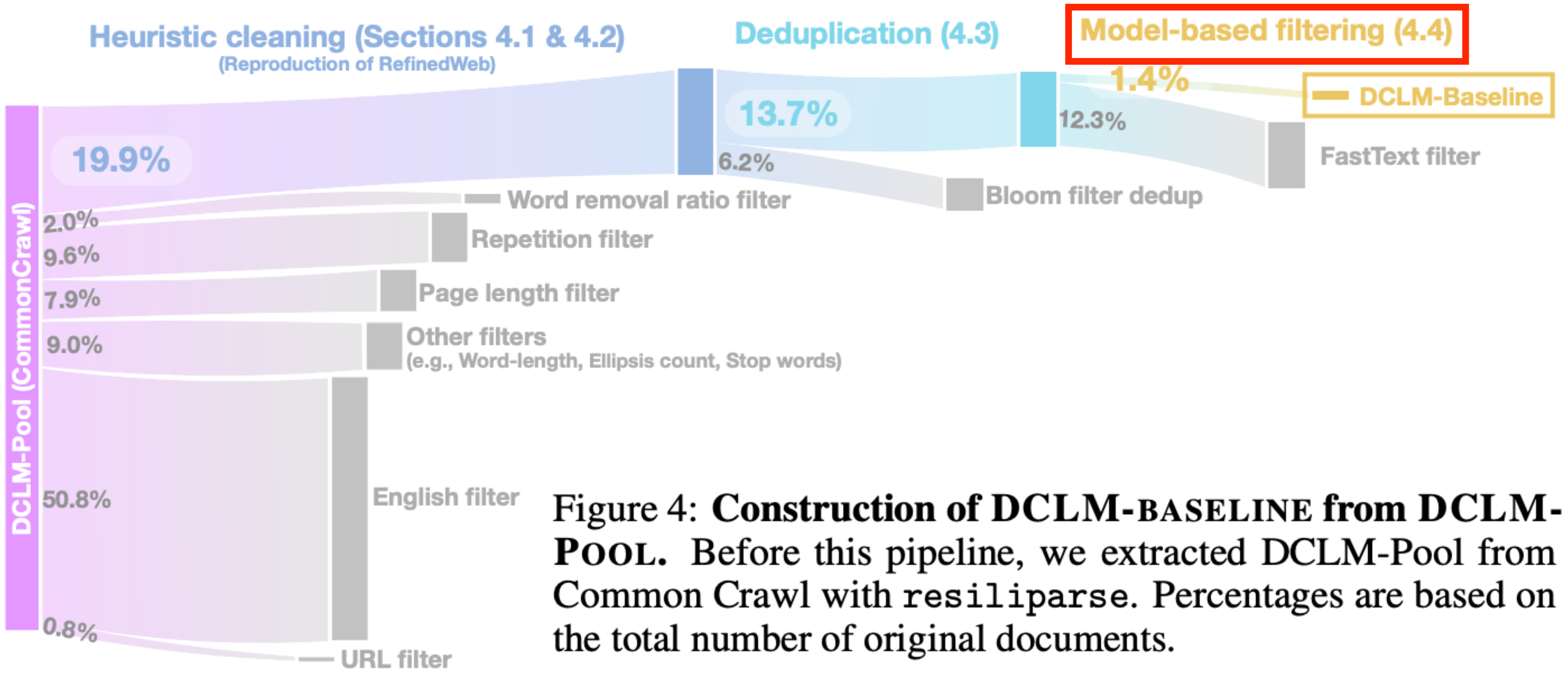


Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

Li et al., 2024. DataComp-LM: In search of the next generation of training sets for language models.
Maini et al., 2024. Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling.

Data transformation: $X \rightarrow X'$

Filtering

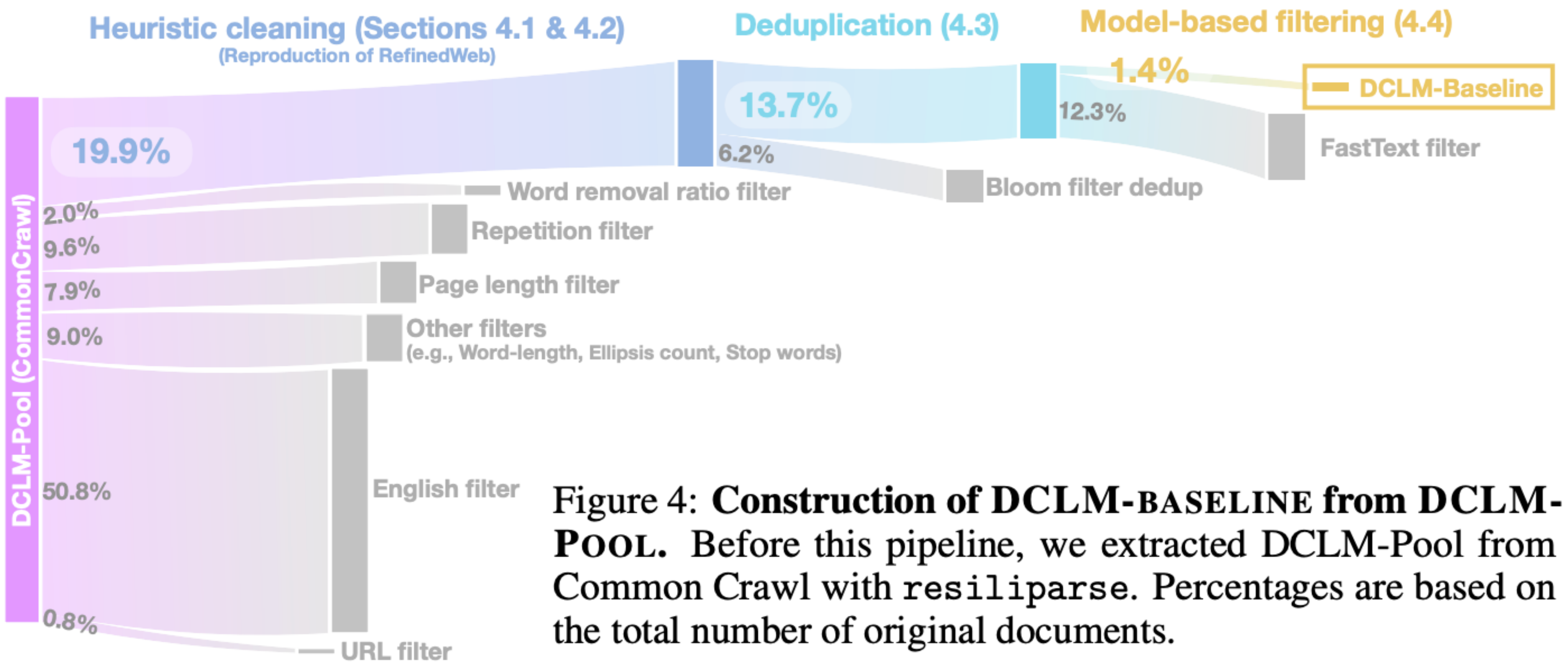


Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

Rewriting

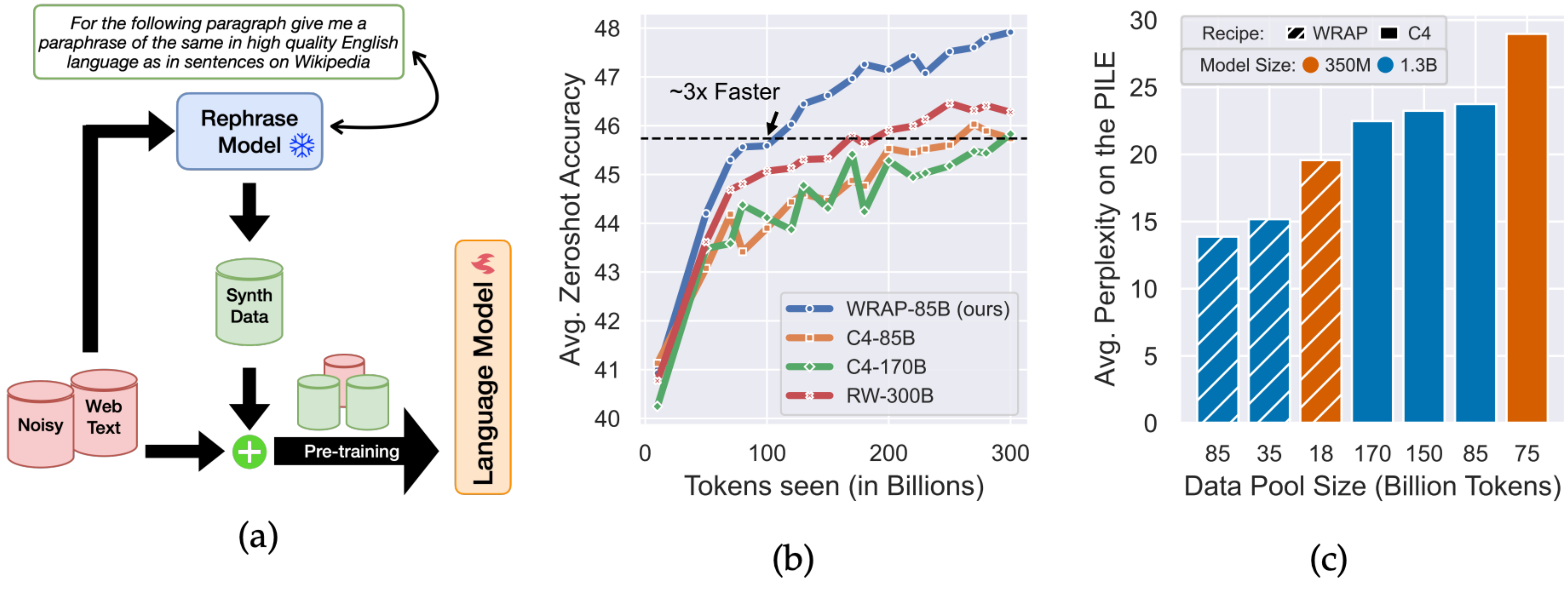


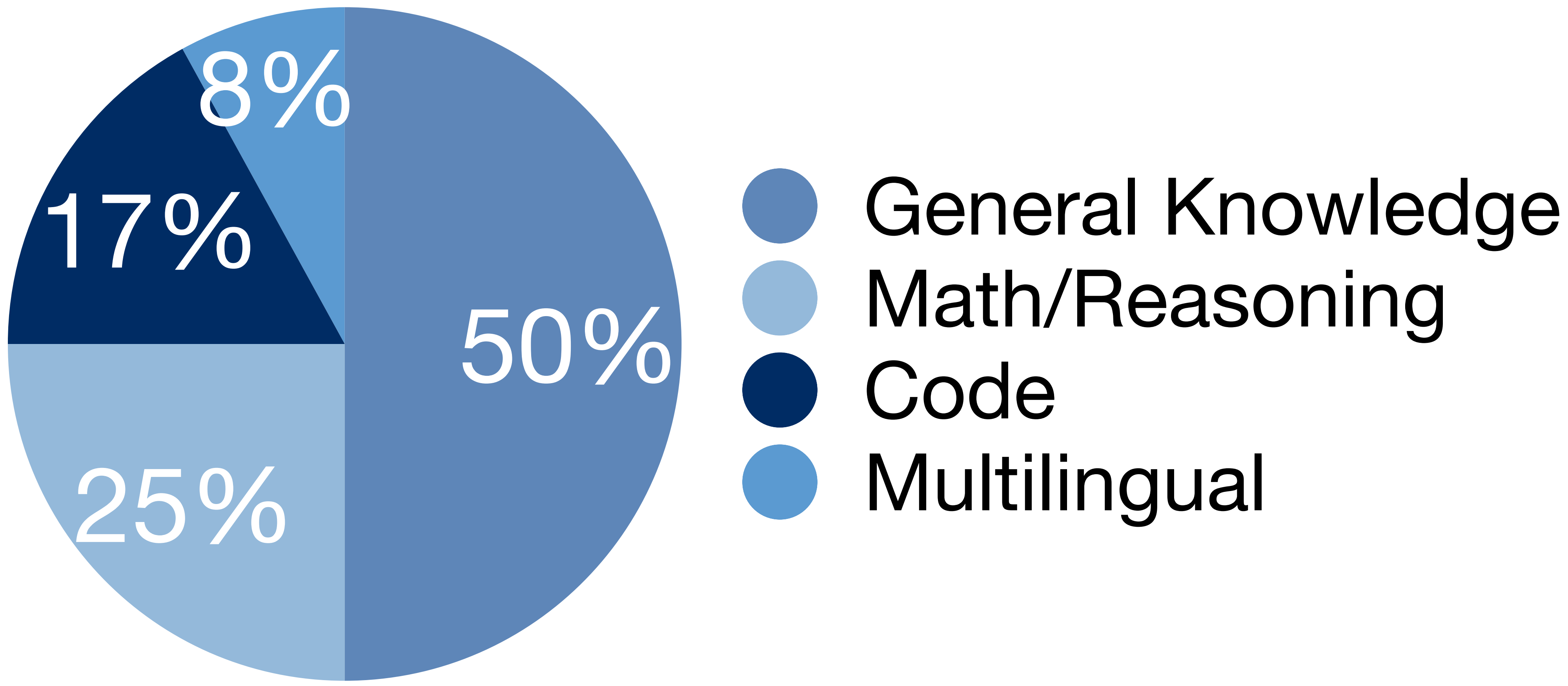
Figure 1: (a) **WRAP** Recipe: We prompt an off-the-shelf instruction-tuned model to rephrase articles on the web, and pre-train an LLM on a mixture of real and synthetic data. (b) Zero-shot performance of GPT 1.3B models trained on combinations of C4 and synthetic variations. Each step corresponds to a batch of 1M samples. (c) Weighted average perplexity over 21 sub-domains of the Pile for varying model sizes and amount of pre-training data.

Li et al., 2024. DataComp-LM: In search of the next generation of training sets for language models.
Maini et al., 2024. Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling.

Data mixing: $X_1, \ldots, X_m \rightarrow X_{final}$

Goal: given m data domains, how should we combine the domains to produce a good model?

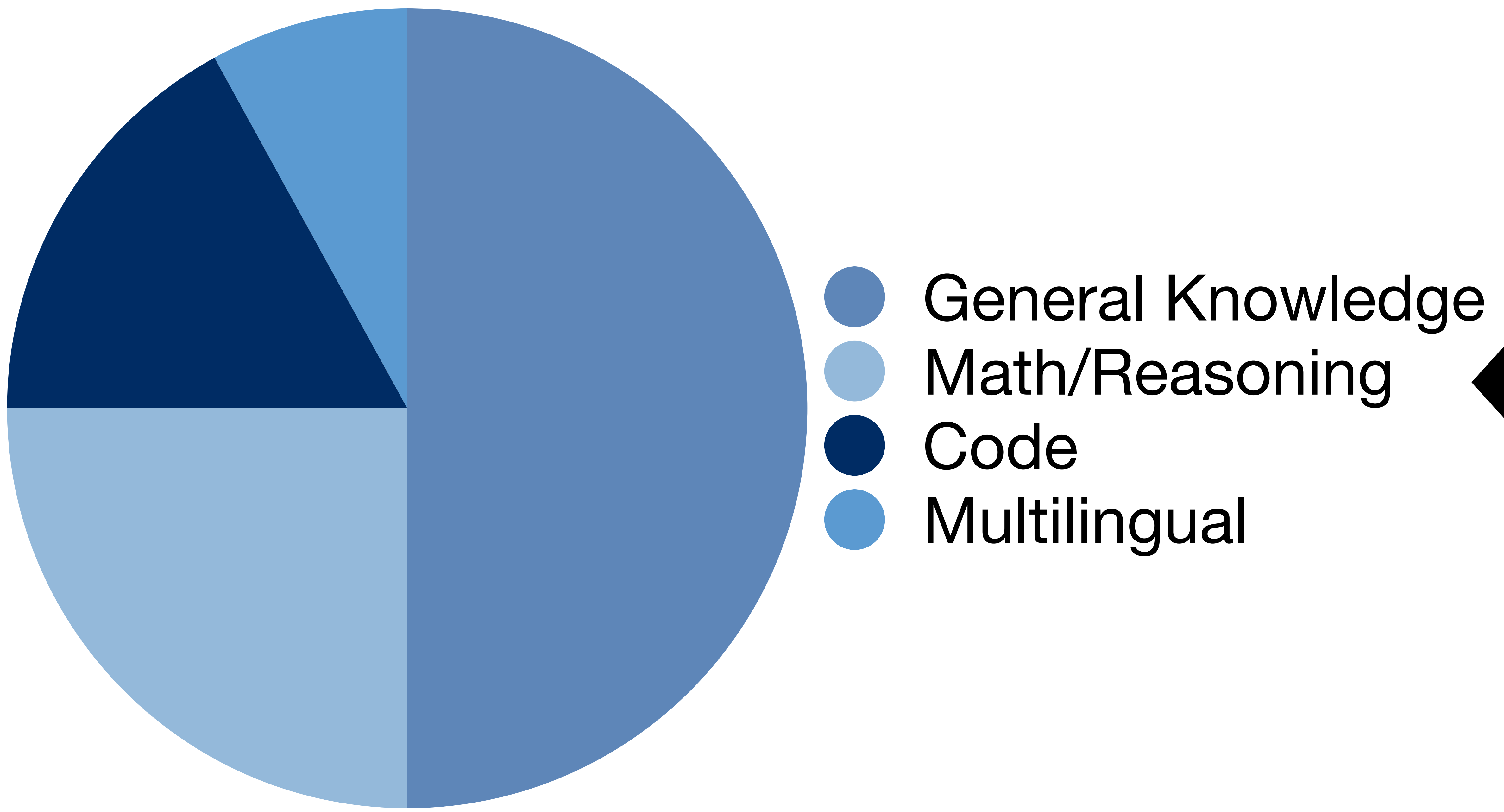
To be discussed in the next part of the tutorial!



