Scaling Language Models

Weijia Shi

CSE 517 and 447: Natural Language Processing

💡Feel free to interrupt with questions / thoughts anytime!

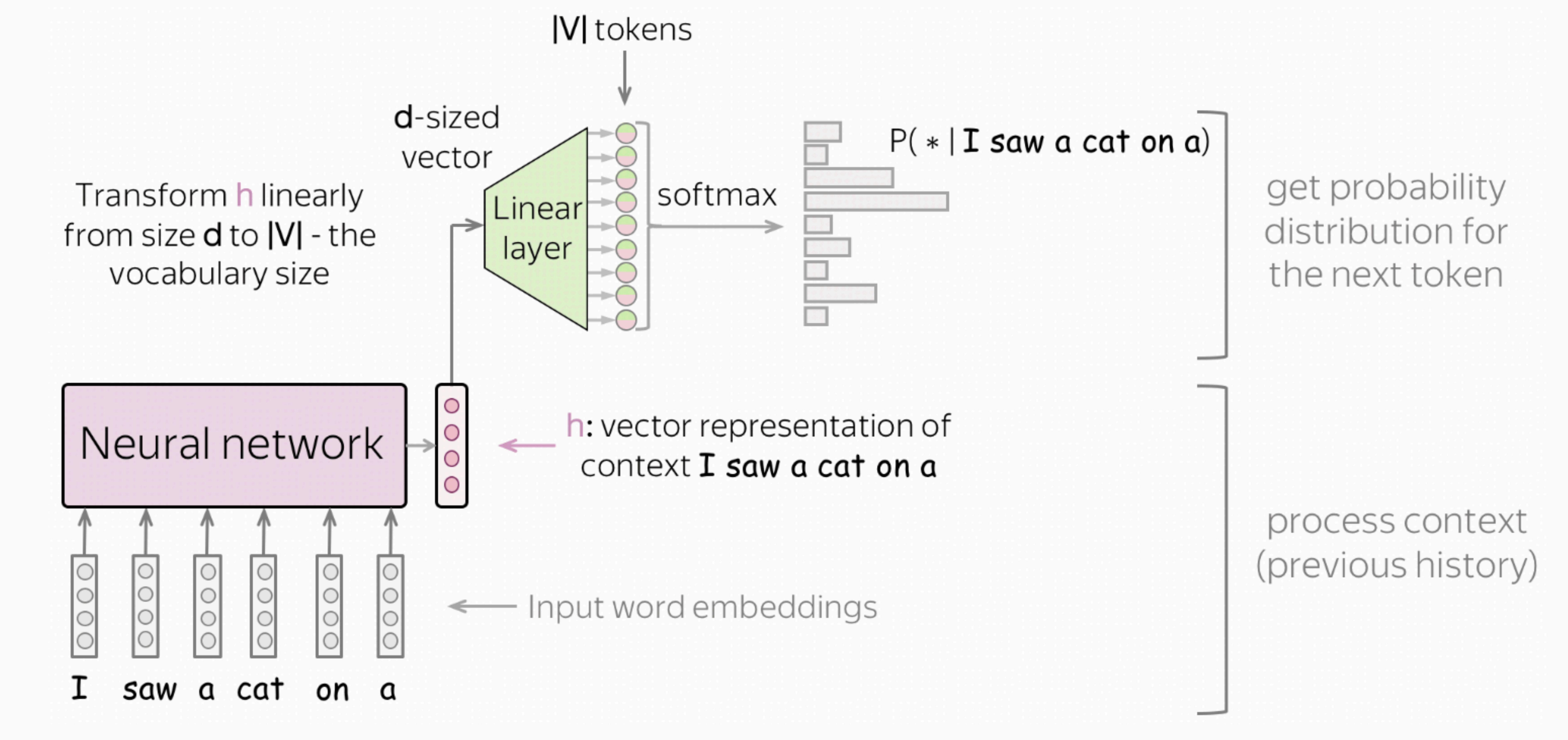
Some slides are adapted from previous presentations by Sean Welleck, Yue Xiang, Niklas Muennighoff, Quentin Anthony, Liwei Jiang

Recap of Language Modeling

Autoregressive Language Modeling

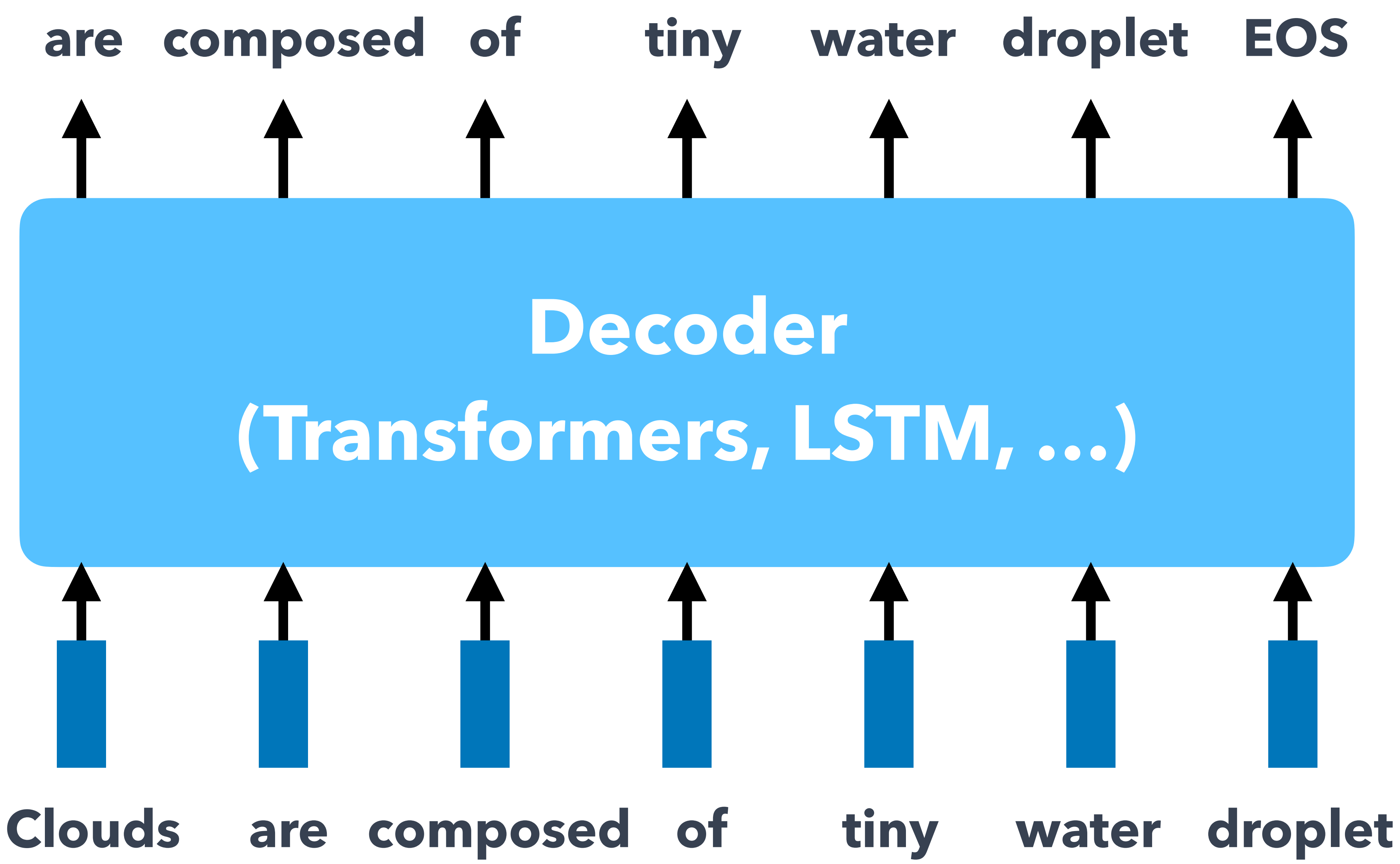
$$P(X) = \prod_{i=1}^{I} P(\underbrace{x_i}_{\text{Next Token}} \mid \underbrace{x_1, \ldots, x_{i-1}}_{\text{Context}})$$

Autoregressive Language Modeling



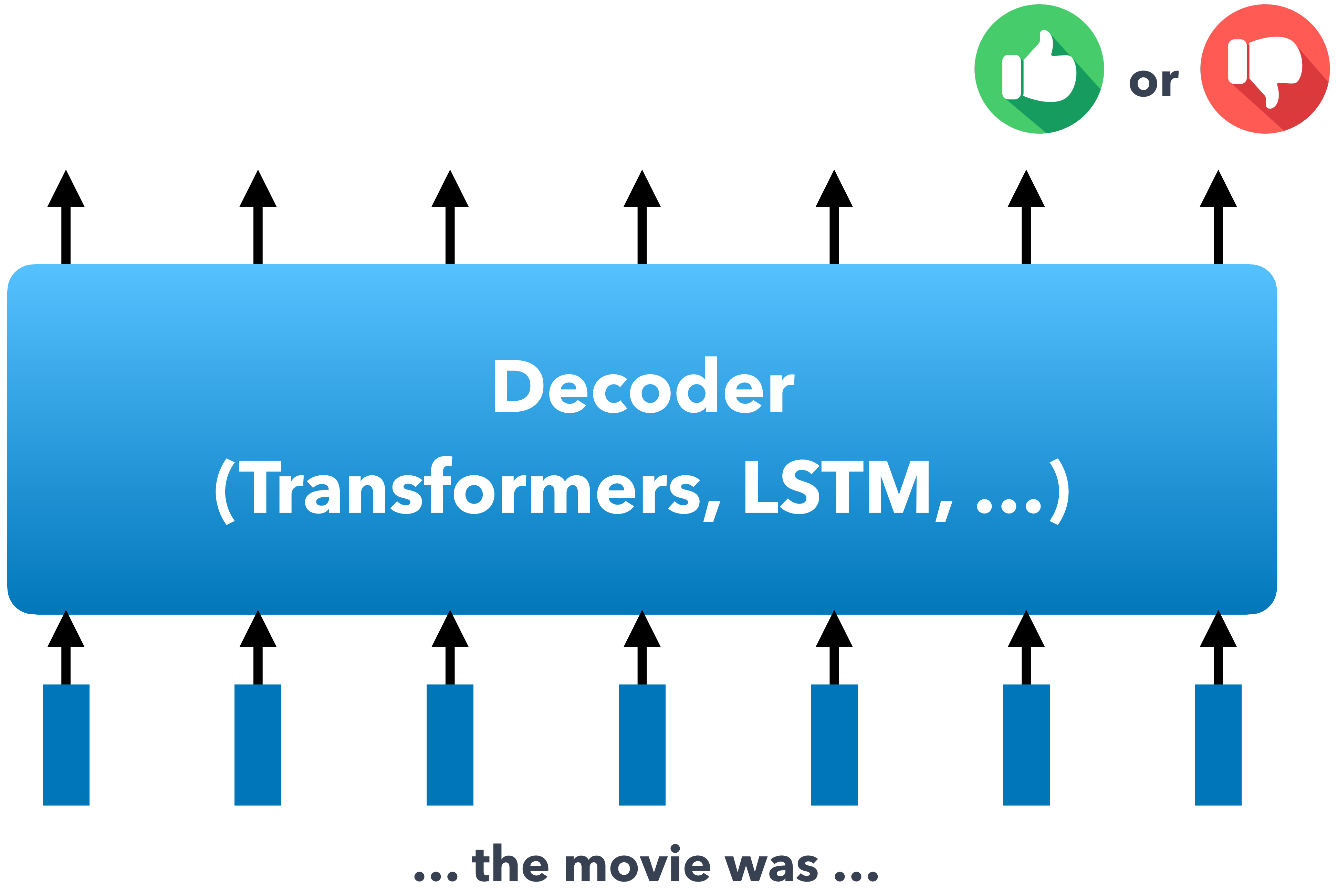
Why Is It Called Pretraining?

Step 1:
Pre-training



Abundant data; learn general language

Step 2:
Fine-tuning



Limited data; adapt to the task

"Pre"training happens before training (fine-tuning)!

Outline

- Overview of LM Pretraining
- Pretraining Data
- Pretraining Setups
- Scaling Law

Outline

- Overview of LM Pretraining
- Pretraining Data
- Pretraining Setups
- Scaling Law

Imagine you’re developing Llama

Overview of Llama Training

Pretraining -> Instruction Fine-tuning -> RLHF

Why Pretraining?

Lots of Information in Raw Texts

Verb	I went to Hawaii for snorkeling, hiking, and whale _watching_.
Preposition	I walked across the street, checking for traffic __over__ my shoulders.
Commonsense	I use ___knife___ and fork to eat steak.
Time	Ruth Bader Ginsburg was born in ___1933___.
Location	University of Washington is located at __Seattle__, Washington.
Math	I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, __34__.
Chemistry	Sugar is composed of carbon, hydrogen, and _oxygen_.

Why Pretraining?

Pretraining Extracts Patterns, Structures, and Semantic Knowledge from Raw Texts

Outline

- Overview of LM Pretraining
- Pretraining Data
- Training Setups
- Scaling Law

What Matters for Pretraining Data?

- **Quantity:** How much data do I have?
- **Quality**: Is it beneficial for training?
- **Coverage**: Does the data cover enough domains for the end task?

Scale Up Data Quantity

	Tokens of training data
Llama 1	1.4 trillion
Llama 2	1.8 trillion
Llama 3	15 trillion
Deepseek 3	15 trillion

How Large are 1T Tokens?

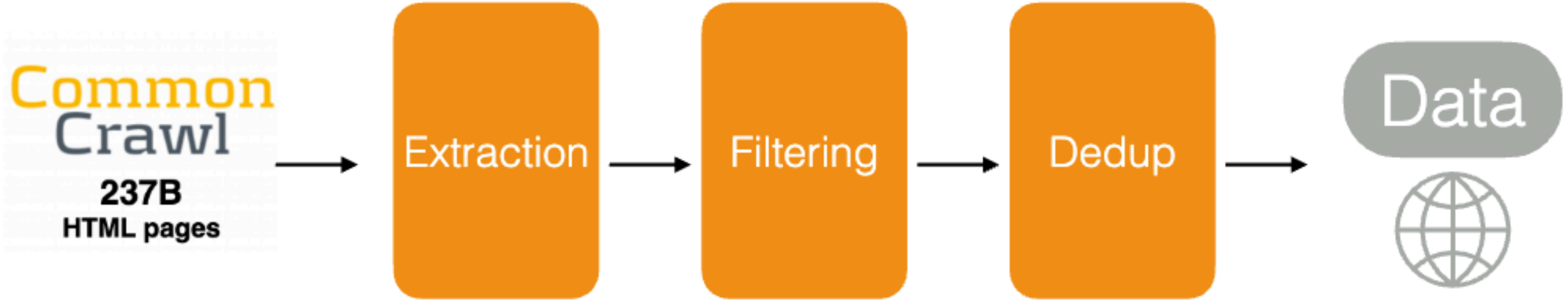
Physical Size (if printed)
- Average words per page: A typical page contains about 300-500 words.
- Words from 1 trillion tokens: Assuming 750 billion words, and an average of 400 words per page:
  - Total Pages: Approximately 1.875 billion pages.