Llama-3 mix

Deep dive: Data Mixing

What is Mixing?



LLM that can do many things:

📝 summarise documents

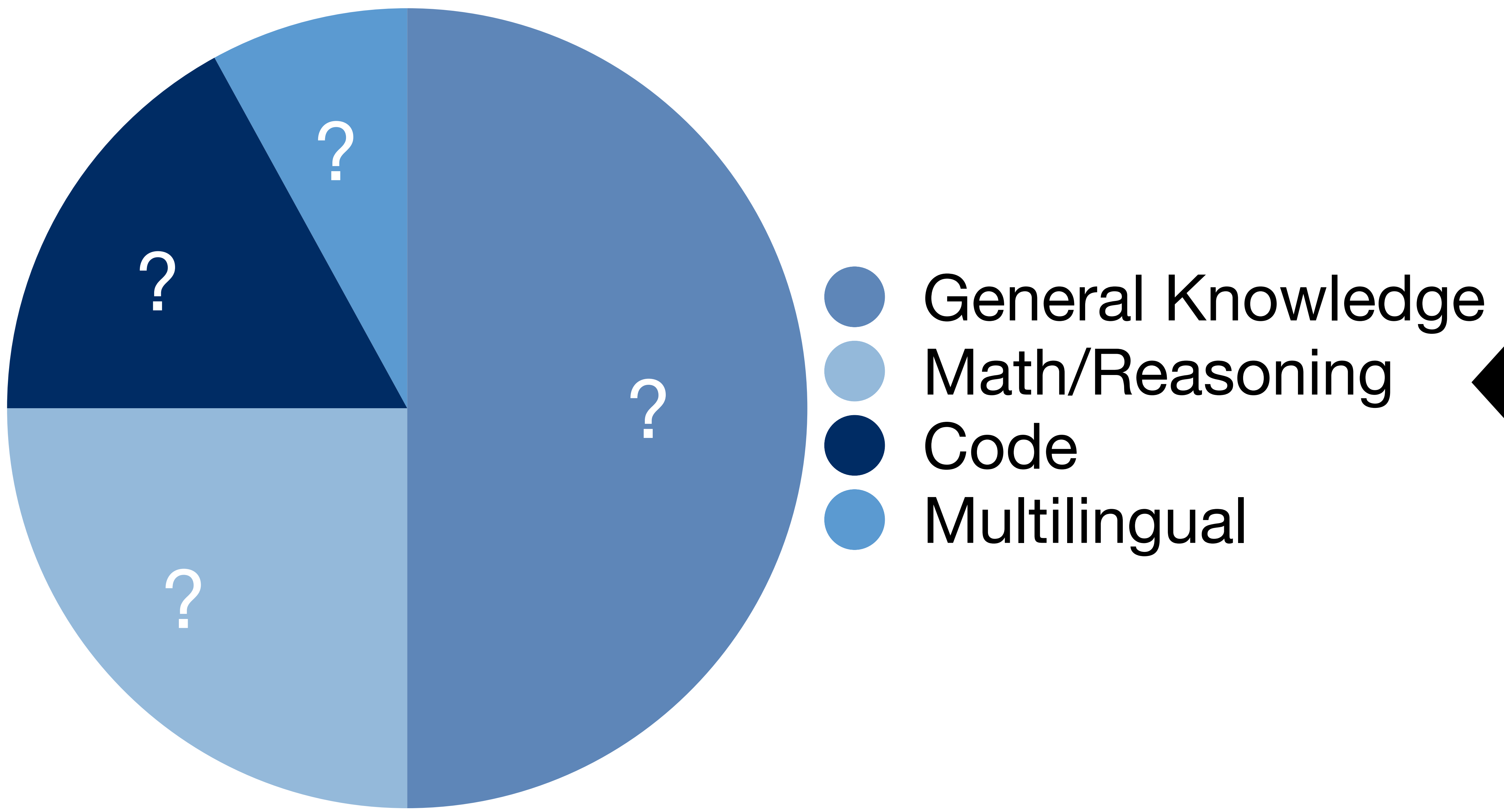
🧑‍💻 write code

🔢 solve math problems

💬 chat with users in many languages

🧪 make scientific discoveries?

What is Mixing?



LLM that can do many things:

📝 summarise documents

🧑‍💻 write code

🔢 solve math problems

💬 chat with users in many languages

🧪 make scientific discoveries?

Goal: given m domains, in what ratios p should we sample the domains to produce a model that excels at all desired capabilities?”

Why mix?

Reality:

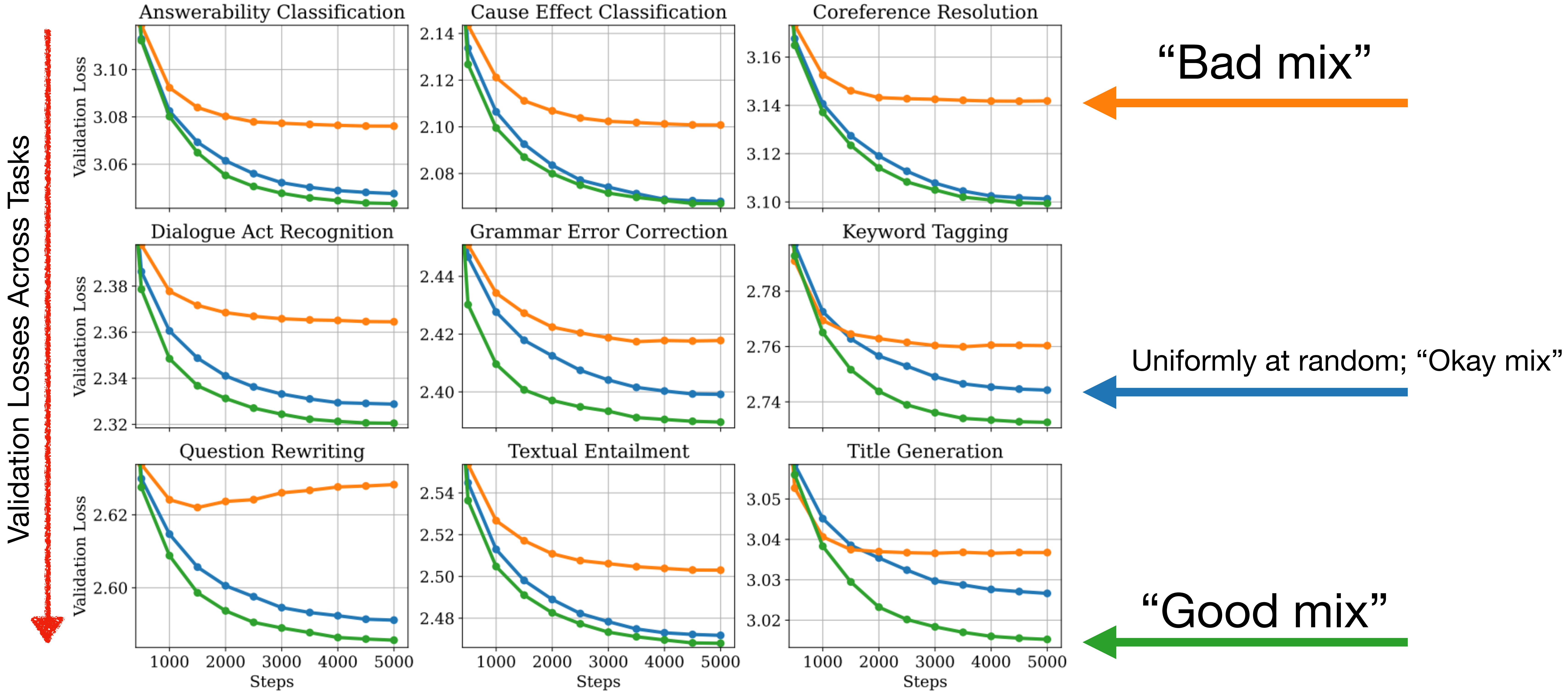
- Models are trained on multiple datasets.
- Mixing is inevitable: even simple concatenation of datasets is a form of mixing.

Mixing lets you:

- Control the training distribution with a low-dimensional knob, p.
- Navigate trade-offs among desired model capabilities

Why mix?

Mixing works! A "good mix" can dramatically improve performance across tasks.

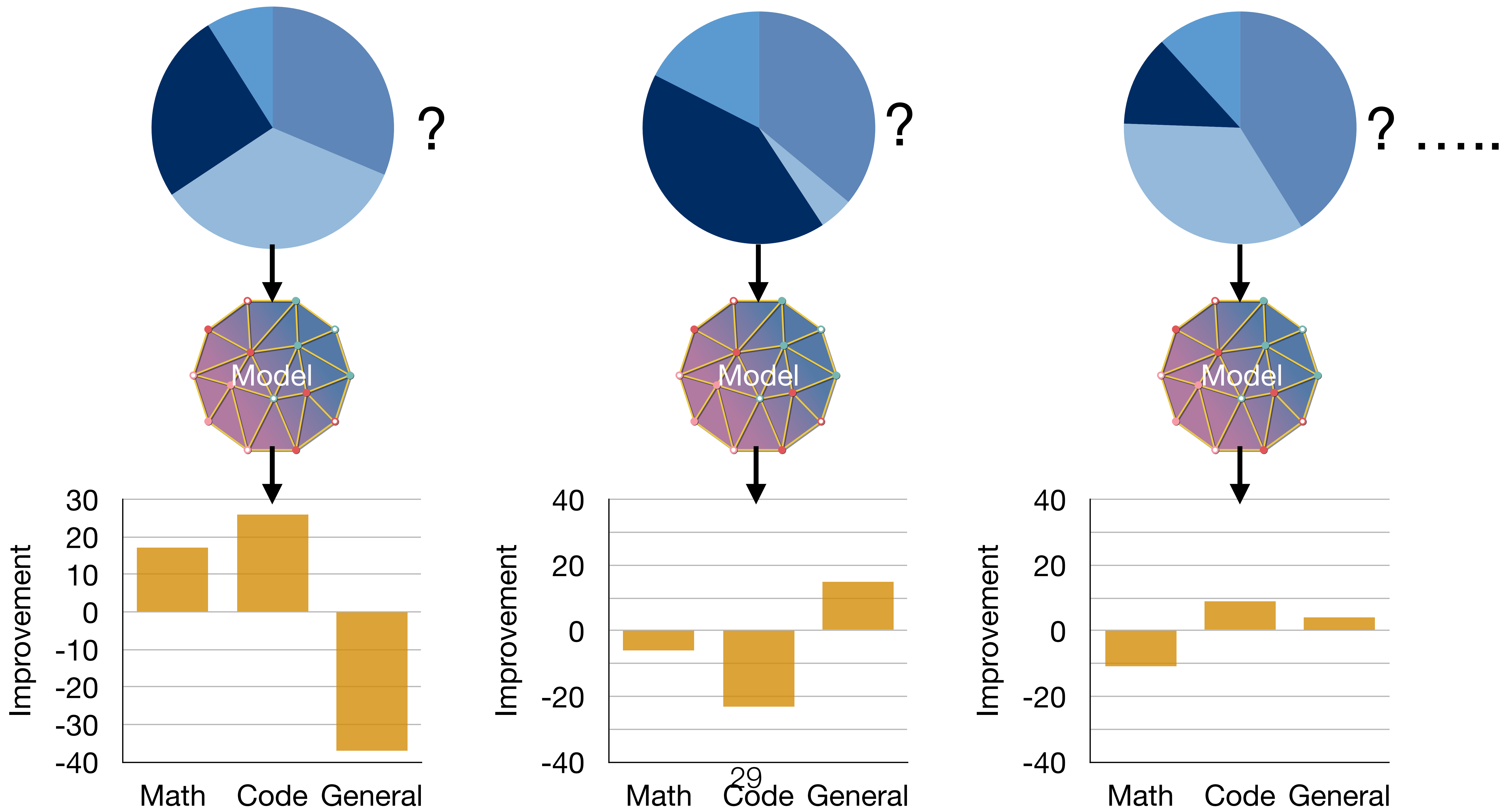


 Chen et al., 2023. Skill-it! A Data-Driven Skills Framework for Understanding and Training Language Models

Why is mixing challenging?

Naive approach: brute-force search/manual tuning to find a good mix = costly!

- Used in GLAM (2021), Tulu3 (2024), OpenVLA (2024)



Mixing settings

Static mixing

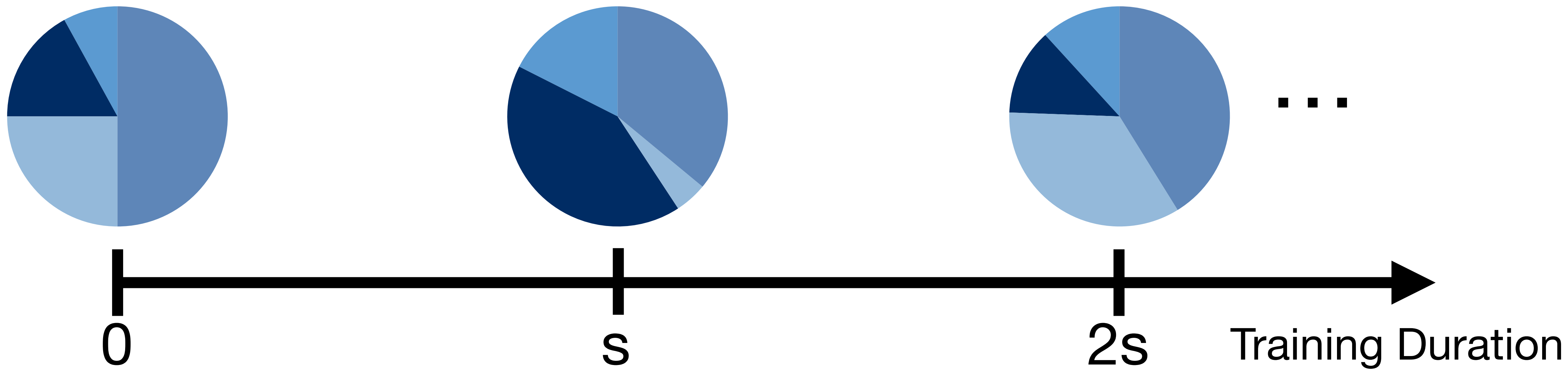
Train entire
model on p

Dynamic mixing

Train model on
p^1 for s steps

Train model on
p^2 for s steps

Train model on
p^3 for s steps



Mixing settings

Static mixing

Train entire
model on p

✅ Simple; prepare mix & hit “run”

✅ Reusable (e.g., “OXE Magic Soup”)

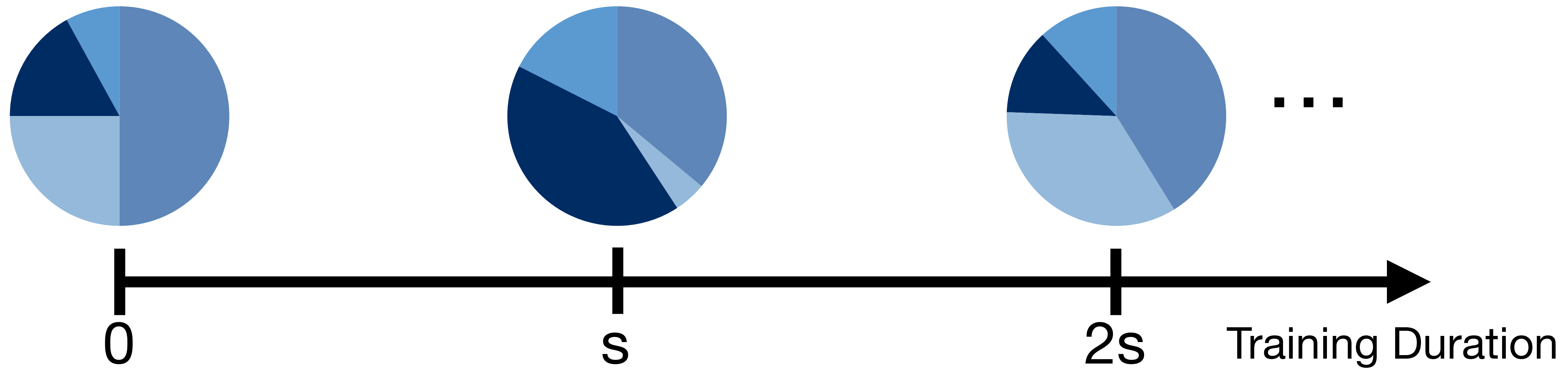
❌ Can leave performance on the table

Dynamic mixing

Train model on
p^1 for s steps

Train model on
p^2 for s steps

Train model on
p^3 for s steps

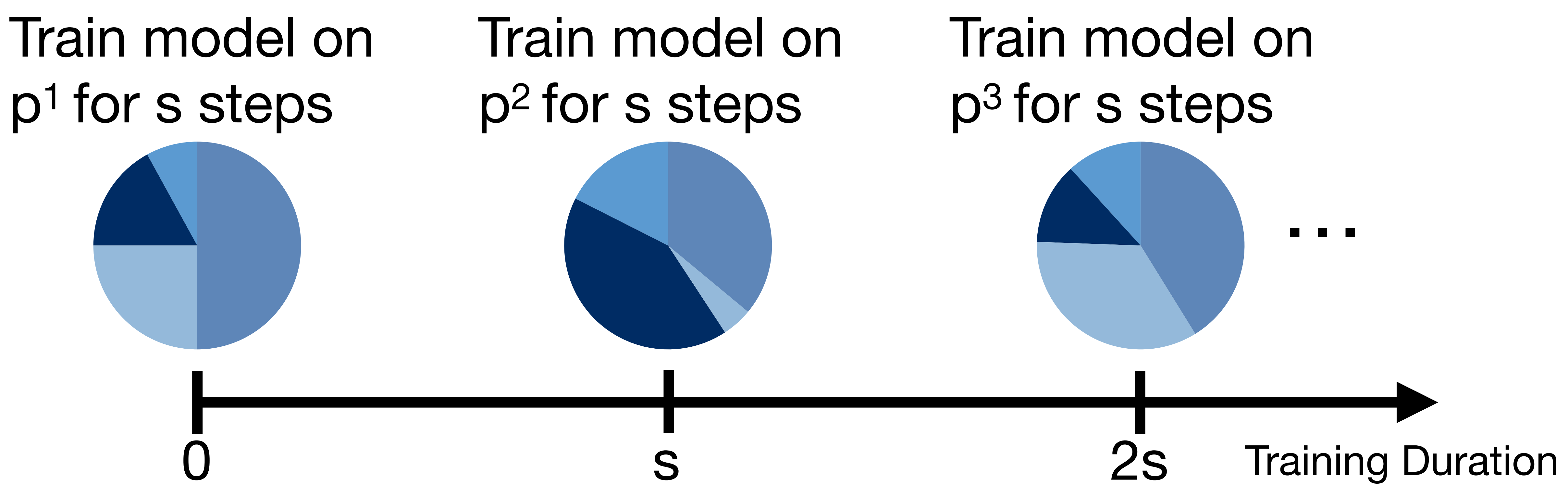


Mixing settings

Static mixing

- ✅ Simple; prepare mix & hit “run”
- ✅ Reusable (e.g., “OXE Magic Soup”)
- ❌ Can leave performance on the table

Dynamic mixing



- ✅ Adapts mix to current model checkpoint
- ✅ Strong evidence that order matters (example: learning 1 digit addition before 2 digit addition)
- ❌ Implementation issues (incompatible with many trainers)
- ❌ Difficult to reuse a dynamic mix

Formal problem (static)

- **Given:** m training domains $D_1, \ldots, D_m$, token budget N

Formal problem (static)

- **Given:** m training domains $D_1, \ldots, D_m$, token budget N
- **Choose:** data mix $p \in \triangle^{m-1}$, then create D_{train} using $N \times p_i$ tokens per domain D_i

Formal problem (static)

- **Given:** m training domains $D_1, \ldots, D_m$, token budget N
- **Choose:** data mix $p \in \triangle^{m-1}$, then create D_{train} using $N \times p_i$ tokens per domain D_i
- **Evaluate:** Train $LM(p)$, compute validation loss $f_i(LM(p))$ for n val datasets
  - Val datasets: held-out split on training domains (n=m), or OOD/downstream

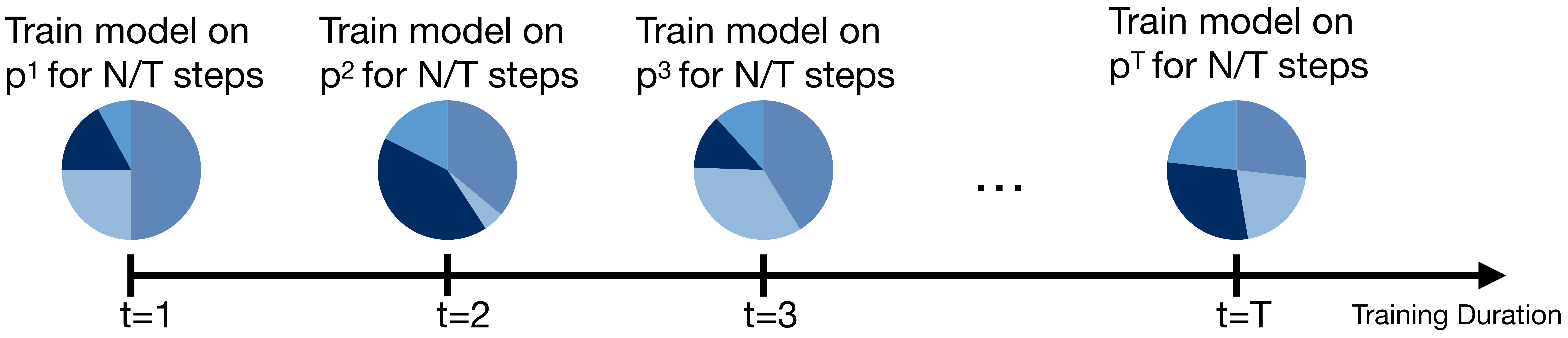
Formal problem (static)

- **Given:** m training domains $D_1, \ldots, D_m$, token budget N
- **Choose:** data mix $p \in \triangle^{m-1}$, then create D_{train} using $N \times p_i$ tokens per domain D_i
- **Evaluate:** Train $LM(p)$, compute validation loss $f_i(LM(p))$ for n val datasets
  - Val datasets: held-out split on training domains (n=m), or OOD/downstream
- **Static Data Mixing Problem:**

$$\text{minimize}_{p \in \triangle^{m-1}} \frac{1}{n} \sum_{i=1}^{n} f_i(LM(p))$$

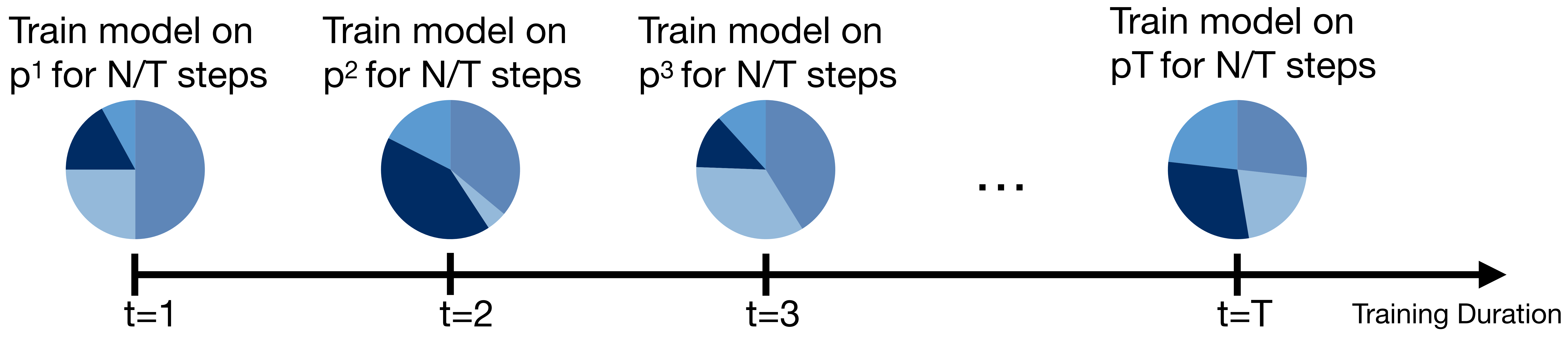
Goal: *compute near-optimal p* in a way that's more efficient than search*

Formal problem (dynamic)



- **Choose:** Split training into *T* stages according to the dynamic mix $p = [p^1, p^2, \ldots, p^T]$ (where each $p^t \in \triangle^{m-1}$)

Formal problem (dynamic)



- **Choose:** Split training into *T stages* according to the dynamic mix $p = [p^1, p^2, \ldots, p^T]$ (where each $p^t \in \triangle^{m-1}$)

- **Dynamic Data Mixing Problem:**

$$\text{minimize}_{\boxed{p \in \triangle^{(m-1)\times T}}} \frac{1}{n} \sum_{i=1}^{n} f_i(LM(p))$$

Goal: *compute near-optimal p* in a way that's more efficient than search*

Many methods…

Efficient Online Data Mixing For Language Model Pre-Training

Alon Albalak[1] Liangming Pan[1] Colin Raffel[2,3] William Yang Wang[1]
[1]University of California, Santa Barbara
[2]University of Toronto
[3]Vector Institute

DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining

Sang Michael Xie*[1,2], Hieu Pham[1], Xuanyi Dong[1], Nan Du[1], Hanxiao Liu[1], Yifeng Lu[1], Percy Liang[2], Quoc V. Le[1], Tengyu Ma[2], and Adams Wei Yu[1]
[1]Google DeepMind
[2]Stanford University

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

Jiasheng Ye[1,*] Peiju Liu[1,*] Tianxiang Sun[1] Yunhua Zhou[2] Jun Zhan[1] Xipeng Qiu[1,†]

OPTIMIZING PRETRAINING DATA MIXTURES WITH LLM-ESTIMATED UTILITY

William Held*[σ,γ] Bhargavi Paranjape[μ] Punit Singh Koura[μ]
Mike Lewis[μ] Frank Zhang[μ] Todor Mihaylov[μ]
[μ]Meta AI [σ]Stanford University [γ]Georgia Institute of Technology

PiKE: Adaptive Data Mixing for Large-Scale Multi-Task Learning Under Low Gradient Conflicts

Zeman Li[1,2*] Yuan Deng[2] Peilin Zhong[2] Meisam Razaviyayn[1,2] Vahab Mirrokni[2]
[1]University of Southern California [2]Google Research
{zemanli,razaviya}@usc.edu
{dengyuan,peilinz,mirrokni}@google.com

REGMIX: Data Mixture as Regression for Language Model Pre-training

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

ADAPTIVE DATA OPTIMIZATION: DYNAMIC SAMPLE SELECTION WITH SCALING LAWS

Yiding Jiang[†*] Allan Zhou[‡*] Zhili Feng[†] Sadhika Malladi[§] J. Zico Kolter[†]
Carnegie Mellon University[†] Stanford University[‡] Princeton University[§]
yidingji@cs.cmu.edu, ayz@cs.stanford.edu

Skill-it! A Data-Driven Skills Framework for Understanding and Training Language Models

Mayee F. Chen*[1] Nicholas Roberts[2] Kush Bhatia[1] Jue Wang[3] Ce Zhang[3,4]
Frederic Sala[2] Christopher Ré[1]

NVIDIA 2025-4-18

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

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

DOGE: Domain Reweighting with Generalization Estimation

Simin Fan[1] Matteo Pagliardini[1] Martin Jaggi[1]

- Numerous techniques: bandits, distributionally robust optimization, multi-task learning, portfolio optimization, …
- Which one to use? What are they really doing?

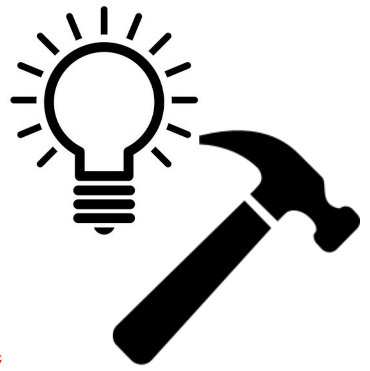
Key insight: mixing laws

Key insight: mixing laws

There is a structured relationship between the data mix *p* and the performance metrics $f_i(LM(p))$.

Key insight: mixing laws

There is a structured relationship between the data mix *p* and the performance metrics $f_i(LM(p))$.



Takeaway: Aim to ***understand*** the relationship between performance and data, then ***exploit*** this understanding to optimize the data mix!

Key insight: mixing laws

The relationship between the mix p and $f_i(LM(p))$ can be modelled by a **mixing law:**

$$f_i(LM(p)) \approx b_i \sigma(-A_i^\top p) + c_i \quad A_i \in \mathbb{R}^m \;\; b_i, c_i \in \mathbb{R} \;\; \forall i \in [n]$$

Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Key insight: mixing laws

The relationship between the mix p and $f_i(LM(p))$ can be modelled by a **mixing law:**

$$f_i(LM(p)) \approx b_i \sigma(-A_i^\top p) + c_i \quad A_i \in \mathbb{R}^m \;\; b_i, c_i \in \mathbb{R} \;\; \forall i \in [n]$$

Monotonic + linear in mix

Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Key insight: mixing laws

The relationship between the mix *p* and $f_i(LM(p))$ can be modelled by a **mixing law:**

$$f_i(LM(p)) \approx b_i \sigma(-A_i^\top p) + c_i \quad A_i \in \mathbb{R}^m \;\; b_i, c_i \in \mathbb{R} \;\; \forall i \in [n]$$

Monotonic + linear in mix

Interpretation:

- Small/big change in p = small/big change in performance
- Each domain linearly contributes A_{ij}, a “score” for how much domain j impacts validation dataset i

Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Case study: two methods that utilise mixing laws