Digital Storage
- Character Encoding: Assuming each character takes up 1 byte (in a simple encoding like ASCII), 1 trillion tokens (4 trillion characters) would require about 4 terabytes (TB) of storage.

Reading Time
- Reading Speed: The average reading speed is about 200-250 words per minute.
- Time to Read 750 Billion Words: At 200 words per minute, it would take about 3.75 billion minutes, or approximately 7,125 years of continuous reading.

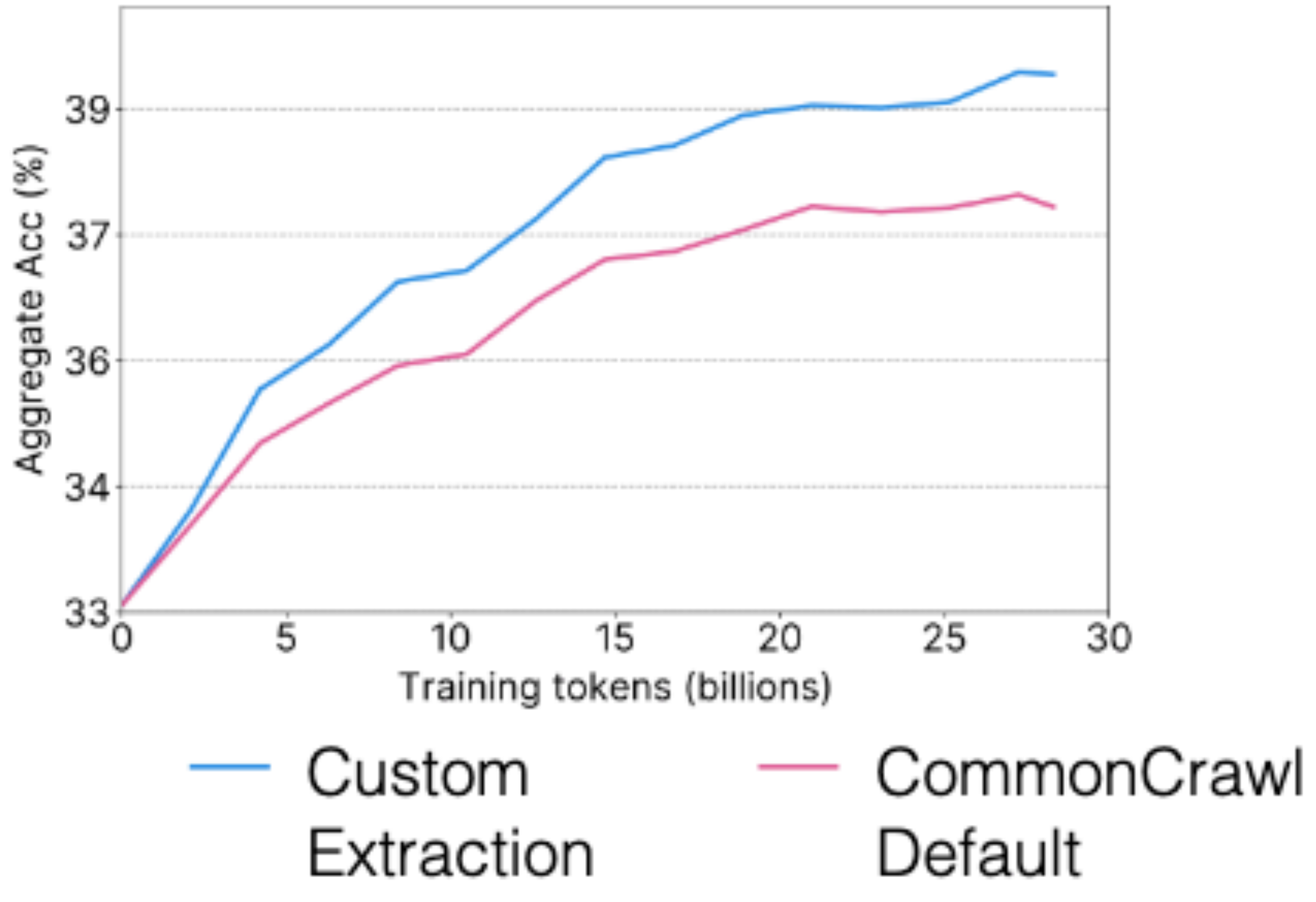
Pretraining Data Comes From Web

- Large snapshots of web pages.
  - Extraction: HTML to text
  - Filtering: filter out unwanted pages
  - Deduplication: many duplicate web pages



Extraction

- Extraction: HTML to text
  - Remove boilerplate
  - Retain Latex, code, etc.



Penedo et al 2024

```
This paper concerns the quantity
<img src="https://s0.wp.com/
latex.php?latex=%7BM%28x%29..."
alt="{M(x)}" />, defined as the
length of the longest
subsequence of the numbers from
```

Image Equations

```
Suppose I have a smooth map
[tex]f\colon \mathbb{R}^3
\longrightarrow S^2[/tex]. If I
identify [tex]\mathbb{R}^3[/tex]
with [tex]U_S = S^3 - \
{(0,0,1)\}[/tex] via
stereographic projection
```

Delimited Math

```
<math>
  <semantics>
    ...
    <annotation ...>
      {\displaystyle \mathrm {MA}
      ={\frac{f_{O}}{f_{E}}}}
    </annotation>
  </semantics>
</math>
```

Special Tags

Filtering

- Filter out unwanted text
  - Language filter
  - Repetitions
  - Too many short lines
  - …

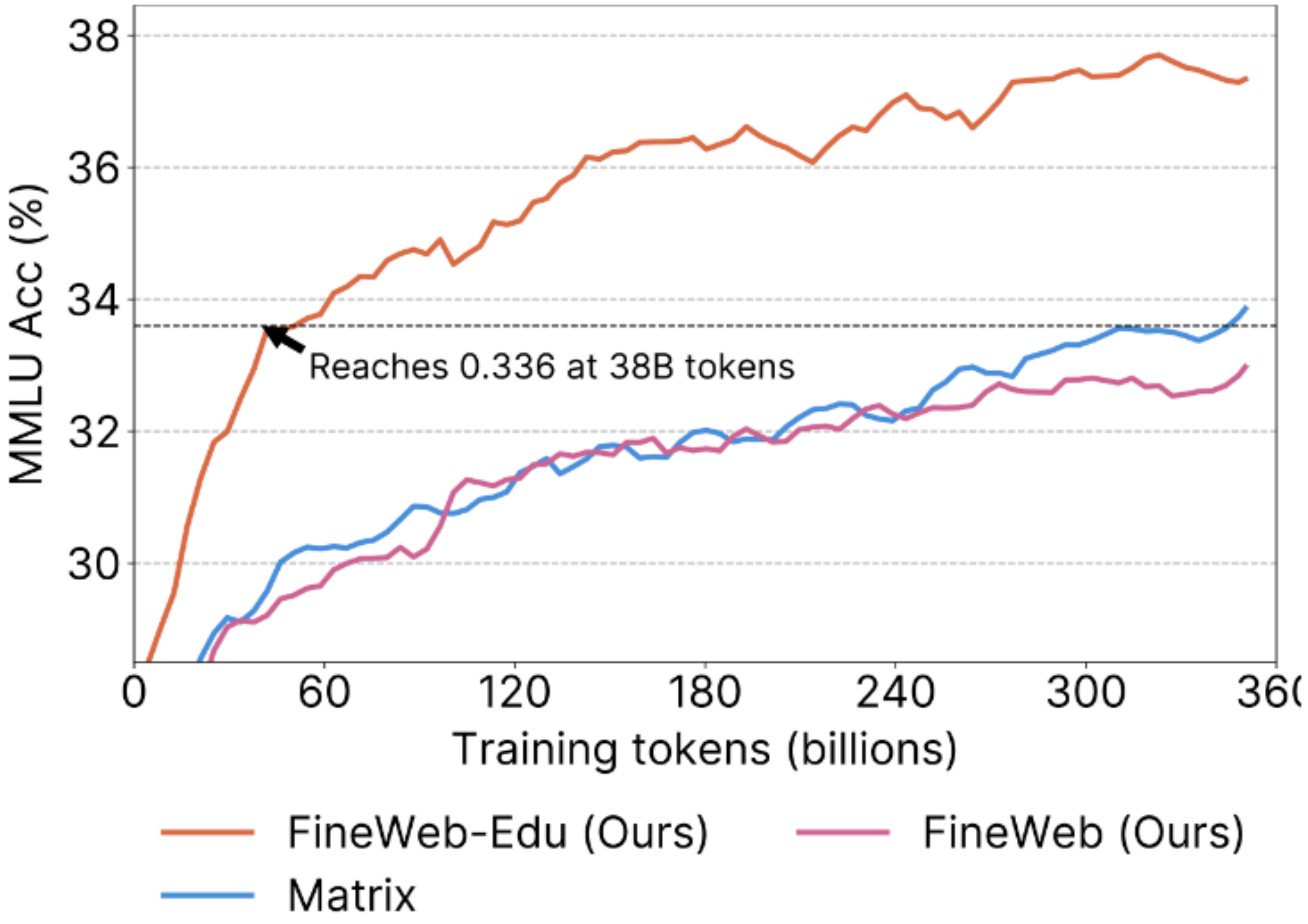
Deduplication

- Remove duplicate content
    - Fuzzy strategy: *minhash*

Data Quality: Model-based Selection

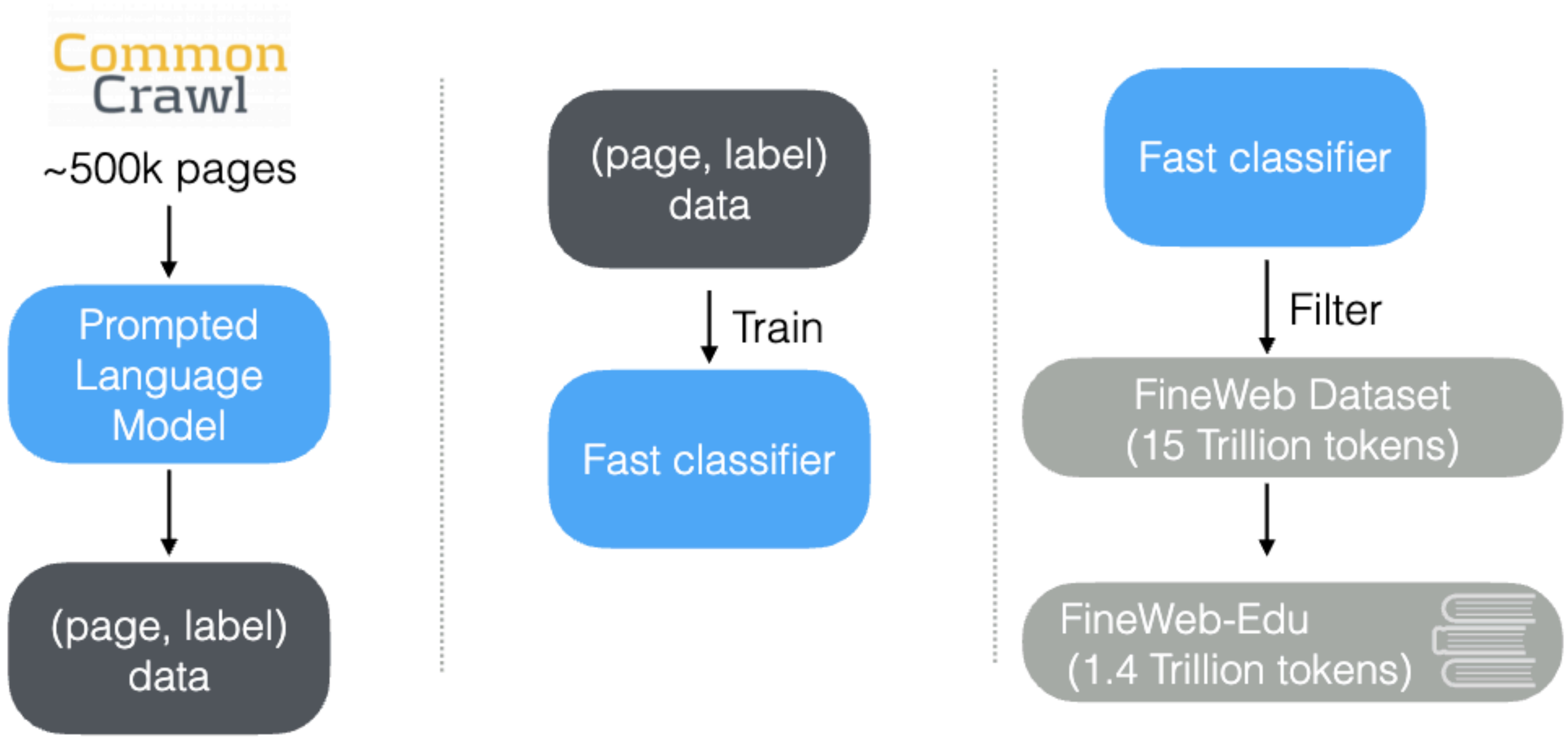
- Example: **FineWeb-Edu** [Penedo et al 2024]

  - Classifier to classify pages as “educational”



Data Quality: Model-based Filtering

- Example: **FineWeb-Edu** [Penedo et al 2024]
  - Classifier to classify pages as "educational"