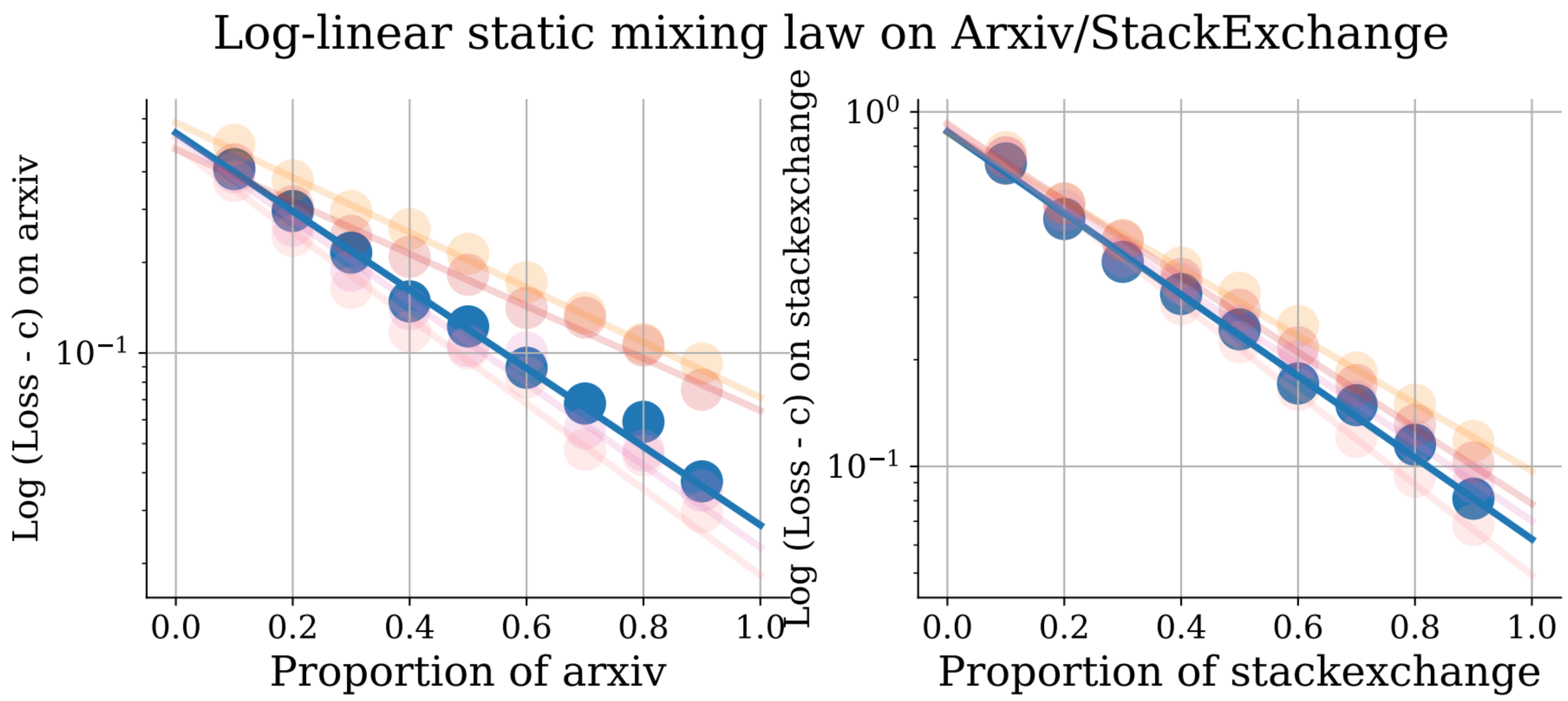
Static setting: Data Mixing Laws (Ye et al., 2024)

$$f_i(LM(p)) \approx \exp(-A_i^\top p) + c_i$$

 Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Static setting: Data Mixing Laws (Ye et al., 2024)