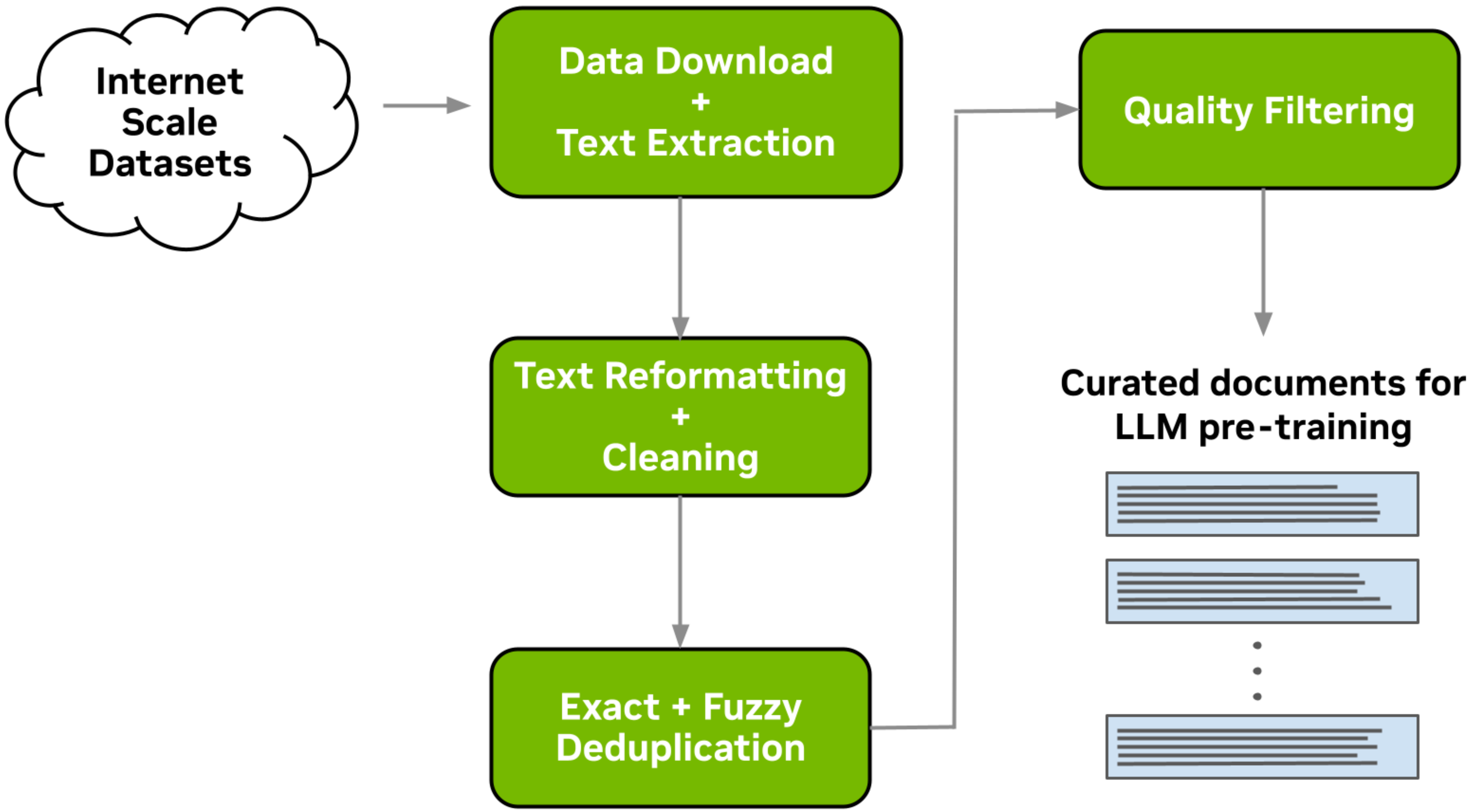
Data Coverage: Mixtures

- **Training Data is a mixture of different sources**

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Most model builders keep pretraining data private; Llama shared some details but not the data itself

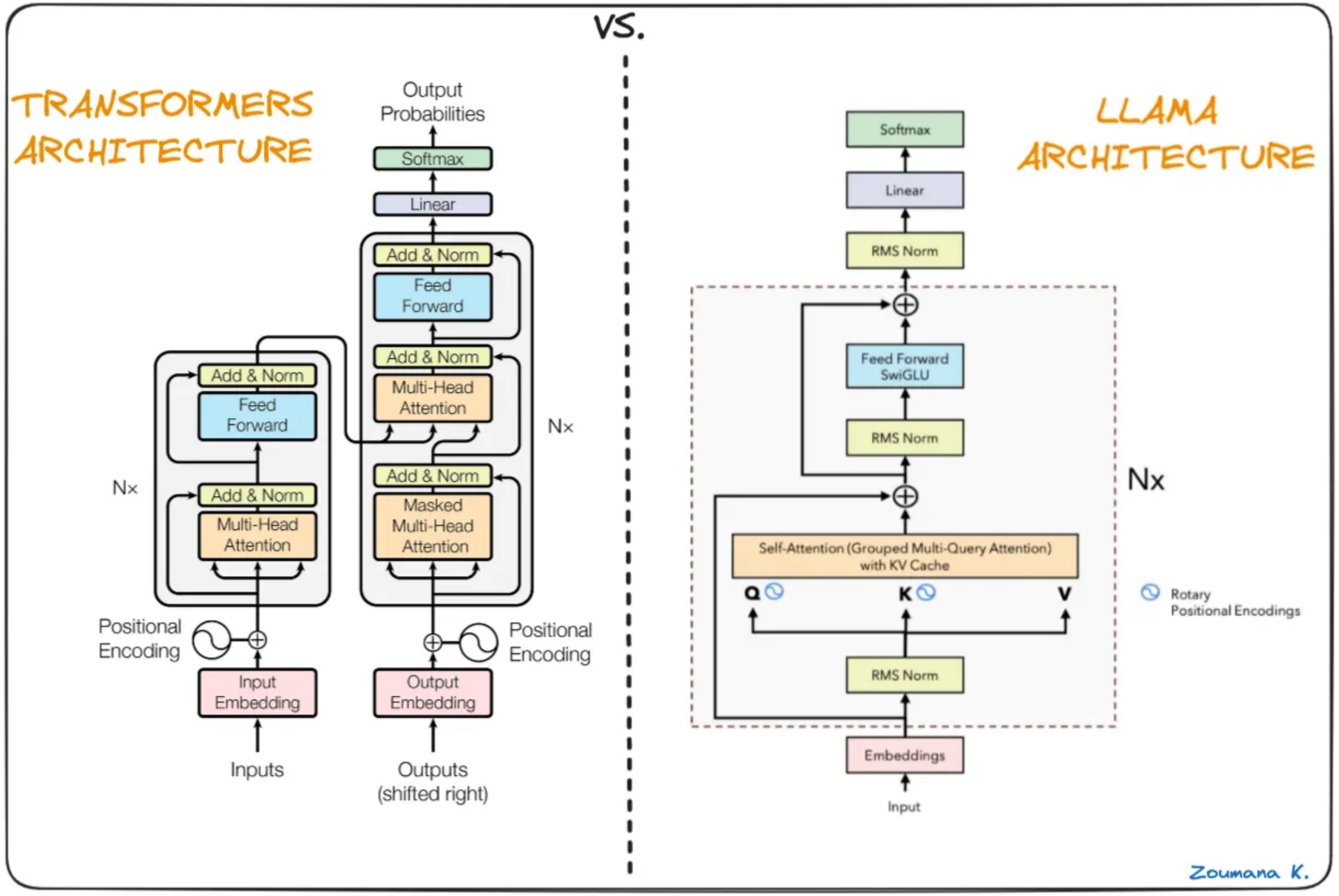
Pretraining Data Summary



Outline

- Overview of LM Pretraining
- Pretraining Data
- Training Setups
- Scaling Law

Architecture (Recap)



Llama Architecture: Grouped Query Attention

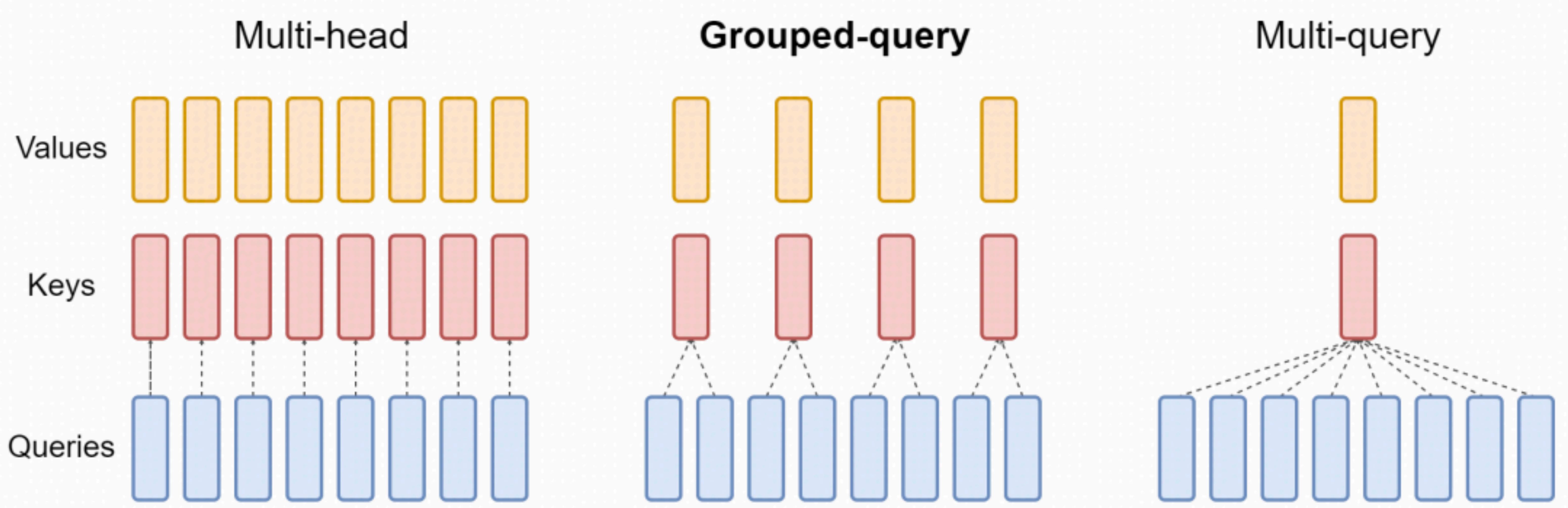


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.

Model	T_{infer}	Average	CNN	arXiv	PubMed	MediaSum	MultiNews	WMT	TriviaQA
	s		R_1	R_1	R_1	R_1	R_1	BLEU	F1
MHA-Large	0.37	46.0	42.9	44.6	46.2	35.5	46.6	27.7	78.2
MHA-XXL	1.51	47.2	43.8	45.6	47.5	36.4	46.9	28.4	81.9
MQA-XXL	0.24	46.6	43.0	45.0	46.9	36.1	46.5	28.5	81.3
GQA-8-XXL	0.28	47.1	43.5	45.4	47.7	36.3	47.2	28.4	81.6

Llama Architecture: Other Setups

params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

Llama Architecture: Other Setups

params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

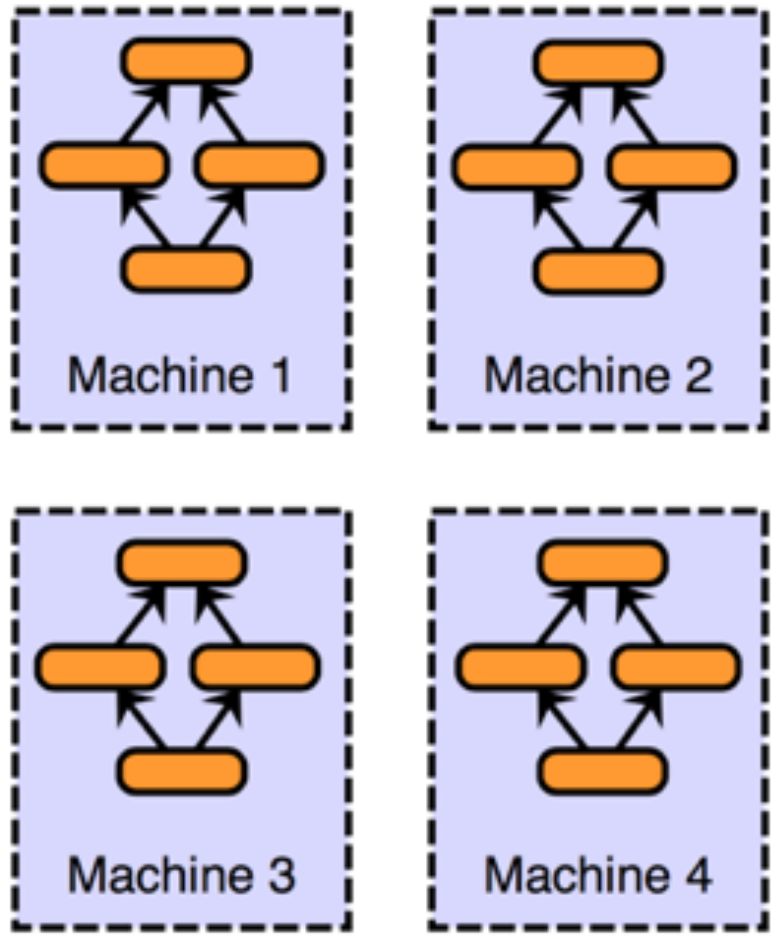
Cannot fit with 1 GPU's memory. How to train?

Bottlenecks of Training Big Models

- **Bottleneck 1:** Iteration time ⌚
  - Each training sample takes longer to propagate through more parameters
- **Bottleneck 2**: Processor Memory 🧠
  - Billions of parameters and 2 bytes each, GPUs have comparatively little memory!

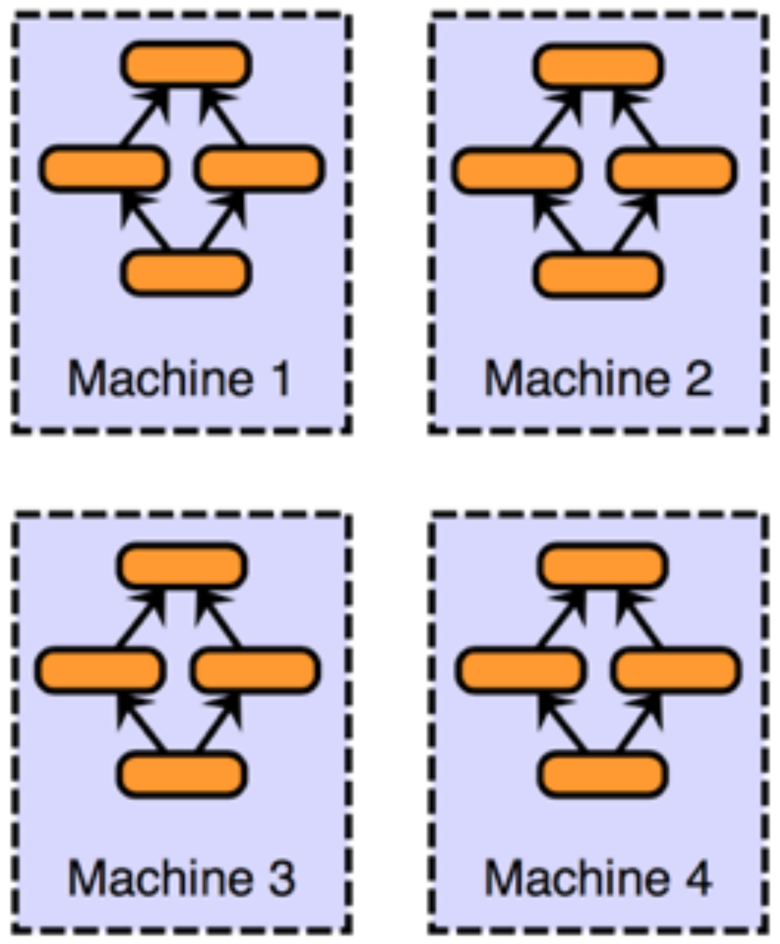
Parallelization Strategies

Data Parallelism

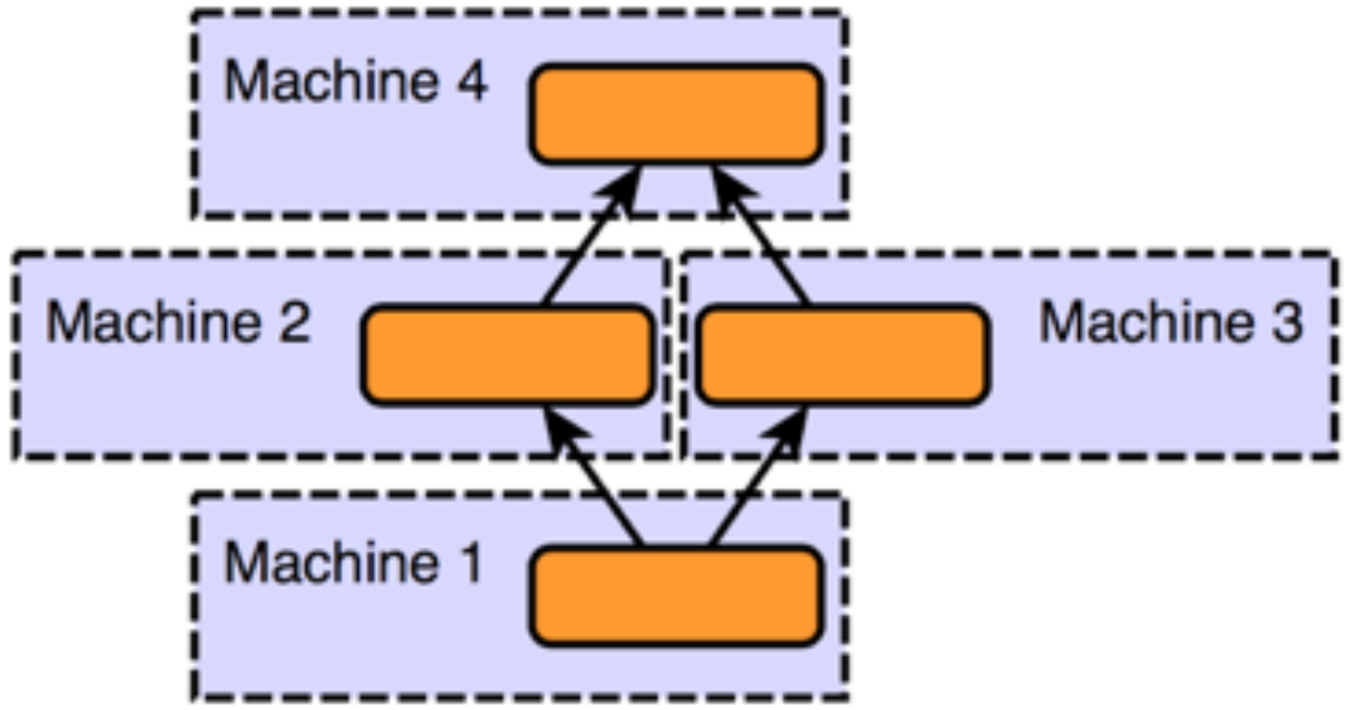


Parallelization Strategies

Data Parallelism



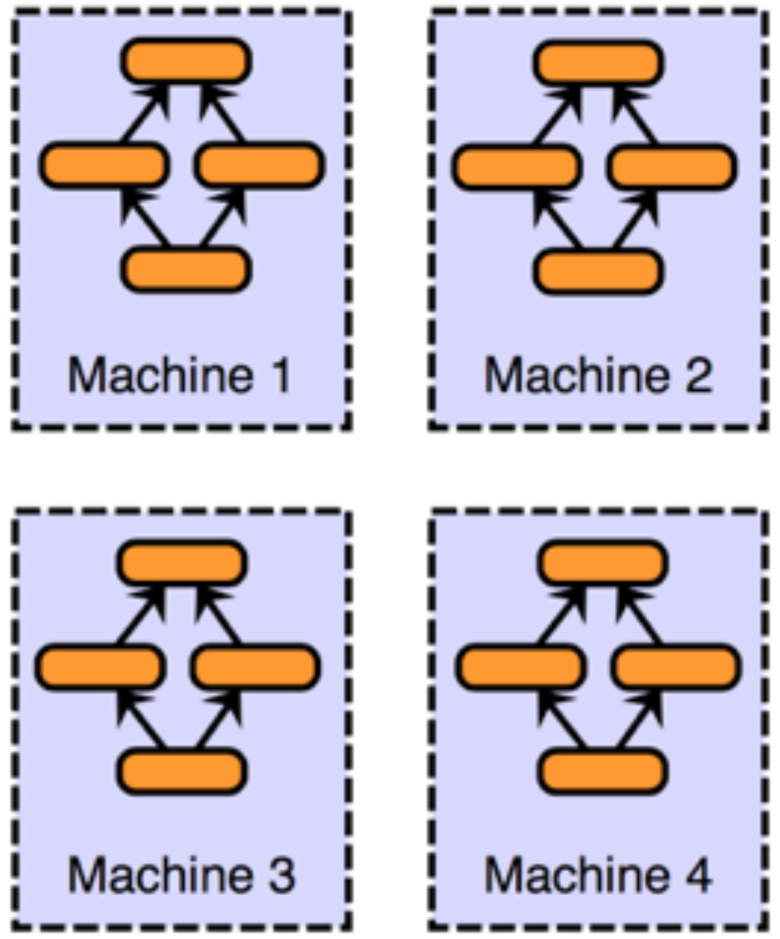
Model Parallelism



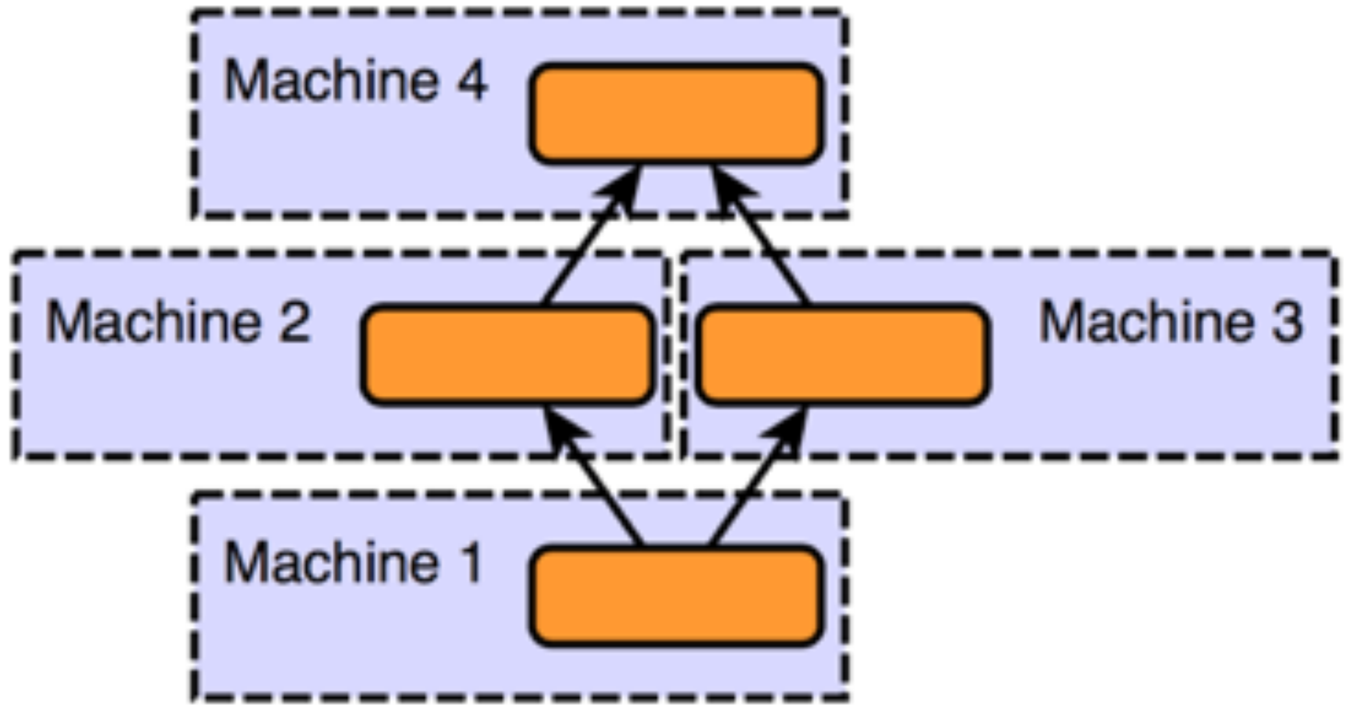
Hybrid (Model and Data) Parallelism

Parallelization Strategies

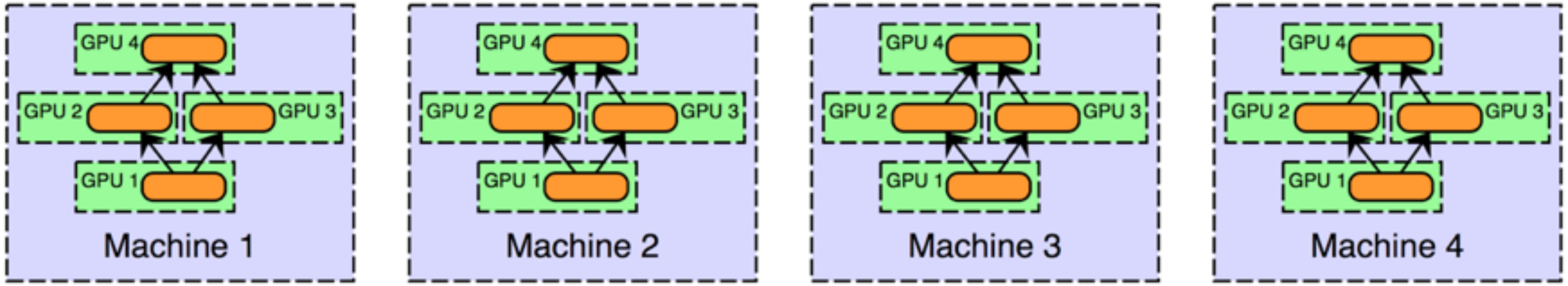
Data Parallelism



Model Parallelism

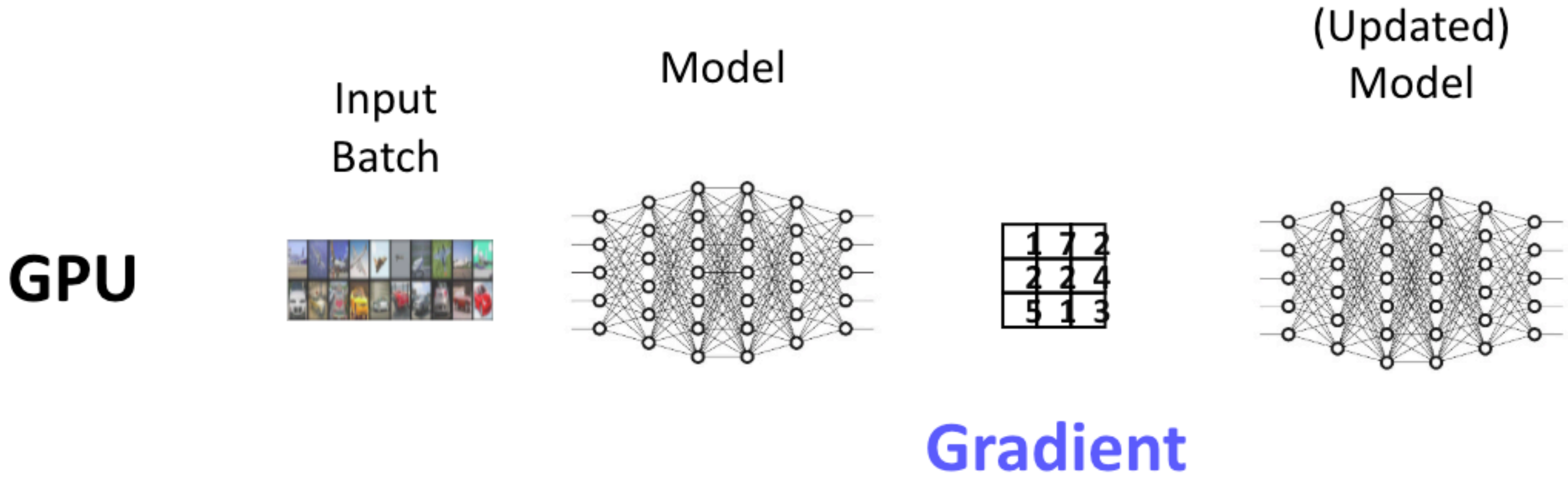


Hybrid (Model and Data) Parallelism



Sequential Training

- At each training iteration:
  - Take batch size *b* training samples from dataset
  - Run a forward pass on each sample and compute each sample's loss
  - Run a backward pass and calculate the gradient
  - Update parameters via gradient