$$f_i(LM(p)) \approx \exp(-A_i^\top p) + c_i$$



R^2 of static mixing law on SlimPajama (7 domains): 0.997

 Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Static Mixing Law: Method

1. Explore

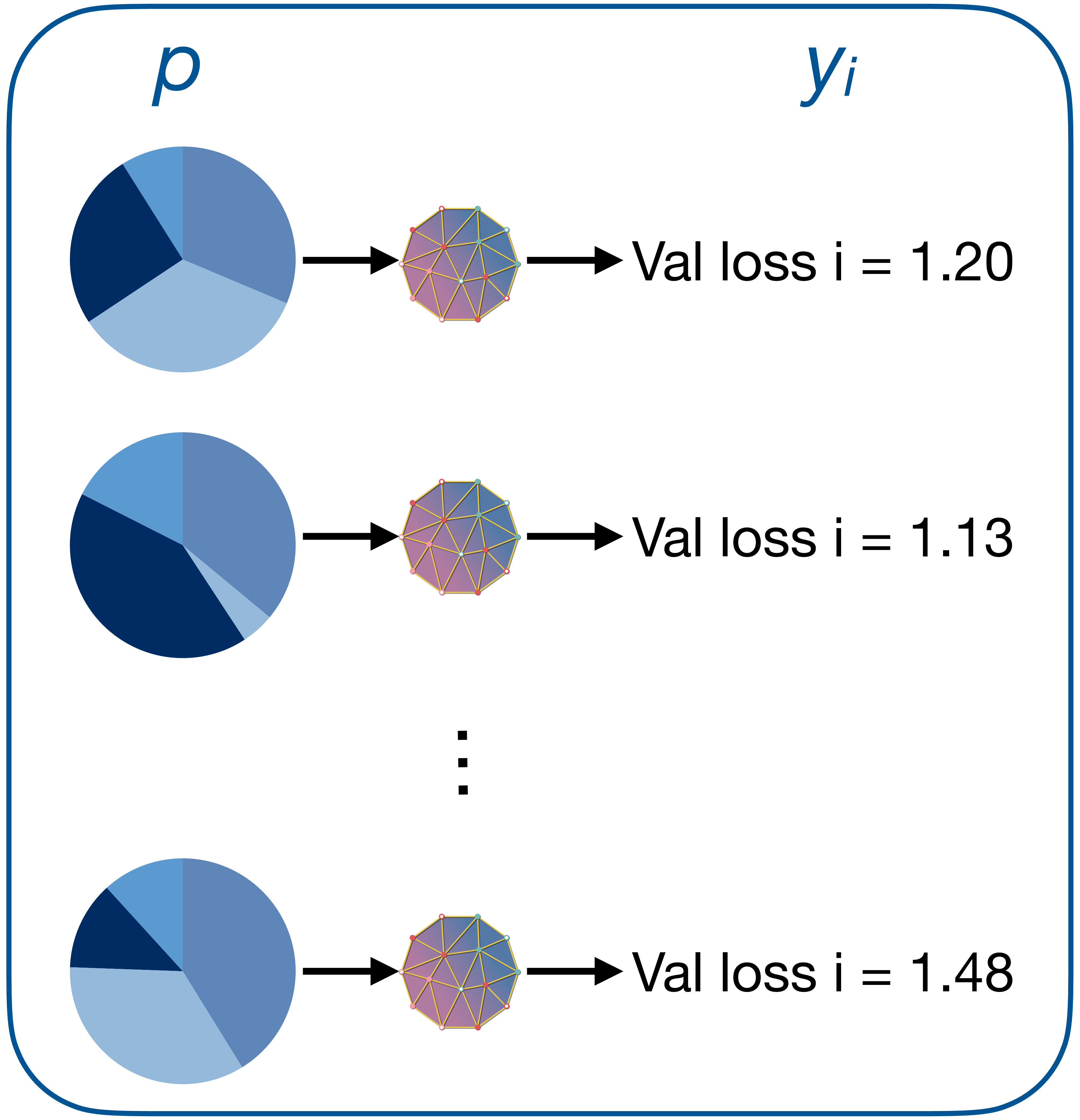
2. Fit

3. Optimize

 Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Static Mixing Law: Method

1. Explore



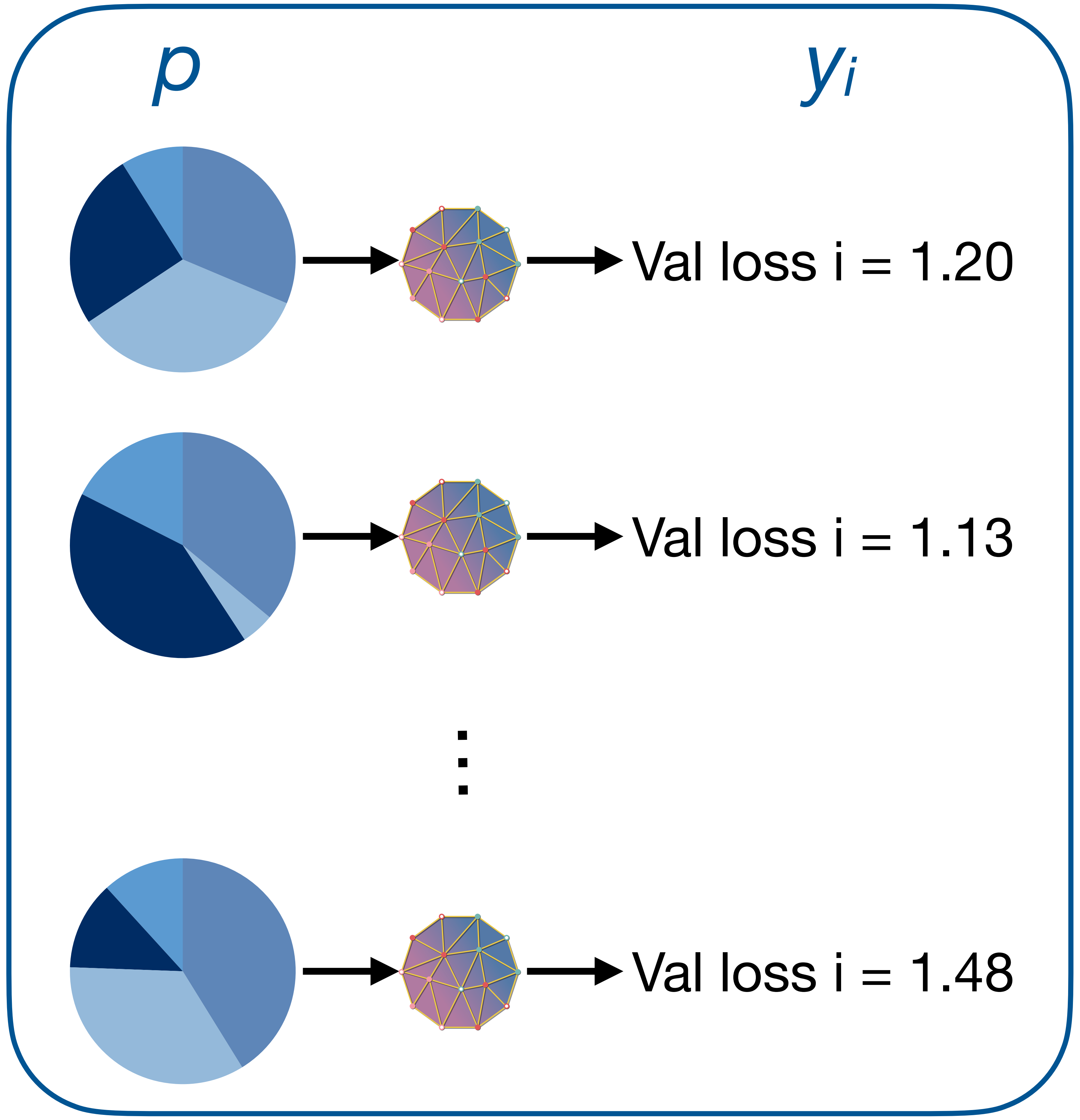
2. Fit

3. Optimize

 Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Static Mixing Law: Method

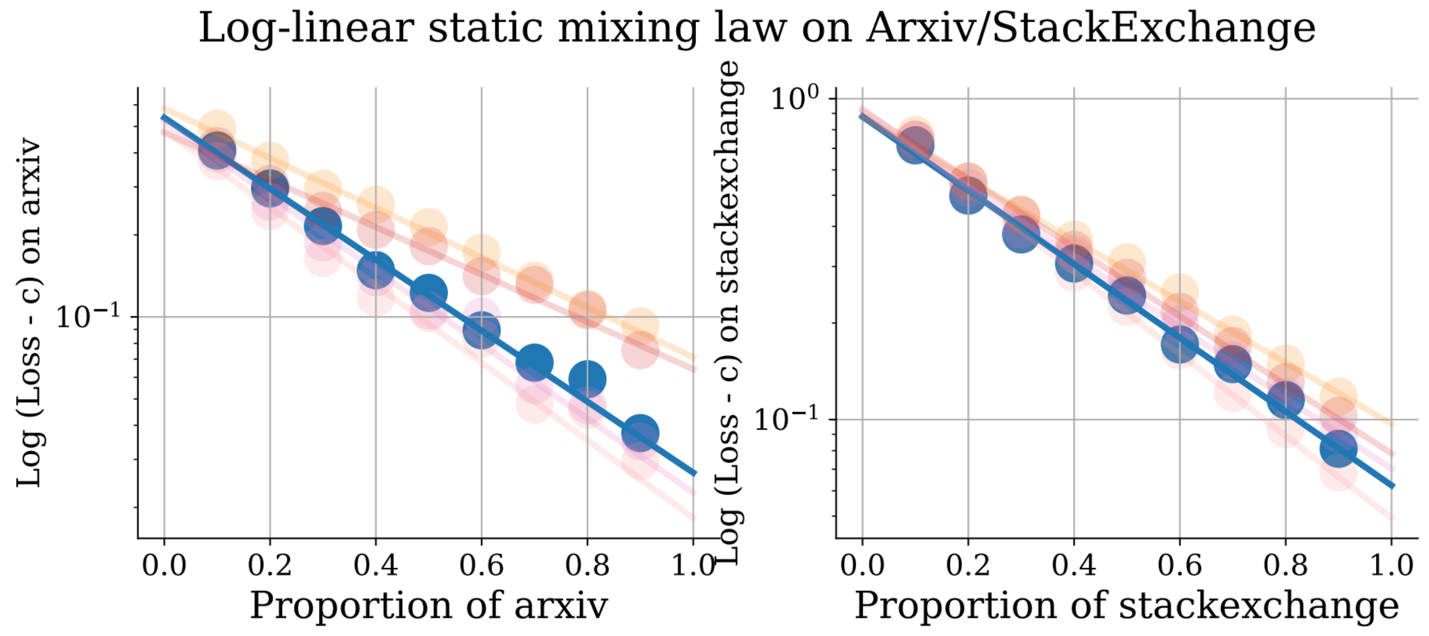
1. Explore



2. Fit

Use (p, y_i) to fit parameters of mixing law

$$\hat{f}_i(LM(p)) := \exp(-\hat{A}_i^\top p) + \hat{c}_i$$

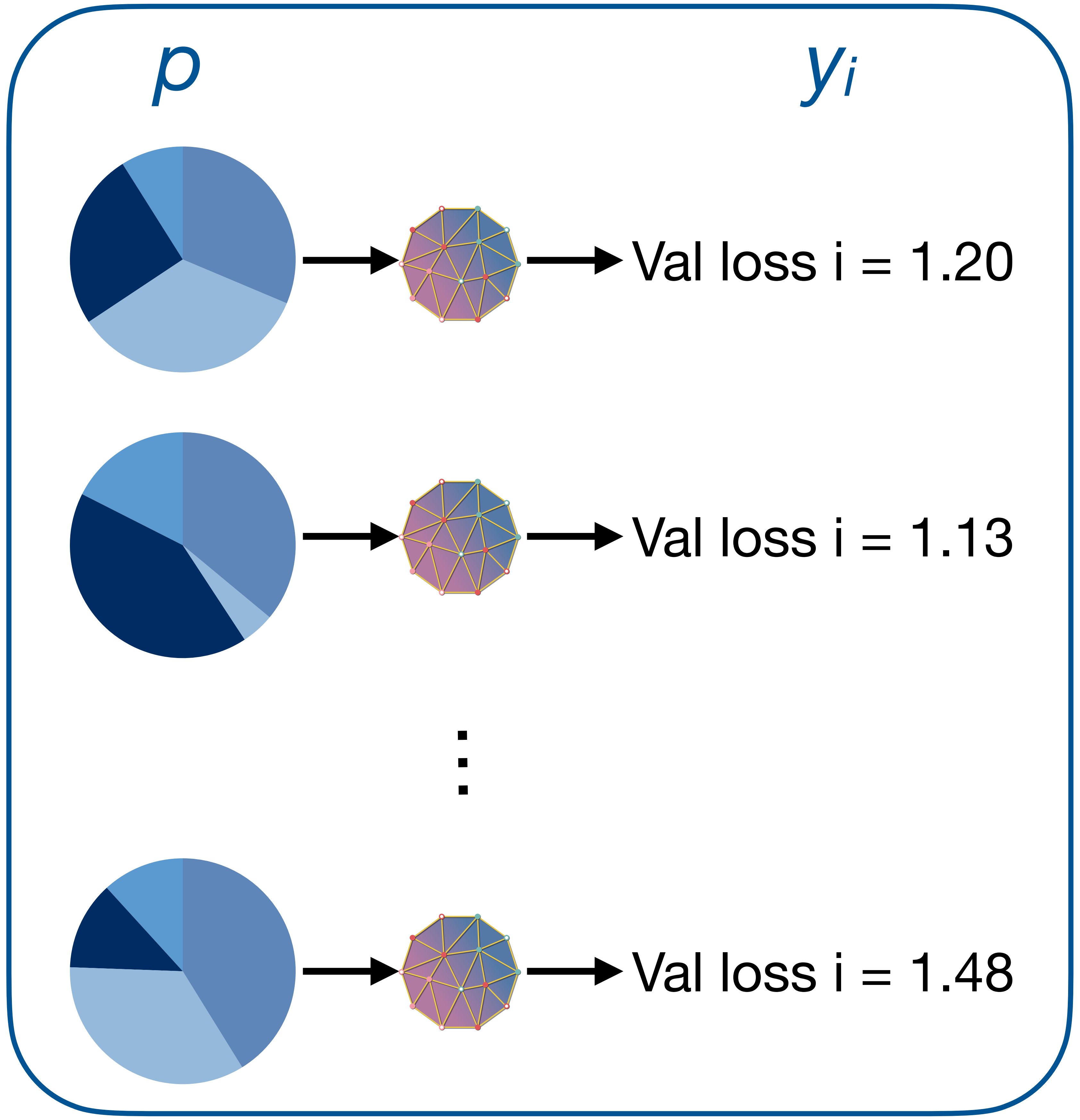


3. Optimize

Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Static Mixing Law: Method

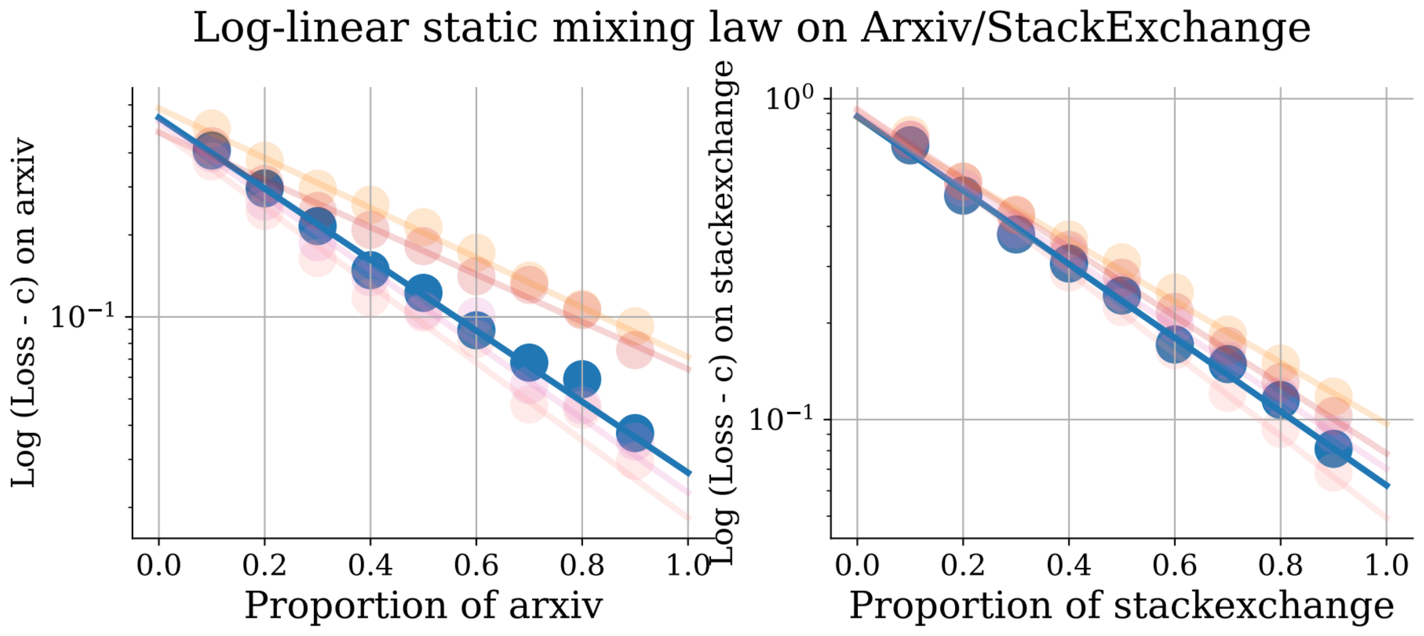
1. Explore



2. Fit

Use (p, y_i) to fit parameters of mixing law

$$\hat{f}_i(LM(p)) := \exp(-\hat{A}_i^\top p) + \hat{c}_i$$

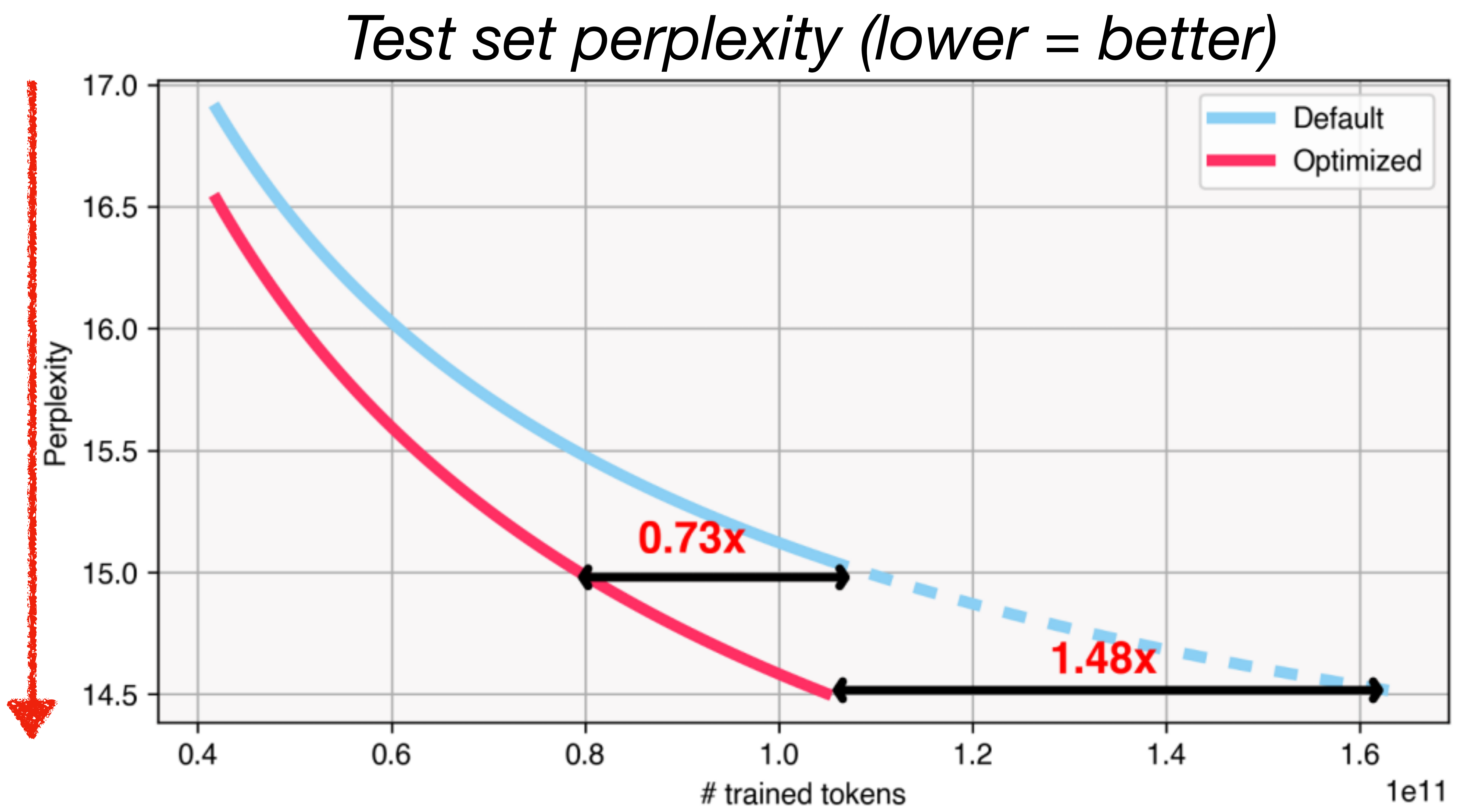


3. Optimize

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

 Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Static Mixing Law: Results



Domains	Default Mixture	Optimized Mixture
CommonCrawl	0.6700	0.1250
C4	0.1500	0.2500
Github	0.0450	0.1406
ArXiv	0.0450	0.2500
Books	0.0450	0.0938
StackExchange	0.0250	0.1250
Wikipedia	0.0200	0.0156

Figure 8: The validation perplexity on the Pile validation set for 1B models trained on the default mixture and the optimized mixture of RedPajama for 100B tokens. Our optimized mixture achieves the performance of the default mixture only using 0.73 of the original number of training steps and eventually achieves a performance comparable to a default mixture trained with 1.48 times more tokens (estimated by the scaling law of training steps, shown as the dashed line). The specific mixture proportions are in the right table.

 Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

Dynamic setting: Aioli (Chen et al., 2024)

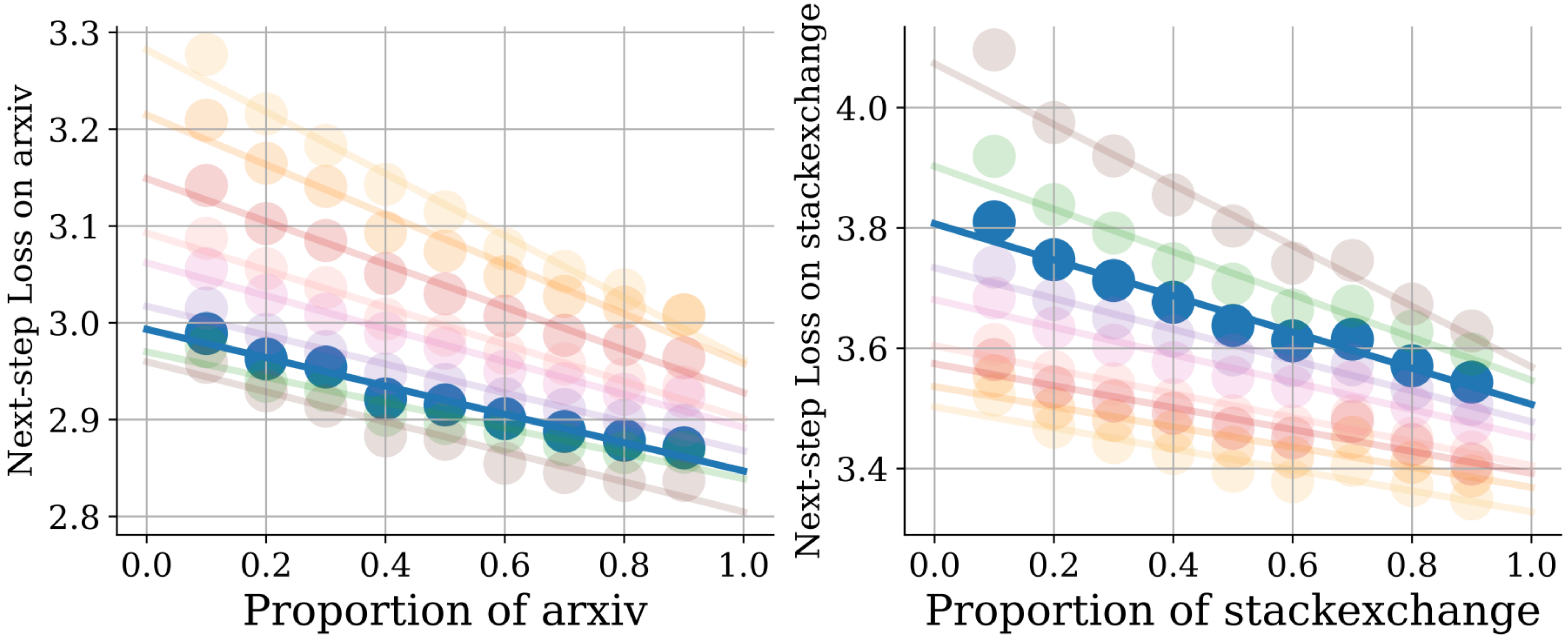
$$f_i^{t+1}(LM(p)) \approx f_i^t(LM(p)) - A_{i,t}^\top p^t$$

Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Dynamic setting: Aioli (Chen et al., 2024)

$$f_i^{t+1}(LM(p)) \approx f_i^t(LM(p)) - A_{i,t}^\top p^t$$

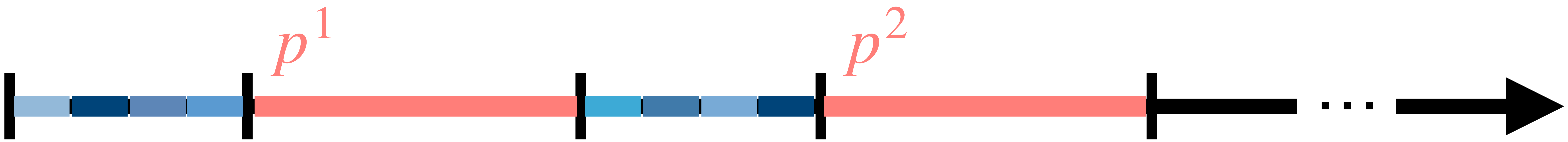
Linear dynamic mixing law on Arxiv/StackExchange



R^2 of dynamic mixing law on SlimPajama (7 domains): 0.938

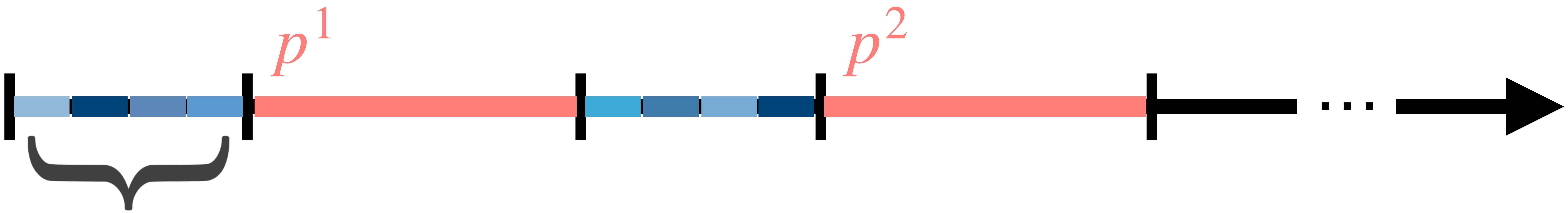
Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Dynamic Mixing Law: method

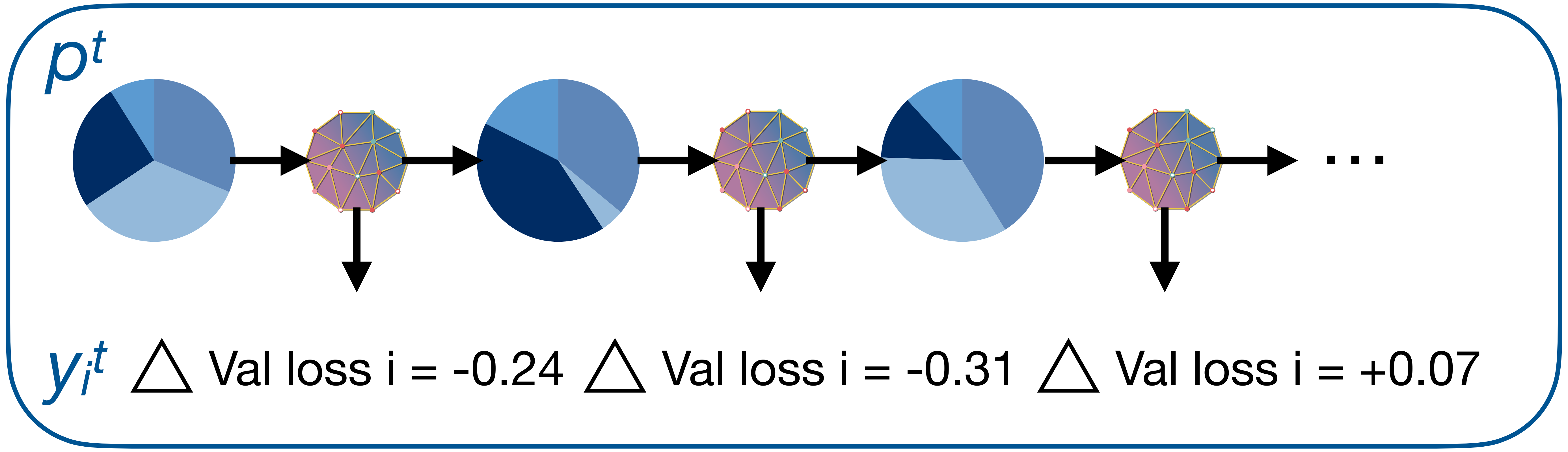


Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Dynamic Mixing Law: method

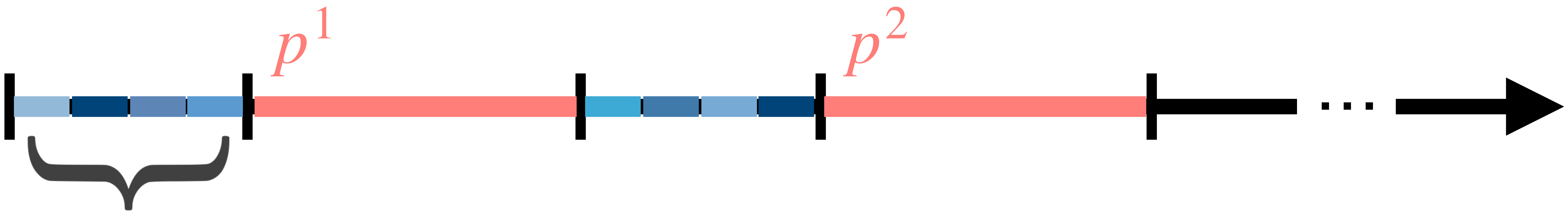


1. Explore



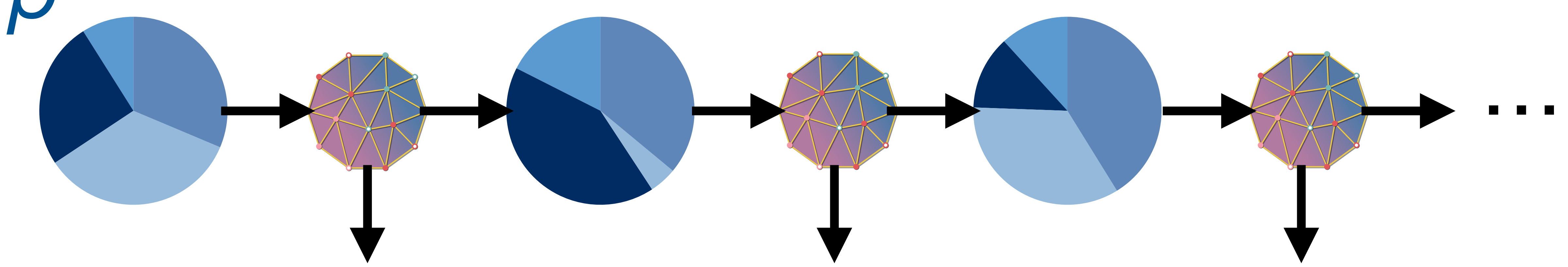
Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Dynamic Mixing Law: method