Loss Curve

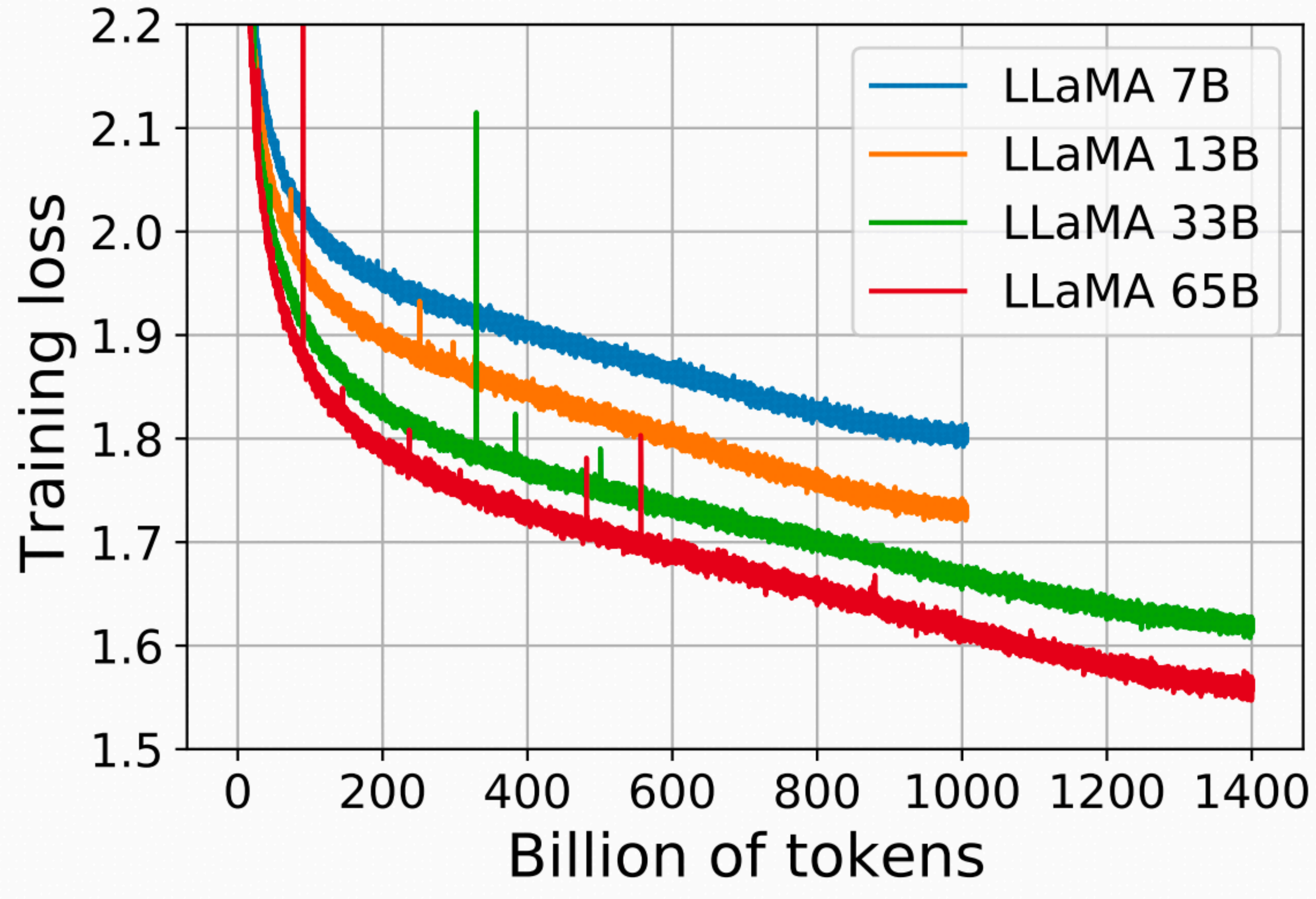


Figure 1: **Training loss over train tokens for the 7B, 13B, 33B, and 65 models.** LLaMA-33B and LLaMA-65B were trained on 1.4T tokens. The smaller models were trained on 1.0T tokens. All models are trained with a batch size of 4M tokens.

LM Performance: More Compute, Better Results

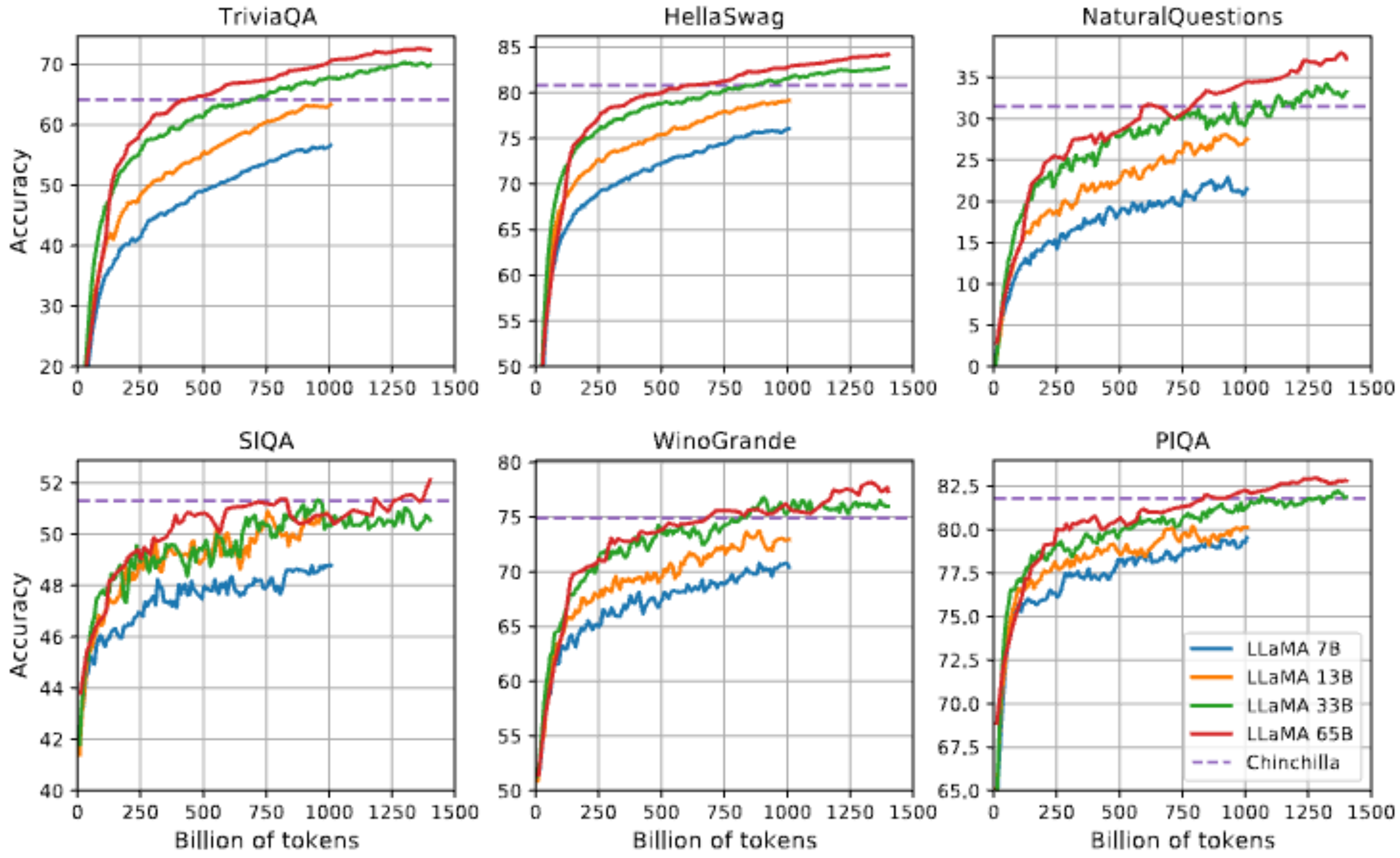


Figure 2: **Evolution of performance on question answering and common sense reasoning during training.**

Llama Family

	Llama				Llama 2				Llama 3			
	Feb 2023				Jul 2023				Apr 2024			
# params	7B	13B	33B	65B	7B	13B	34B	70B	8B			70B
# training tokens	1T	1T	1.4T	1.4T	2T	2T	2T	2T	15T			15T
hidden embed dim	4096	5120	6656	8192	4096	5120		8192	4096			8192
# attn heads	32	40	52	64	32	40		64	32			64
# attn layers	32	40	60	80	32	40		80	32			80
attention	MHA	MHA	MHA	MHA	MHA	MHA	GQA	GQA	GQA			GQA
# kv heads	32	40	52	64	32	40		8	8			8
mlp intermediate size	11008	13824	17920	22016	11008	13824		28672	14336			28672
context	2048				4096				8192			
tokenizer	BPE sentencepiece				BPE sentencepiece				BPE tiktoken			
token vocabulary	32000				32000				128256			
fine-tuned models	-				Llama-2-Chat (Jul 2023) Code Llama (Aug 2023)				Llama-3-Instruct (Apr 2024)			

BPE: Byte Pair Encoding

■ Not released by Meta

MHA: Multi-Head Attention

GQA: Grouped-Query Attention

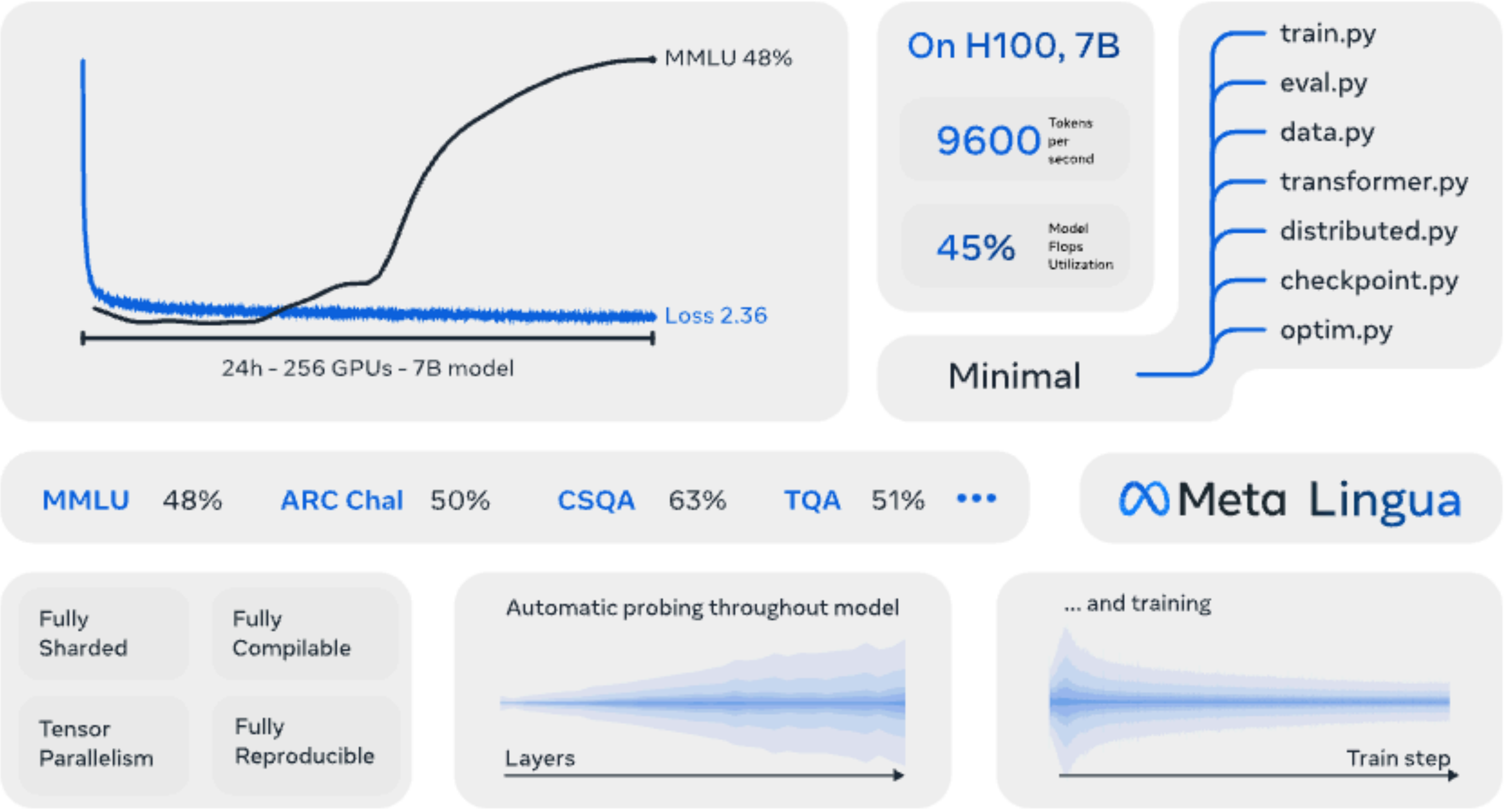
Try Training Llama By Yourself

github.com/facebookresearch/lingua

Meta Lingua

Mathurin Videau*, **Badr Youbi Idrissi***, Daniel Haziza, Luca Wehrstedt, Jade Copet, Olivier Teytaud, David Lopez-Paz. ***Equal and main contribution**

Meta Lingua is a minimal and fast LLM training and inference library designed for research. Meta Lingua uses easy-to-modify PyTorch components in order to try new architectures, losses, data, etc. We aim for this code to enable end to end training, inference and evaluation as well as provide tools to better understand speed and stability. While Meta Lingua is currently under development, we provide you with multiple `apps` to showcase how to use this codebase.



Outline

- Overview of LM Pretraining
- Pretraining Data
- Training Setups
- Scaling Law

Pretraining and Compute

- Goal: get a better pretrained model by "adding more compute"

  - *"The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin."*

    \- *The Bitter Lesson*, Richard Sutton 2019

What Is Compute?

α x Model size x Training data = Training compute

	Model size (# parameters)	Training data (# tokens)	Training compute (FLOPs)	Resources

$$C \approx 6ND$$

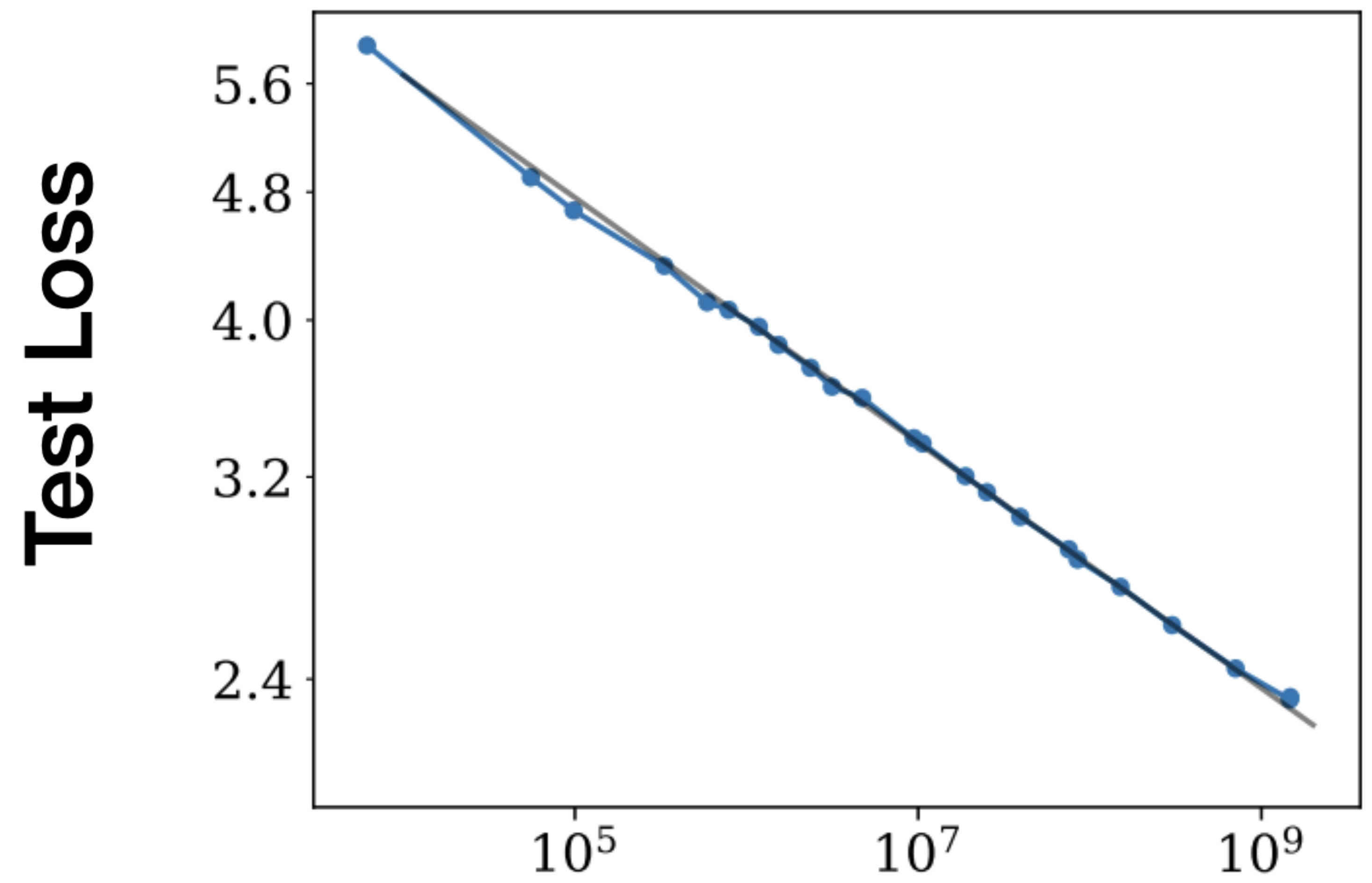
N: number of model parameters
D: number of tokens
C: compute; floating point operations (FLOPs)

What Is Compute?

α x Model size x Training data = Training compute

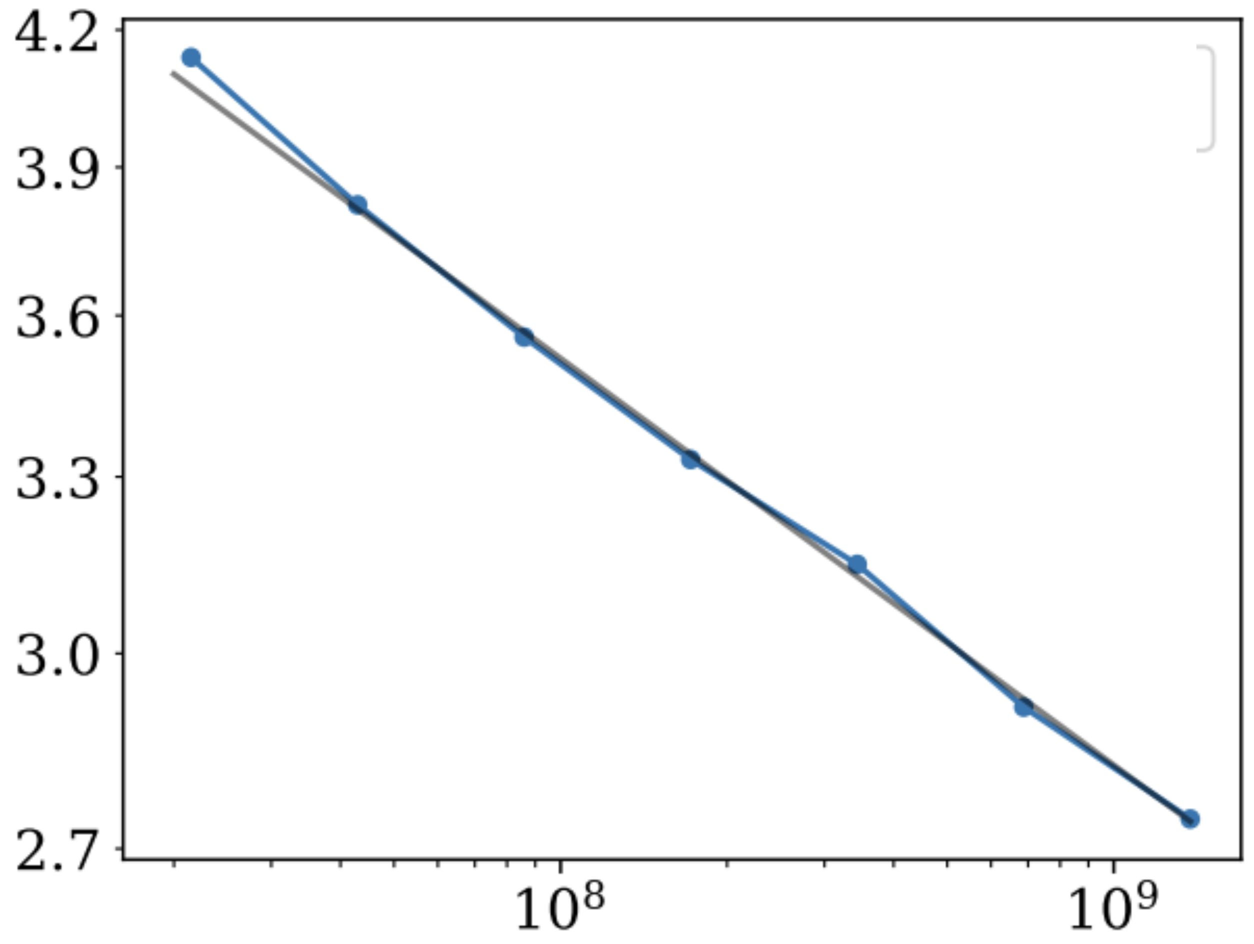
	Model size (# parameters)	Training data (# tokens)	Training compute (FLOPs)	Resources
BERT-base (2018)	109M	250B	1.6e20	64 TPU v2 for 4 days (16 V100 GPU for 33 hrs)
GPT-3 (2020)	175B	300B	3.1e23	~1,000x BERT-base
PaLM (2022)	540B	780B	2.5e24	6k TPU v4 for 2 months

How important is scaling? **(Return)**



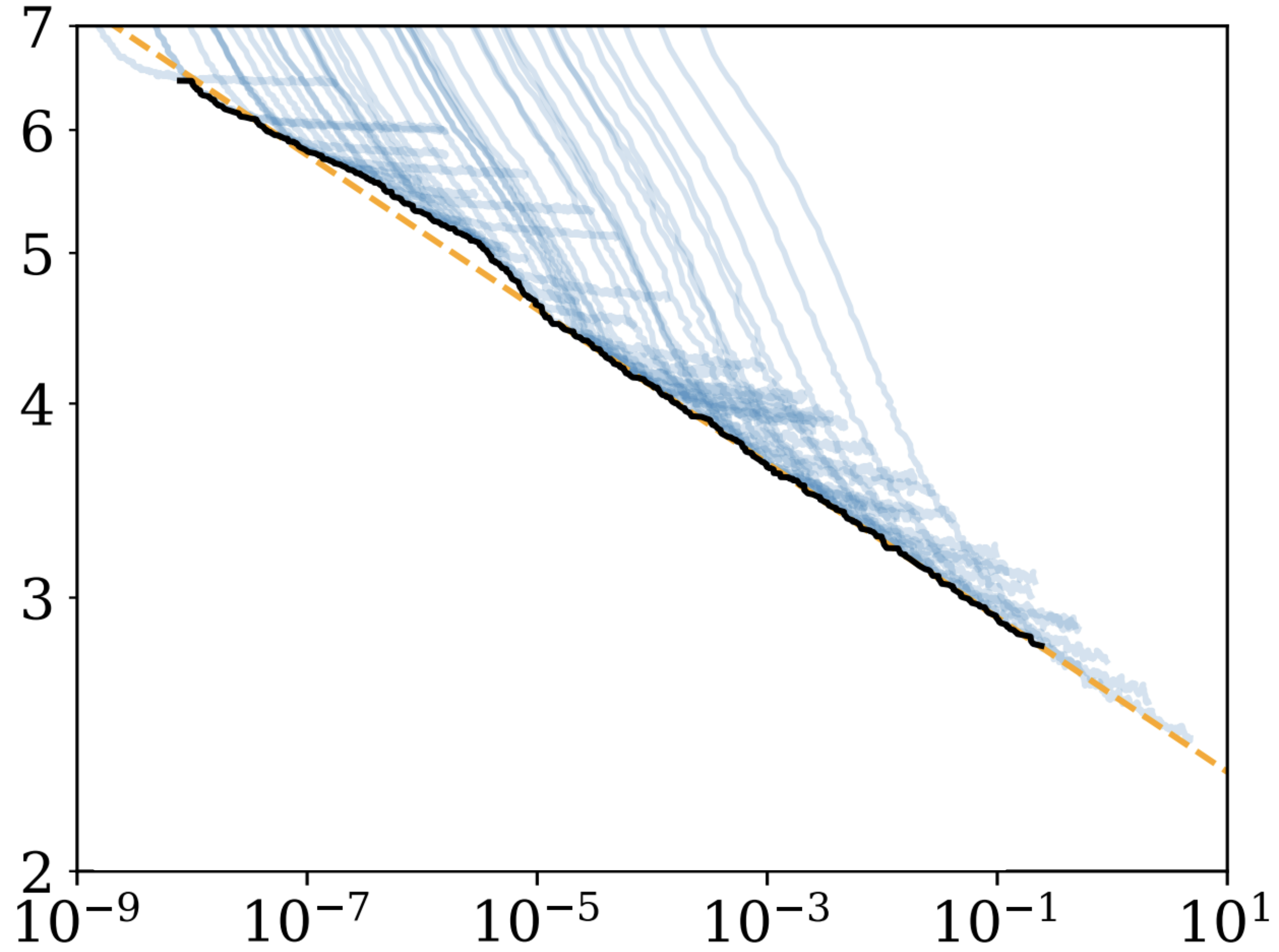
Model size

🤖



Training data

📚



Compute

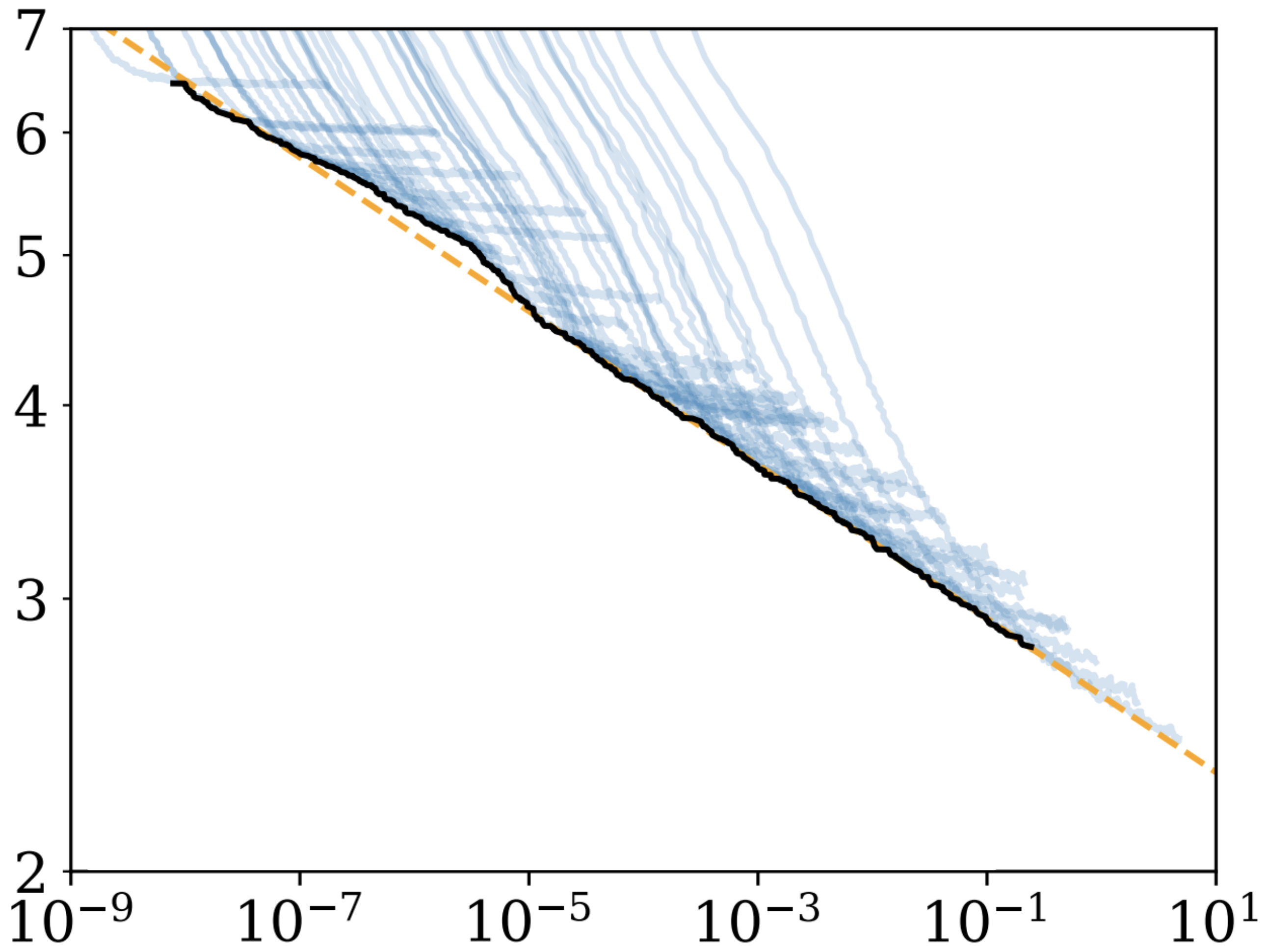
💻

*Test loss predictably **improves** with more **compute***

[Scaling laws for neural language models (2020)]

Scaling Laws

Given a fixed compute budget, can we predict the optimal test loss?