1. Explore

p^t

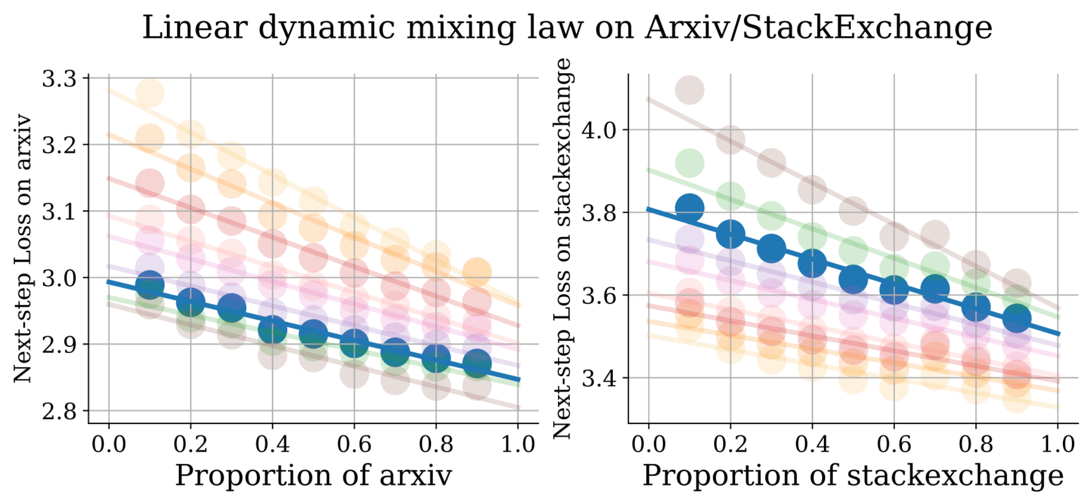


y_i^t △ Val loss i = -0.24 △ Val loss i = -0.31 △ Val loss i = +0.07

2. Fit

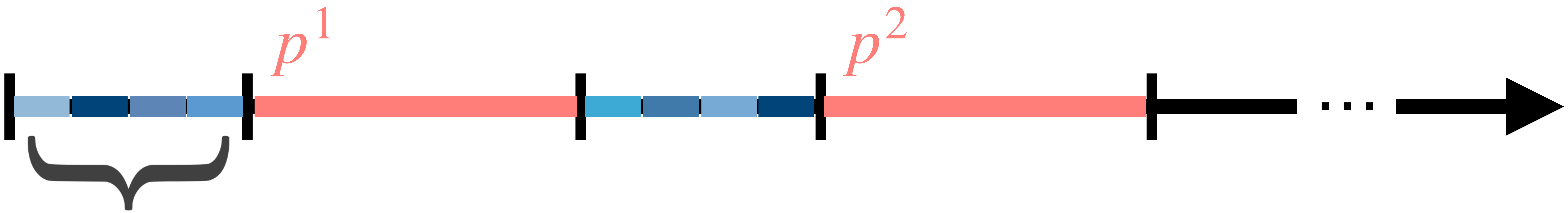
Use (p^t, y_i^t) to fit parameters of mixing law

$$\hat{f}_i^{t+1}(LM(p)) - \hat{f}_i^t(LM(p)) := -\hat{A}_{i,t}^\top p^t$$



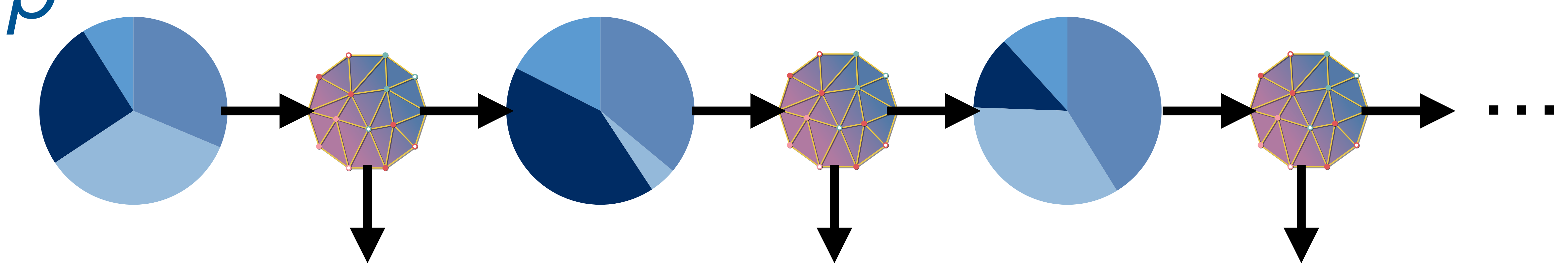
Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Dynamic Mixing Law: method



1. Explore

p^t

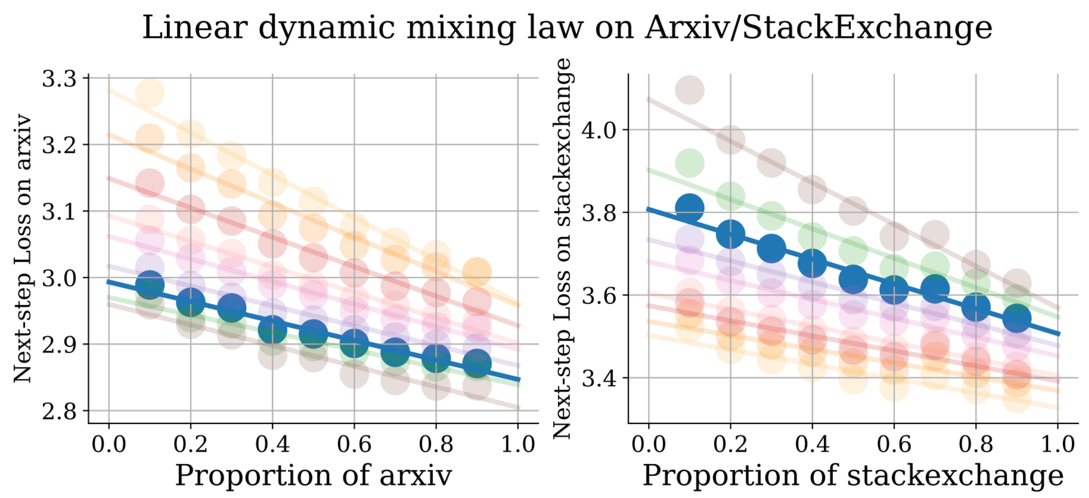


y_i^t △ Val loss i = -0.24 △ Val loss i = -0.31 △ Val loss i = +0.07

2. Fit

Use (p^t, y_i^t) to fit parameters of mixing law

$$\hat{f}_i^{t+1}(LM(p)) - \hat{f}_i^t(LM(p)) := -\hat{A}_{i,t}^\top p^t$$



3. Optimize

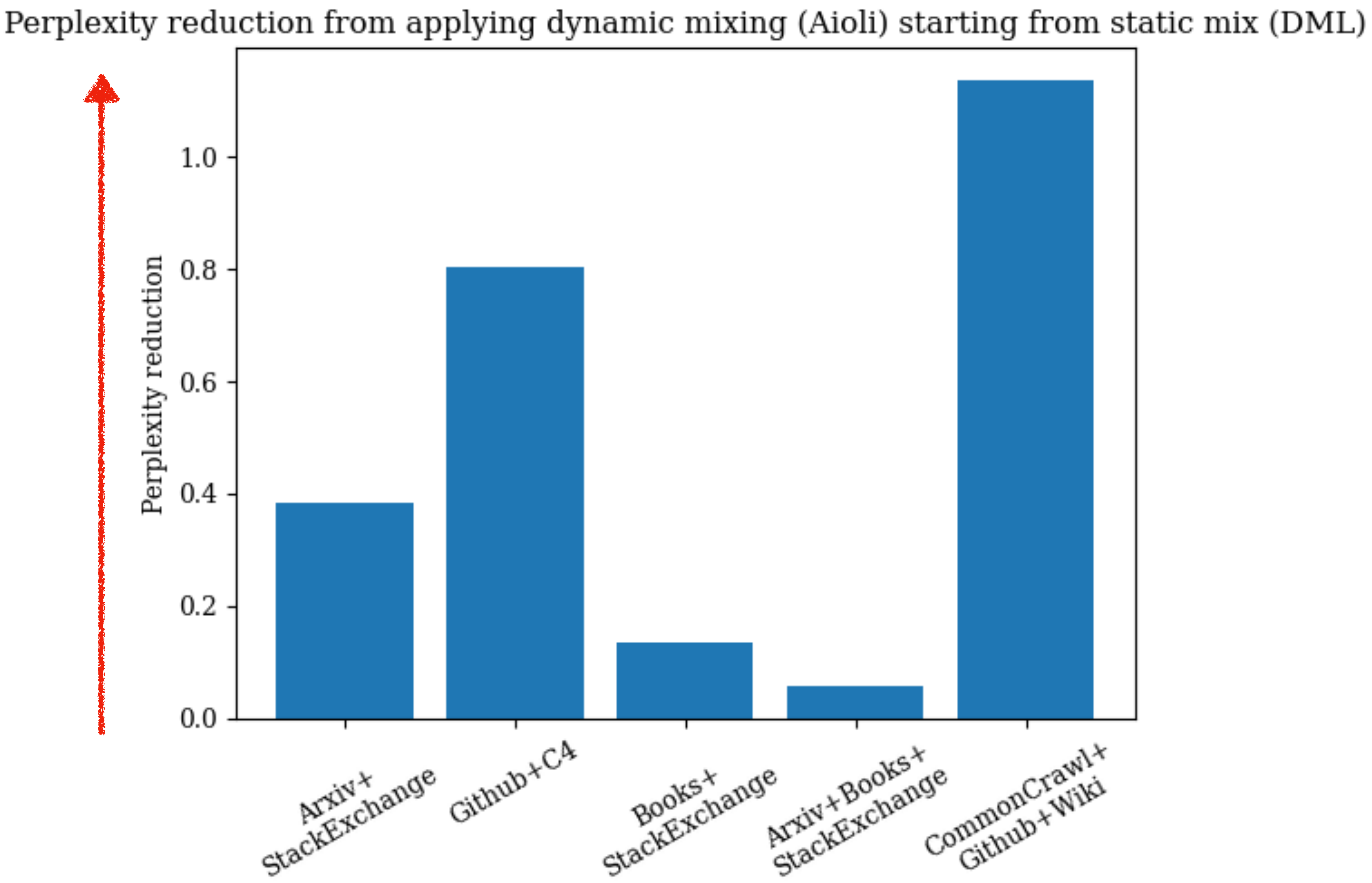
$$\text{minimize}_{p \in \triangle^{(m-1)\times T}} \frac{1}{n} \sum_{i=1}^{n} \hat{f}_i^T(LM(p))$$

$$\Rightarrow p_j^{t+1} \propto p_j^t \exp\left(\eta \sum_{i=1}^{m} \hat{A}_{ij,t}\right)$$

Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Dynamic Mixing Method: Results

Dynamic mixing improves over static mixing



Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Implications of mixing laws: improving understanding

Understanding existing methods

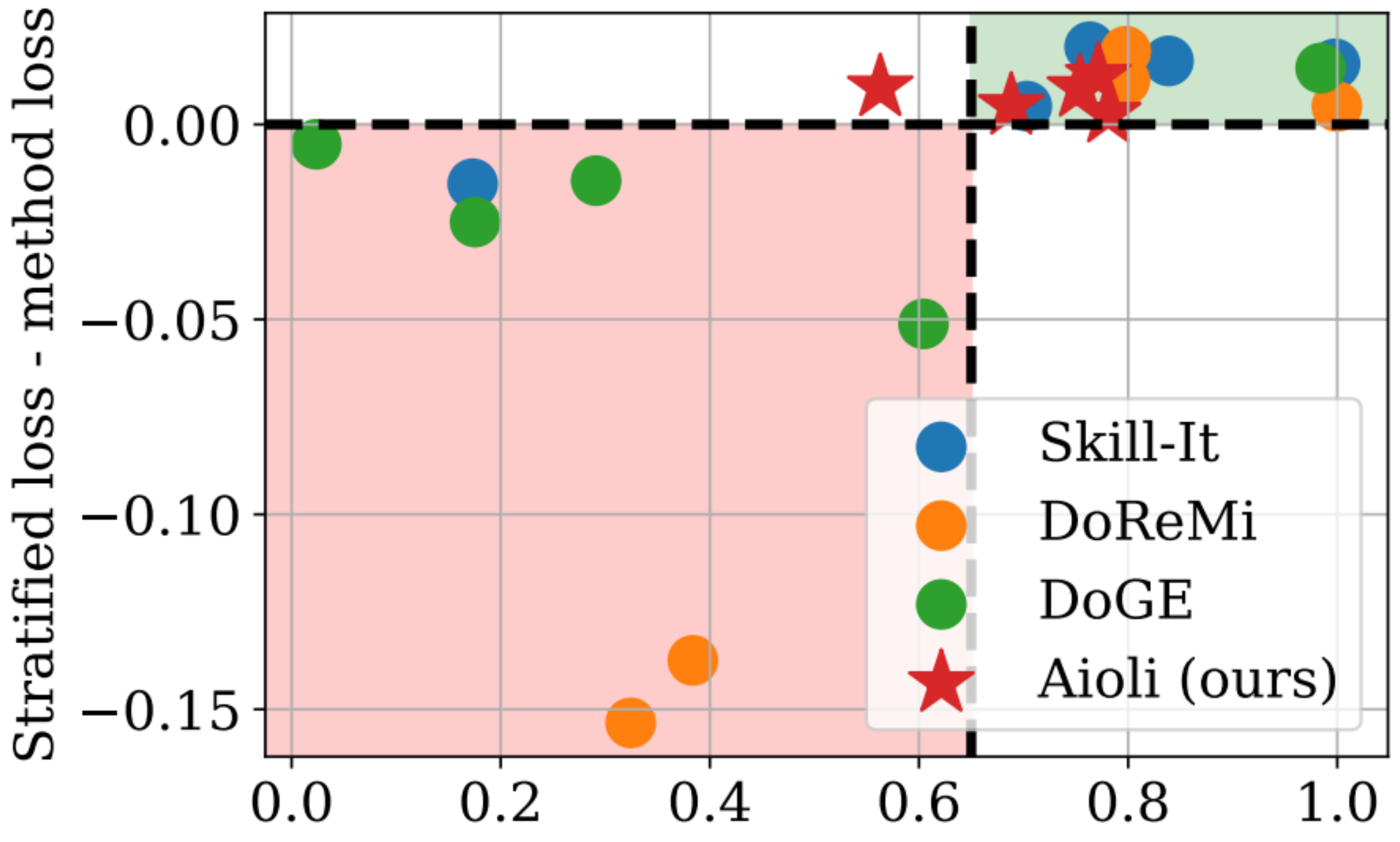
- Many mixing methods share the same meta-procedure: **explore, fit, and optimize**.
- May use different mixing laws implicitly

Method	A_t from $f_i^{t+1}(LM(p)) \approx f_i^t(LM(p)) - A_t^\top p^t$
DoReMi (Xie et al., 2023)	$A_{ii,t} = \min\{f_i^t(LM(p)) - f_i^T(LM(p_{ref})), 0\}$
DoGE (Fan et al., 2024)	$A_{ij,t} = \langle \nabla f_i^t(LM(p)), \nabla f_j^t(LM(p)) \rangle$
Skill-It (Chen et al., 2023)	$A_{ij,t} = f_i^t(LM(p)) \cdot \dfrac{f_i^T(LM(\mathbf{1}_j)) - f_i^1(LM(\mathbf{1}_j))}{f_i^1(LM(\mathbf{1}_j))}$
Aioli (Chen et al., 2024)	Learned from fitting data to dynamic mixing law

Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Understanding existing methods

- Performance of existing method is correlated with the accuracy of its implicit mixing law
- Hardcoded params can produce inconsistent gains



Similarity between hardcoded and true mixing law parameters

Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Understanding how models learn from data

$$f_i(LM(p)) \approx b_i \sigma(-A_i^\top p) + c_i$$

Recall interpretation: each domain linearly contributes A_{ij}, a “score” for how much domain j impacts validation dataset i. What does $A \in \mathbb{R}^{n \times m}$ actually look like?