Compute

Train models of different sizes and numbers of tokens

[Scaling laws for neural language models (2020)]

Predictive formula

🤖 📚

We can estimate loss (L) given **model size (N),** training data (D), and learned constants:

$$L(N, D) = \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + E$$

Fitting the constants, yields: $\alpha \approx \beta$

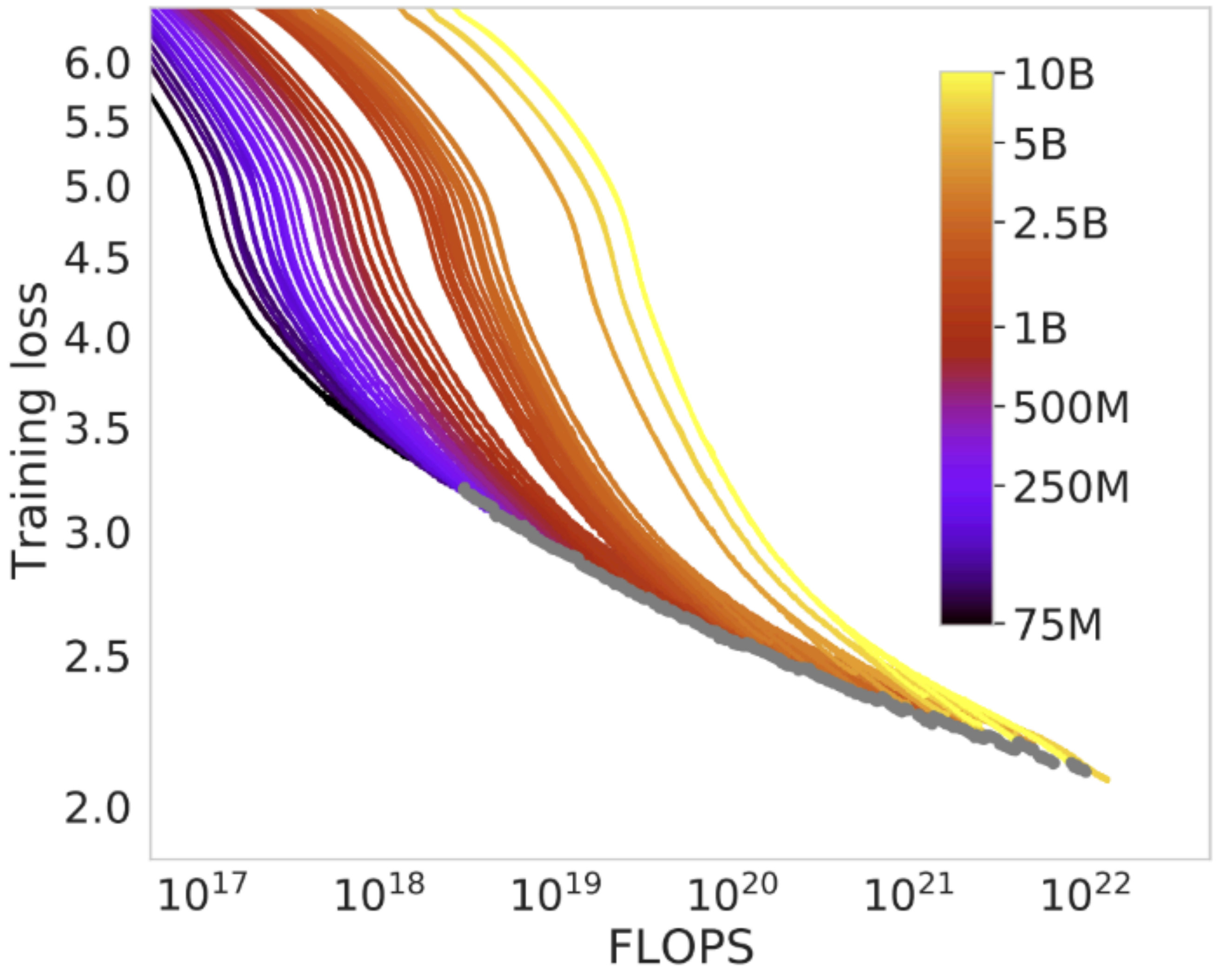
i.e. equal scaling of **N** and **D**.

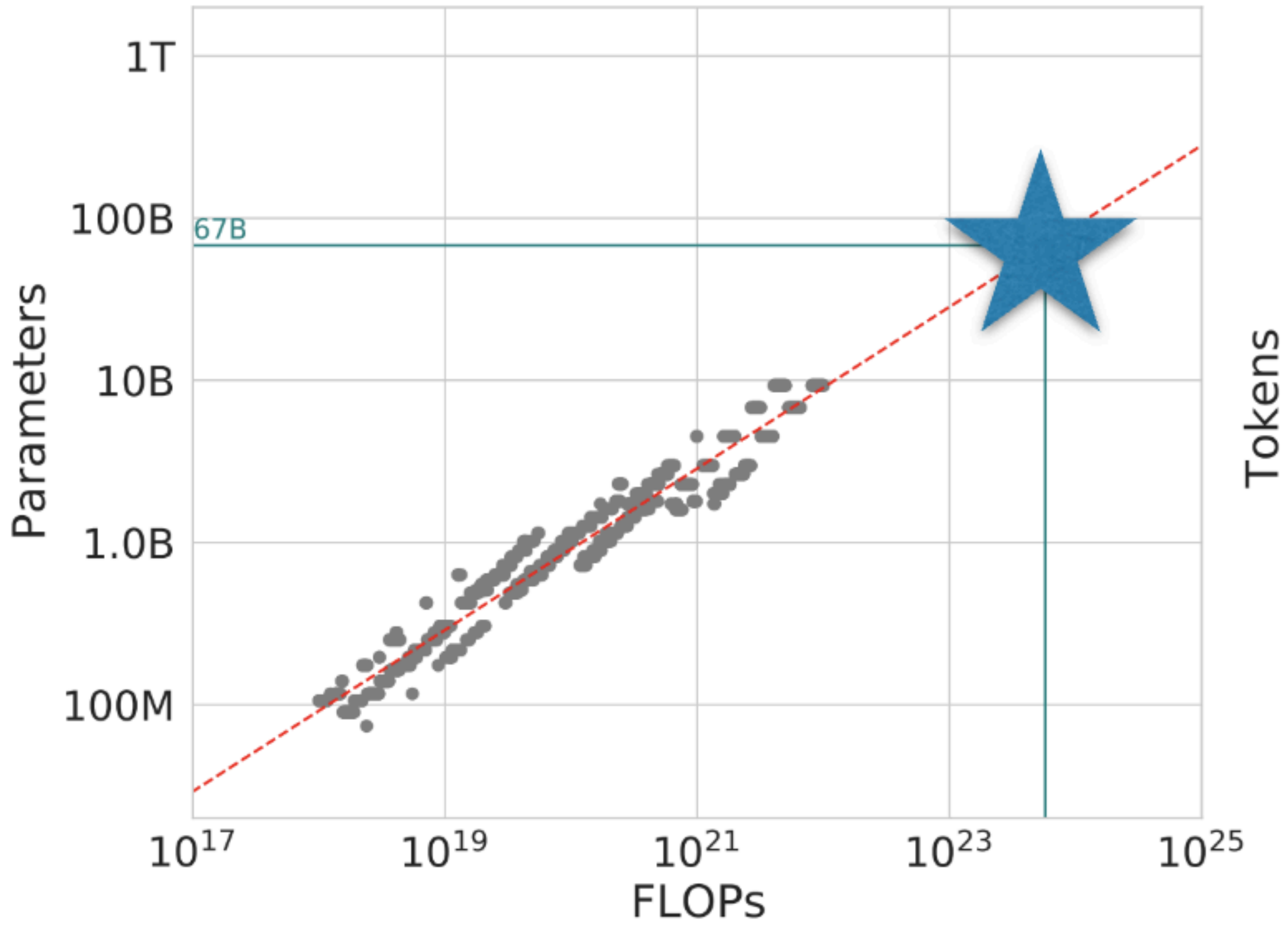
Training compute-optimal large language models (2022)

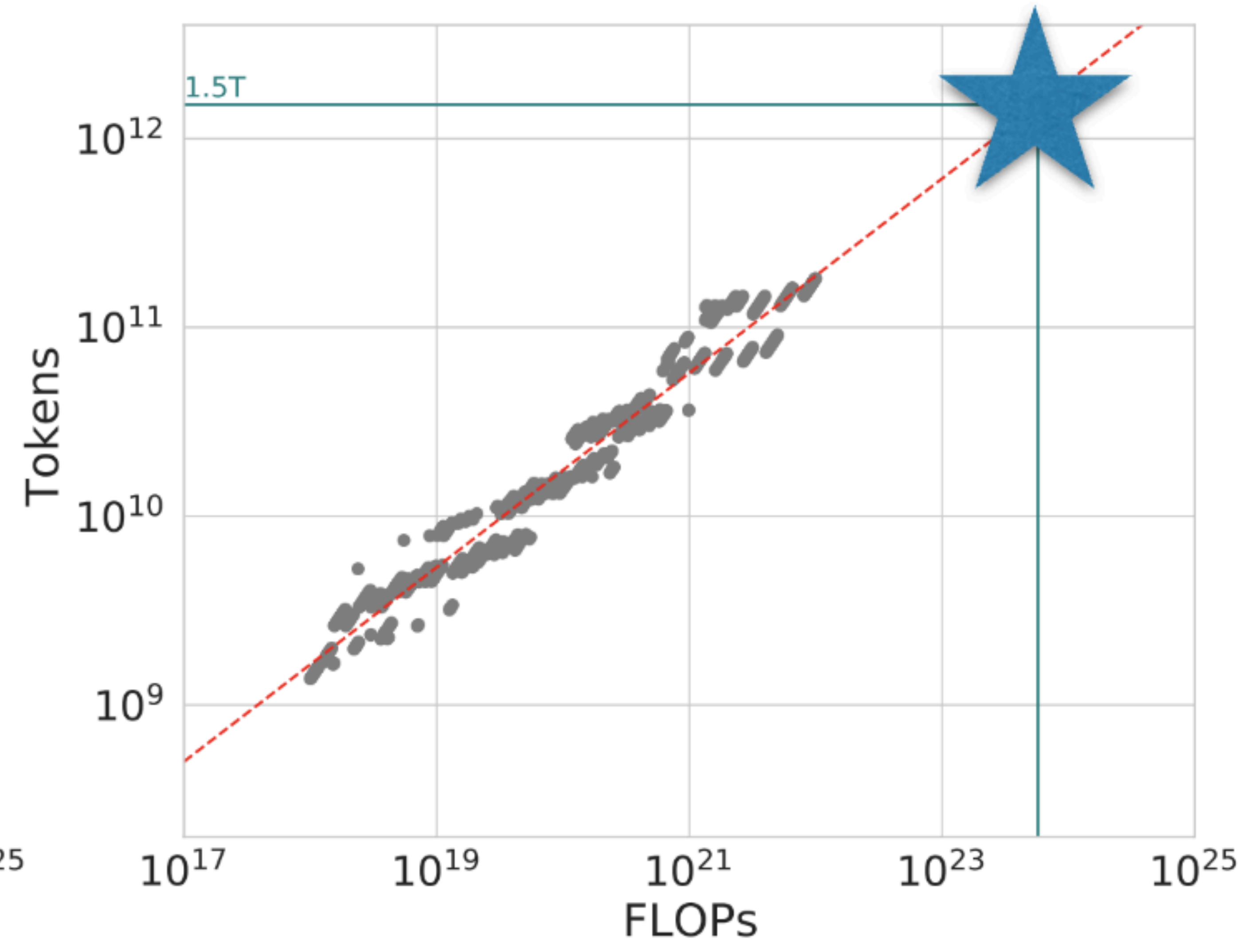
Using Scaling Laws

- Scaling laws are also used to choose hyper parameters
- Basic idea:
  - Run many experiments at a small scale
  - Use a scaling law to estimate the best hyper parameter for a large-scale model / training run