- If *A* is sparse and does not change over time, life is easy but boring

Understanding how models learn from data

1. *A* matrix has asymmetries; not just one domain affecting one validation task

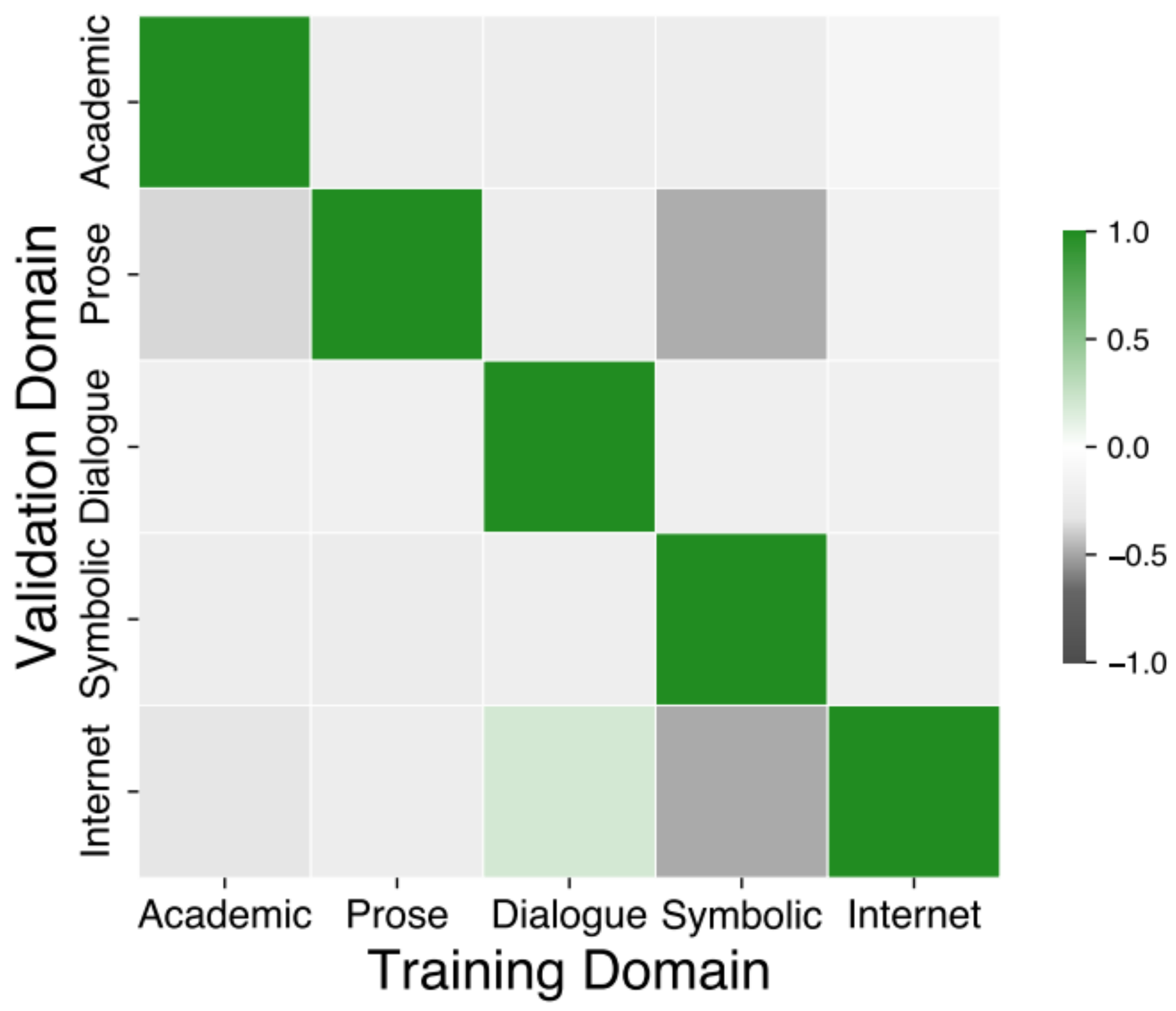
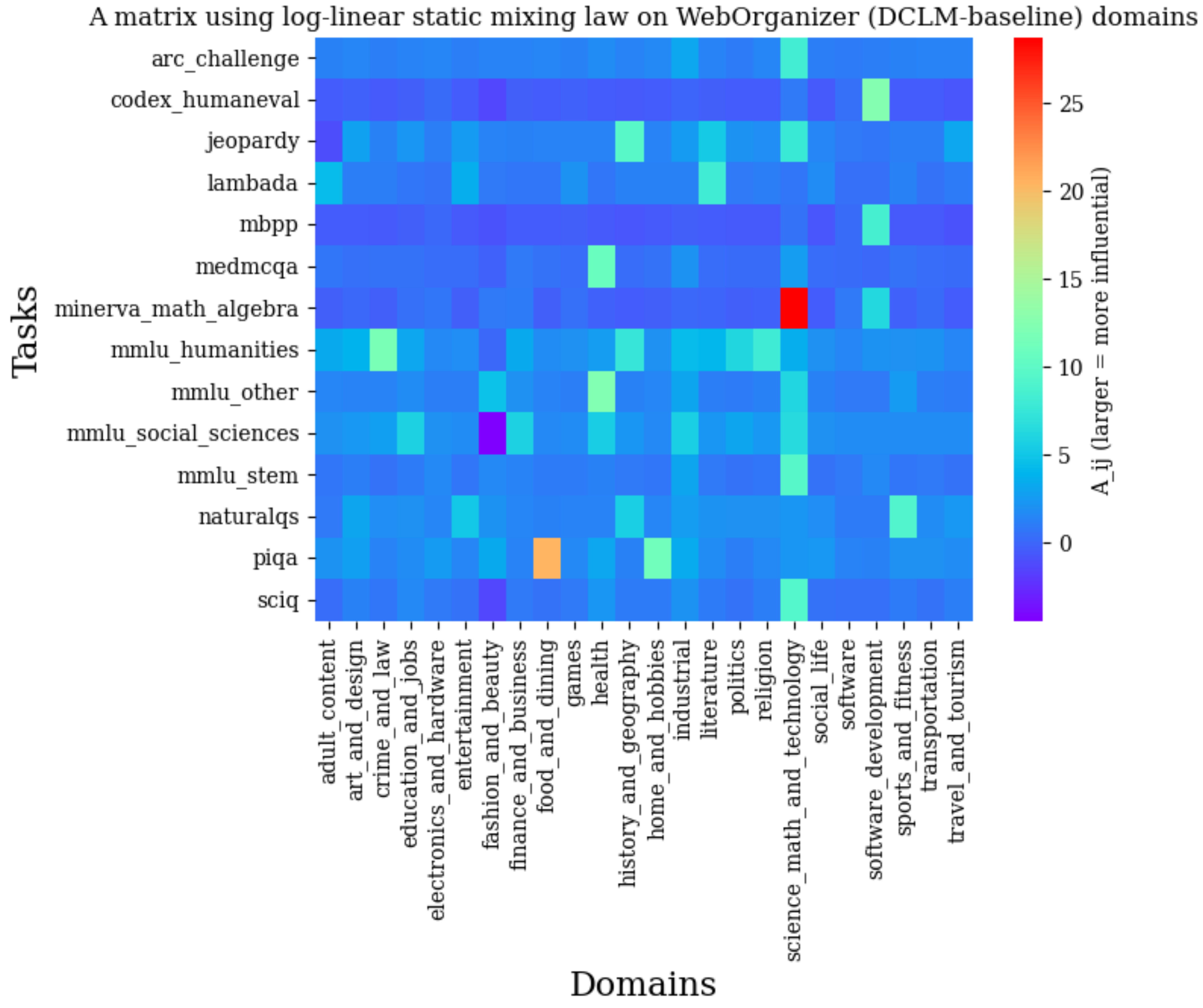


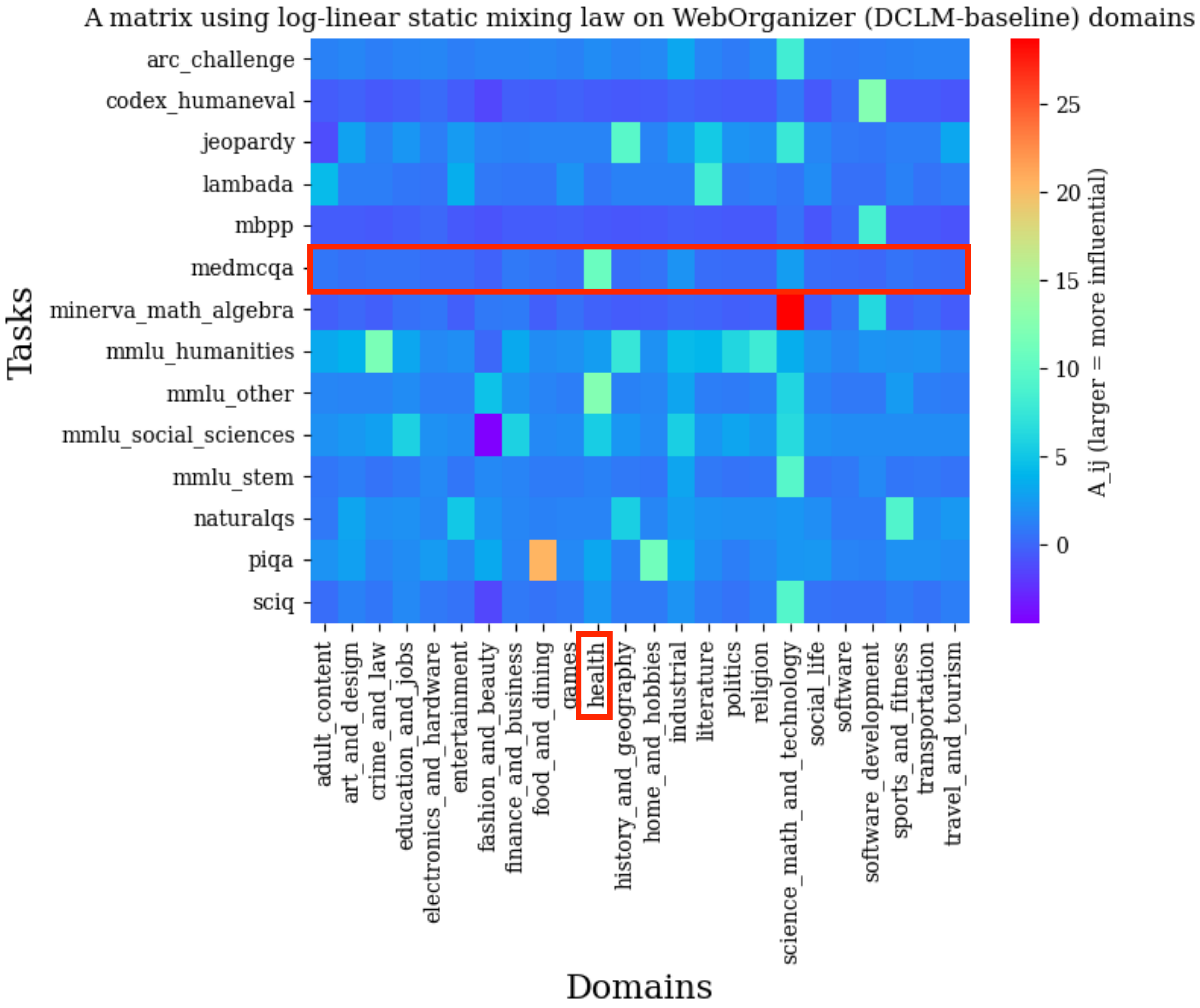
Figure 4: The interaction between different training and validation domains on the Pile. Each boxes are fitted normalized t_{ij} from Eqn. 7. We normalize the value by t_{ij} with the maximum absolute value for each validation set i (i.e., $t_{ij}/t_{i,\arg\max_j |t_{ij}|}$). A larger value (greener) indicates more mutual facilitation.



 Ye et al., 2024. Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance

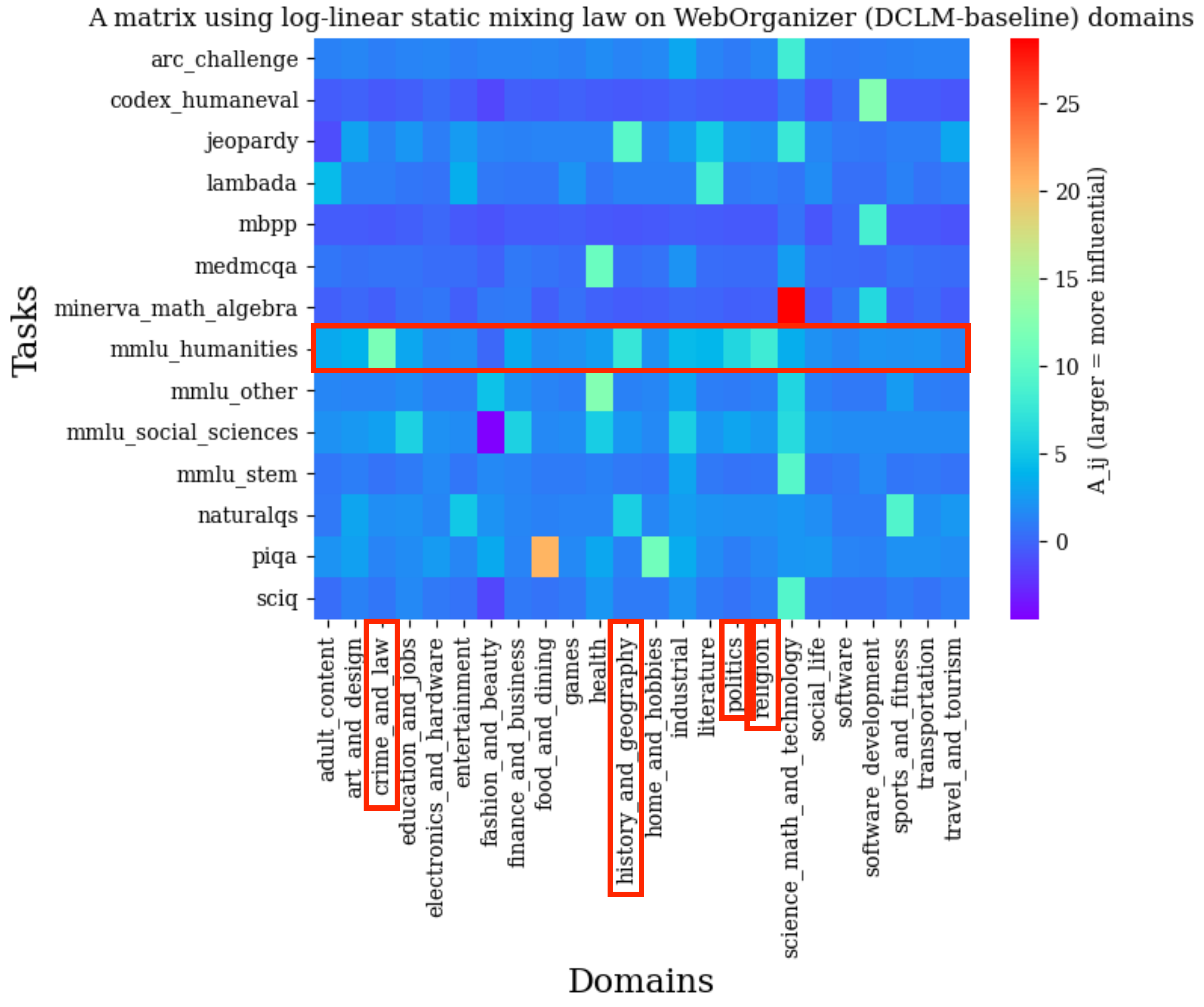
Understanding how models learn from data

1. *A* matrix has asymmetries; not just one domain affecting one validation task



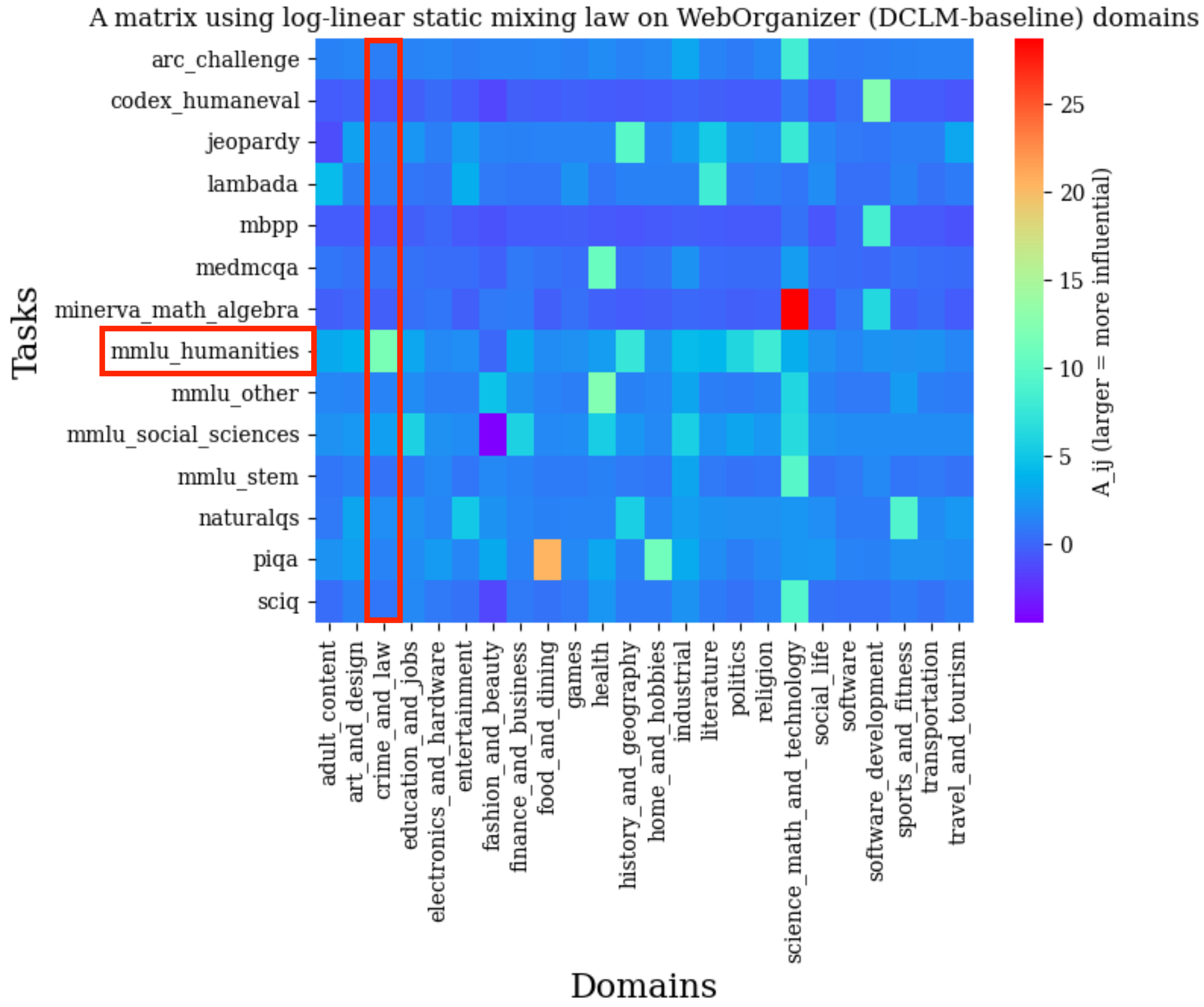
Understanding how models learn from data

1. *A* matrix has asymmetries; not just one domain affecting one validation task



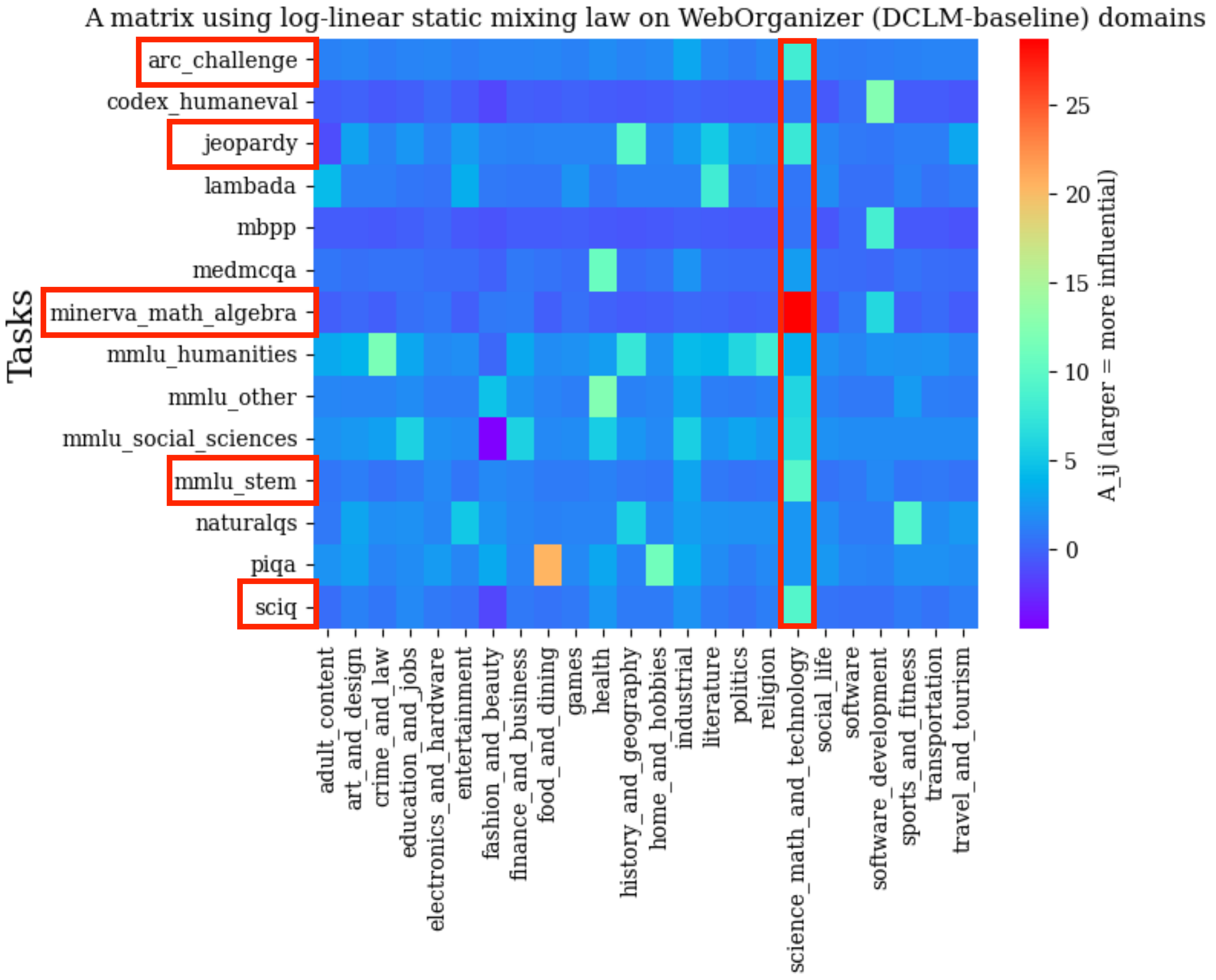
Understanding how models learn from data

1. *A* matrix has asymmetries; not just one domain affecting one validation task



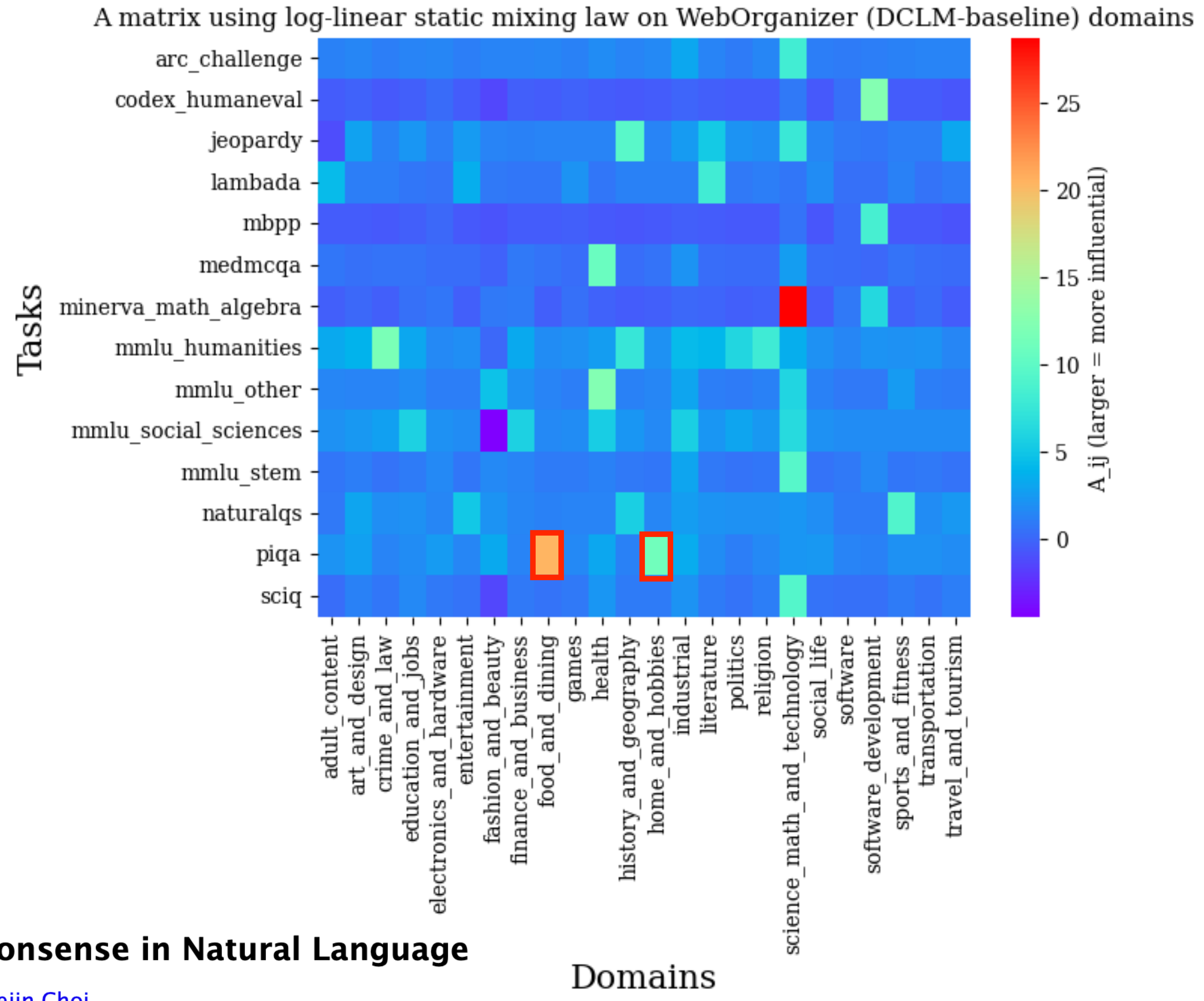
Understanding how models learn from data

1. *A* matrix has asymmetries; not just one domain affecting one validation task



Understanding how models learn from data

1. *A* matrix has asymmetries; not just one domain affecting one validation task



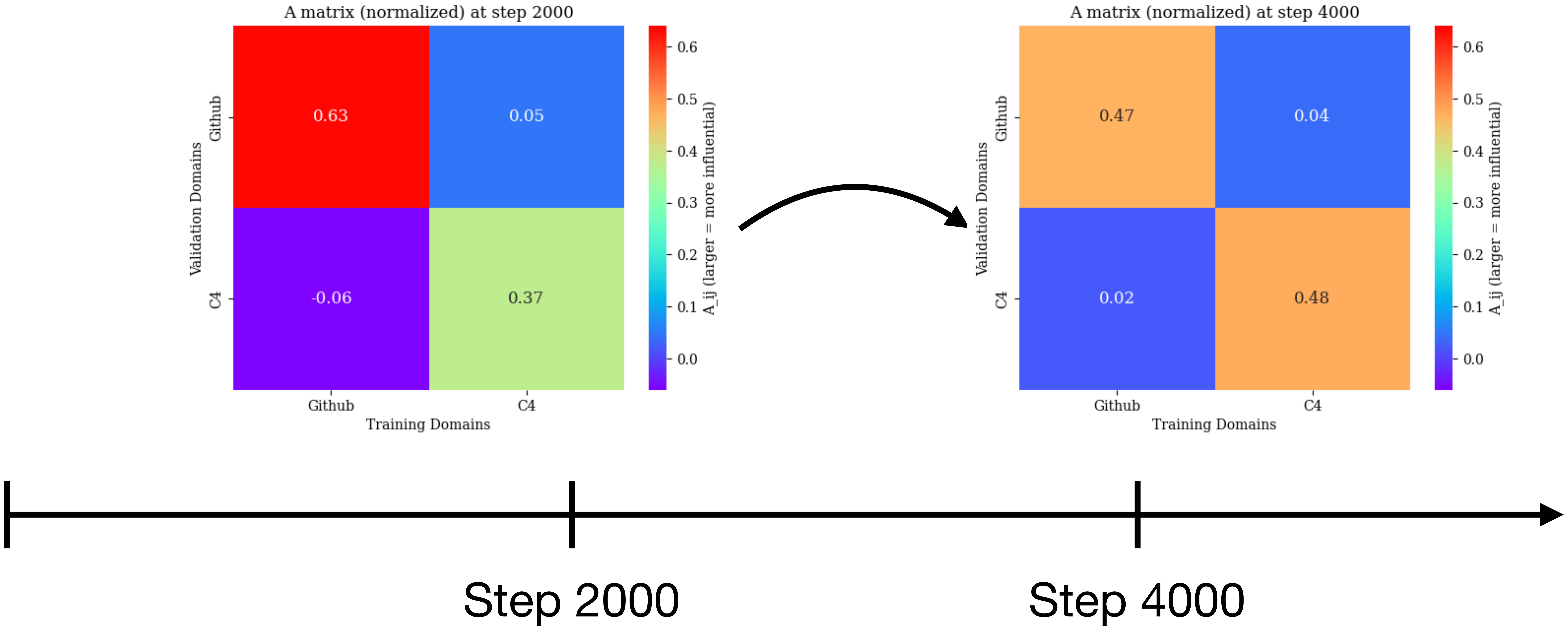
PIQA: Reasoning about Physical Commonsense in Natural Language

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, Yejin Choi

To apply eyeshadow without a brush, should I use a cotton swab or a toothpick? Questions requiring this kind of physical

Understanding how models learn from data

2. *A* matrix can change over time



Chen et al., 2024. Aioli: A Unified Optimization Framework for Language Model Data Mixing

Summary

- Data development pipeline: acquire (quantity), transform (quality), mix (composition)
- Mixing is an critical step that allows us to align the data distribution with a set of desired model capabilities, navigate tradeoffs
- Key development: performance is often roughly linear in the data mix!
- Mixing methods should exploit this structure to produce good mixes efficiently.

Looking forward

- What should a domain be?
  - We can mix across *any unit.*
  - Domains as sources (conventional), vs. topics and formats
- Can we use mixing to understand how to better acquire data? What can the *A* matrix tell us about what data the model needs the most?

Thank you!

Some suggested readings:

- General data development:
  - The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale
  - BeyondWeb: Lessons from Scaling Synthetic Data for Trillion-scale Pretraining
  - DataComp-LM: In search of the next generation of training sets for language models
  - Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research
- Mixing:
  - Data Mixing Laws: Optimizing Data Mixtures by Predicting Language Modeling Performance
  - Aioli: A Unified Optimization Framework for Language Model Data Mixing
  - Organize the Web: Constructing Domains Enhances Pre-Training Data Curation

Email: mfchen@stanford.edu