Example: choose model size and # of tokens







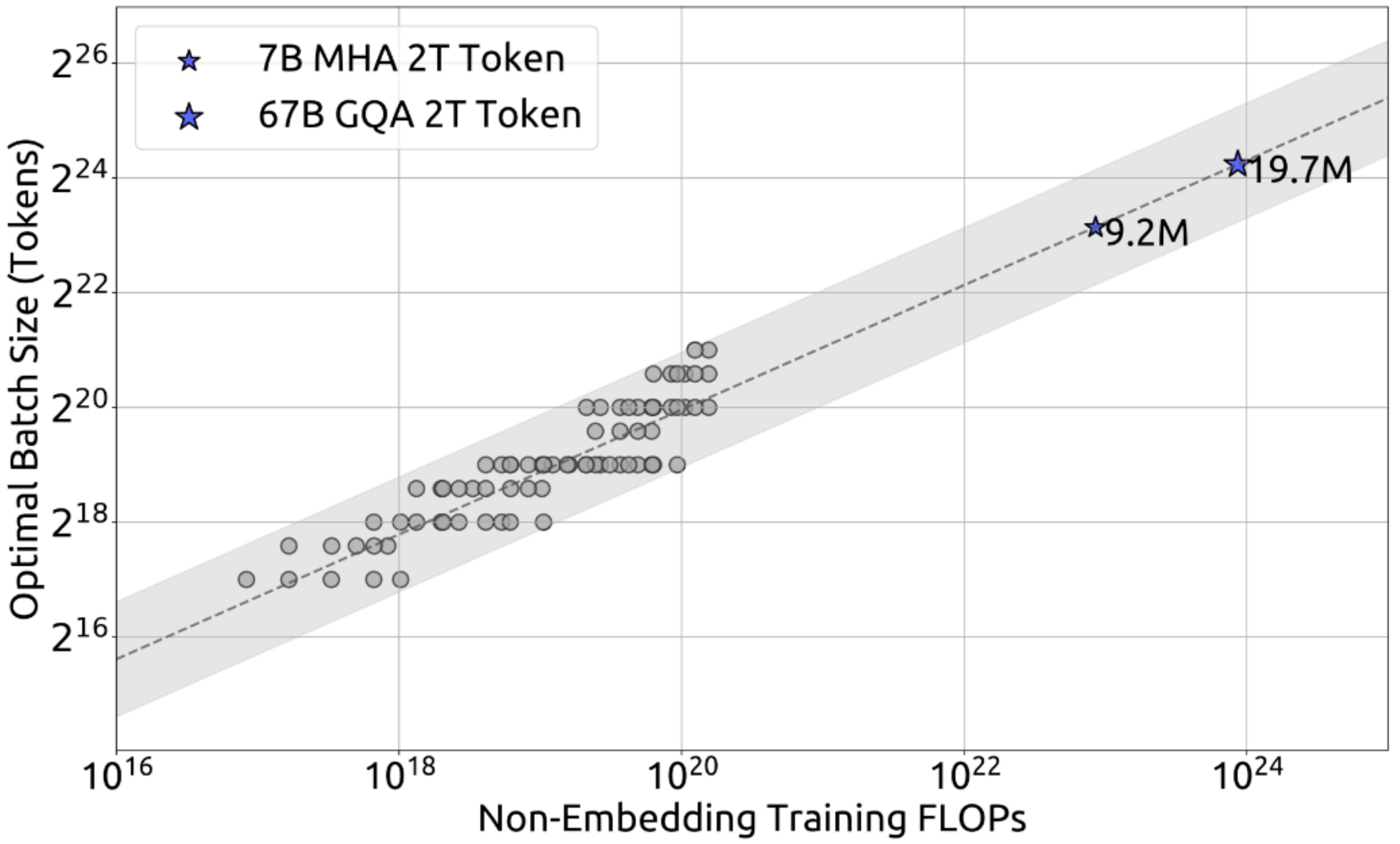
Run experiments

Fit a line and predict optimal model size

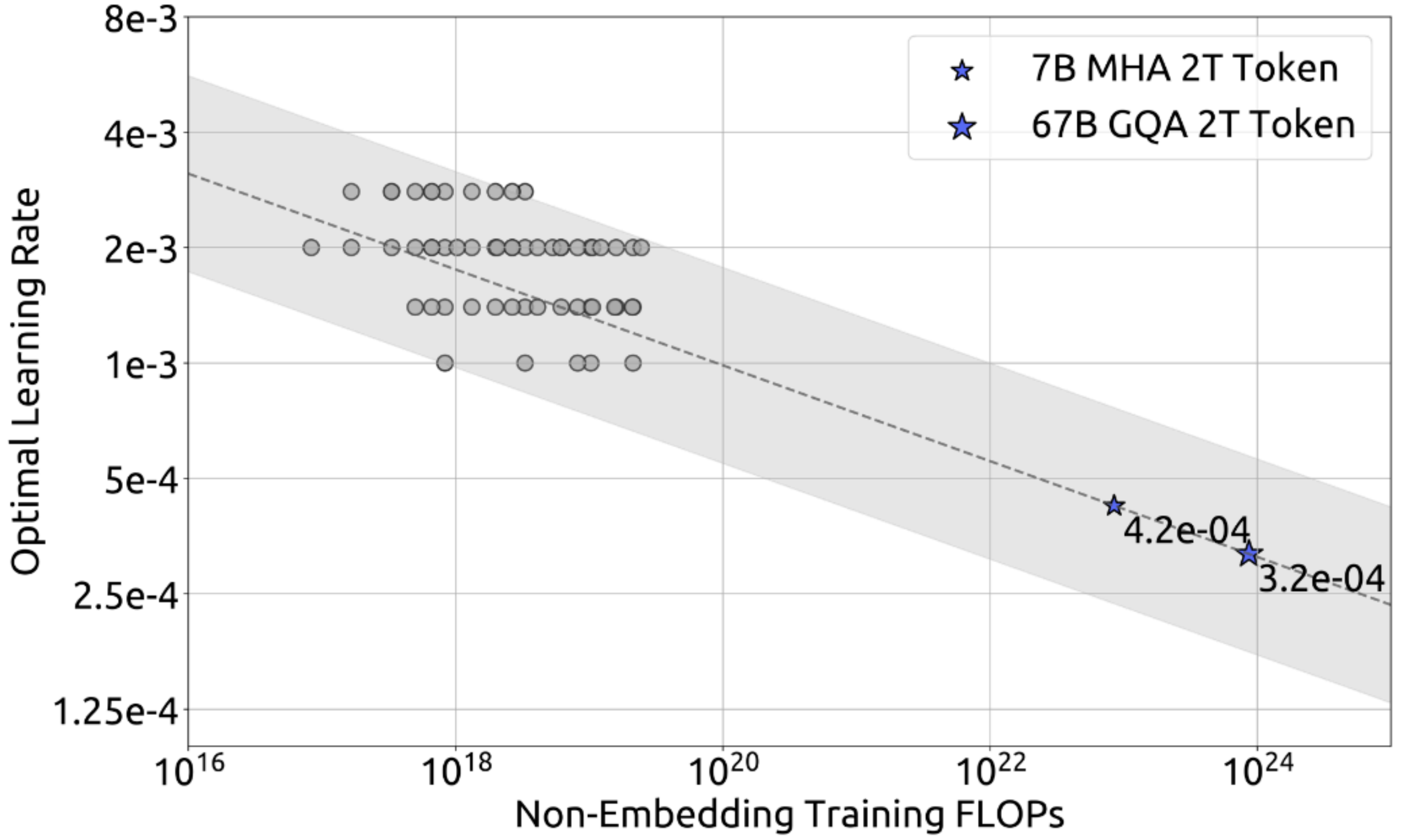
Fit a line and predict optimal # of tokens

"Optimal": best loss for a given compute budget (FLOPs)

Example: choose batch size, learning rate



Optimal batch size



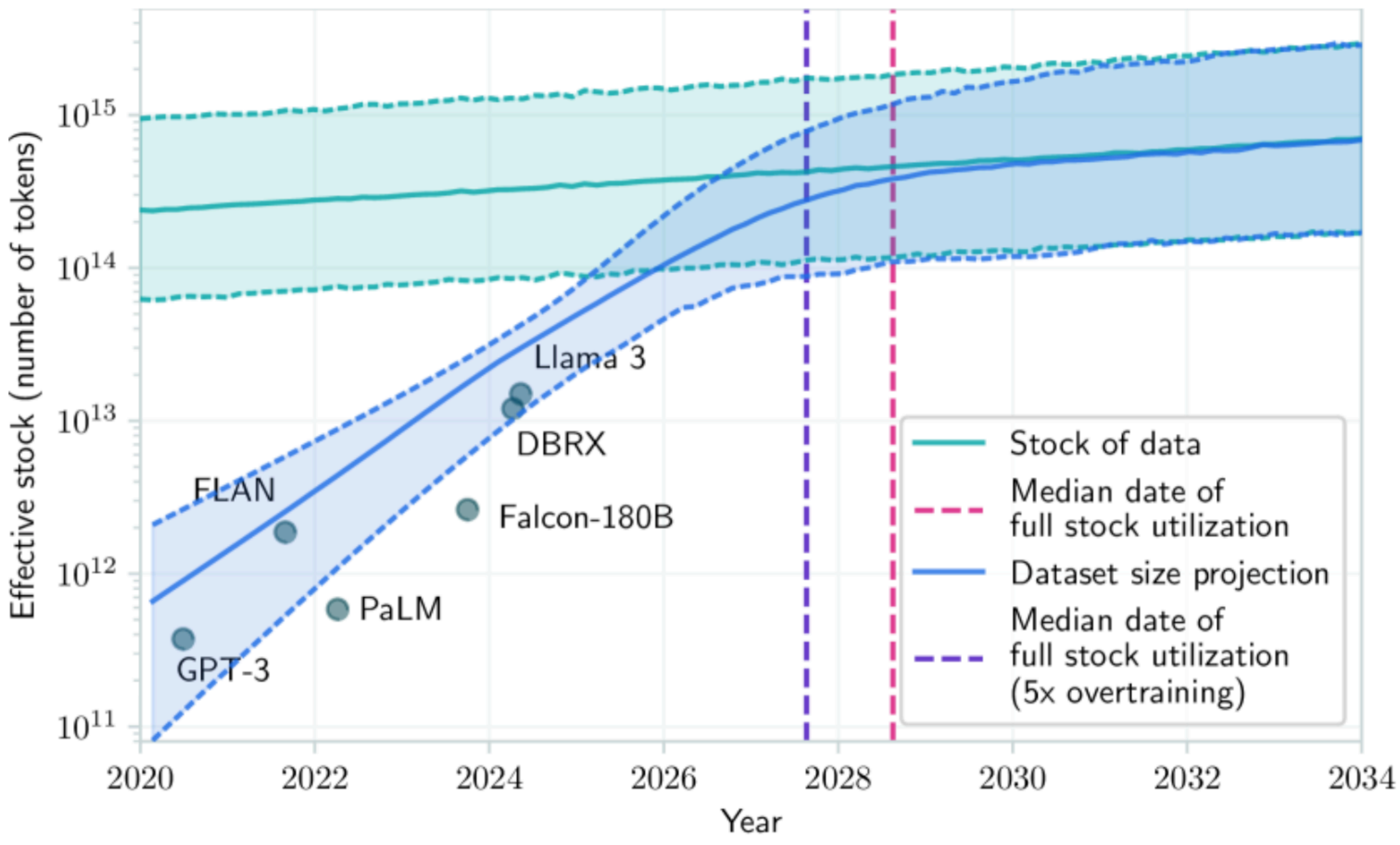
Optimal learning rate

Limits of Scaling

Limits of Scaling

- **Limits on data**: Modern LLMs are trained on basically the *entire internet* - we can't find 10 new internets out of nowhere
- **Limits on compute**: Big tech companies can't continue to 10x their model sizes for much longer

Limits on Data: Data Is Running Out



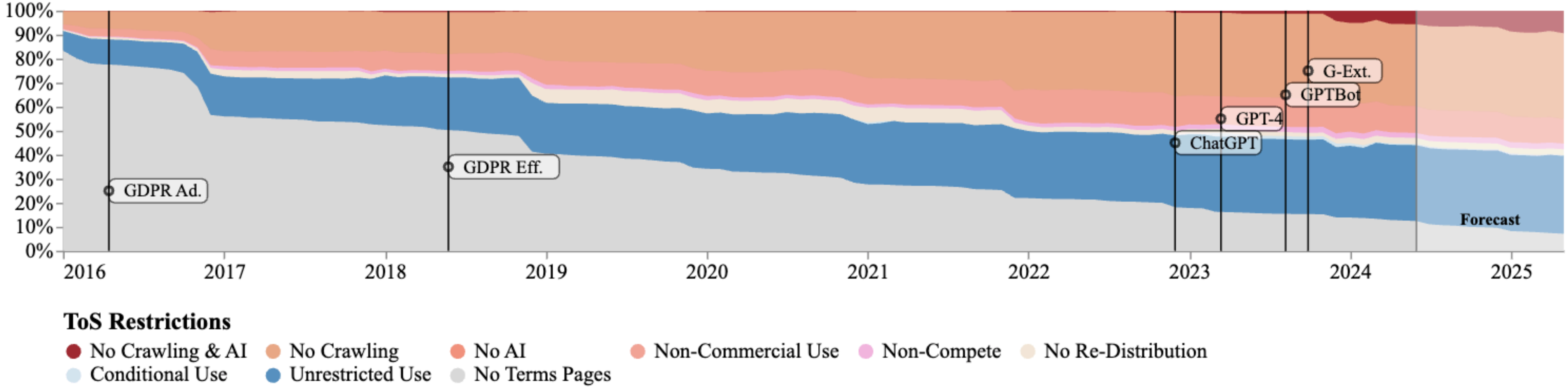
Limits on Data: Restrictions in Use

Public data is always usable, but proprietary/licensed data is not

Public Data

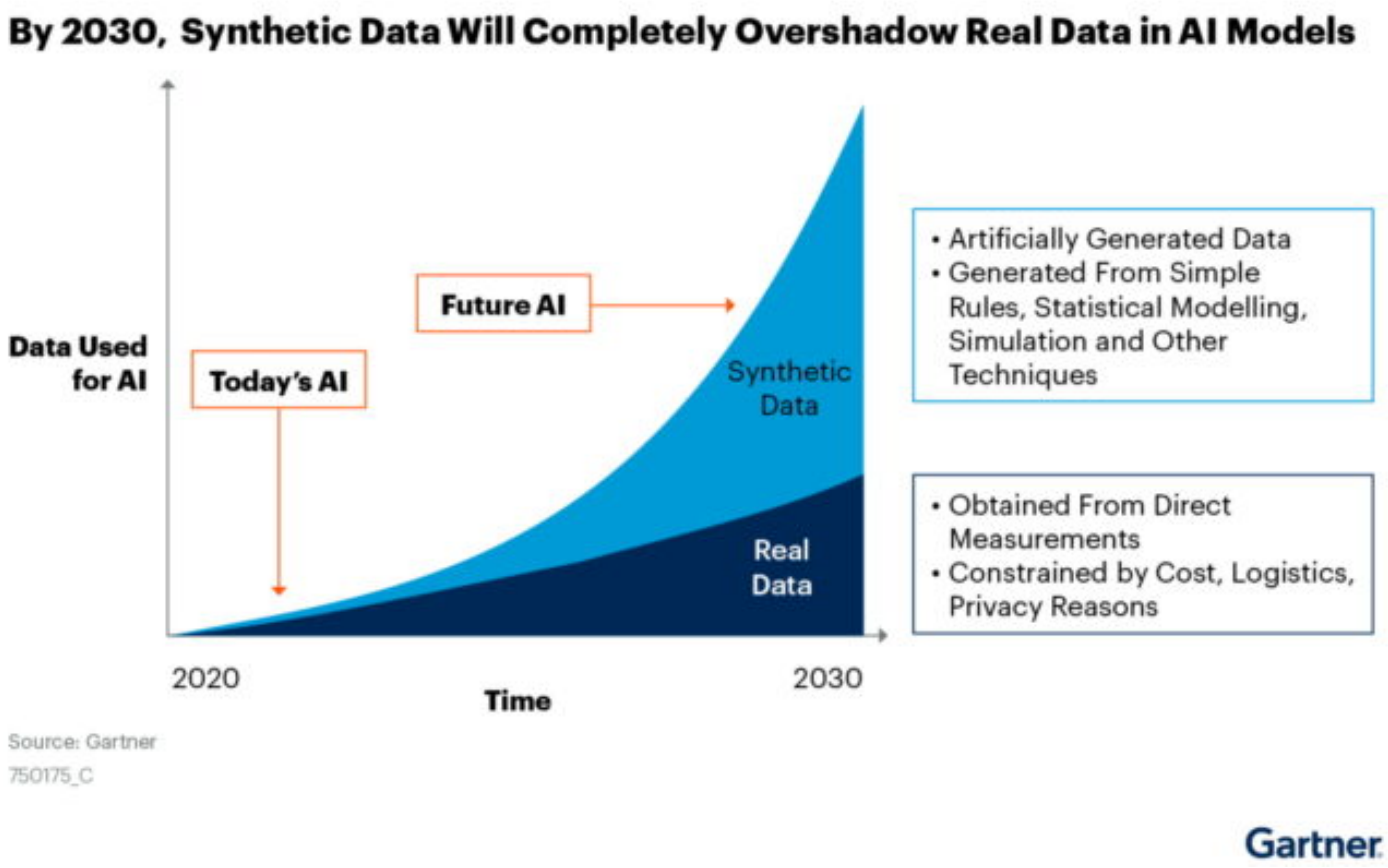
Proprietary Data

Limits on Data: Restrictions in Use



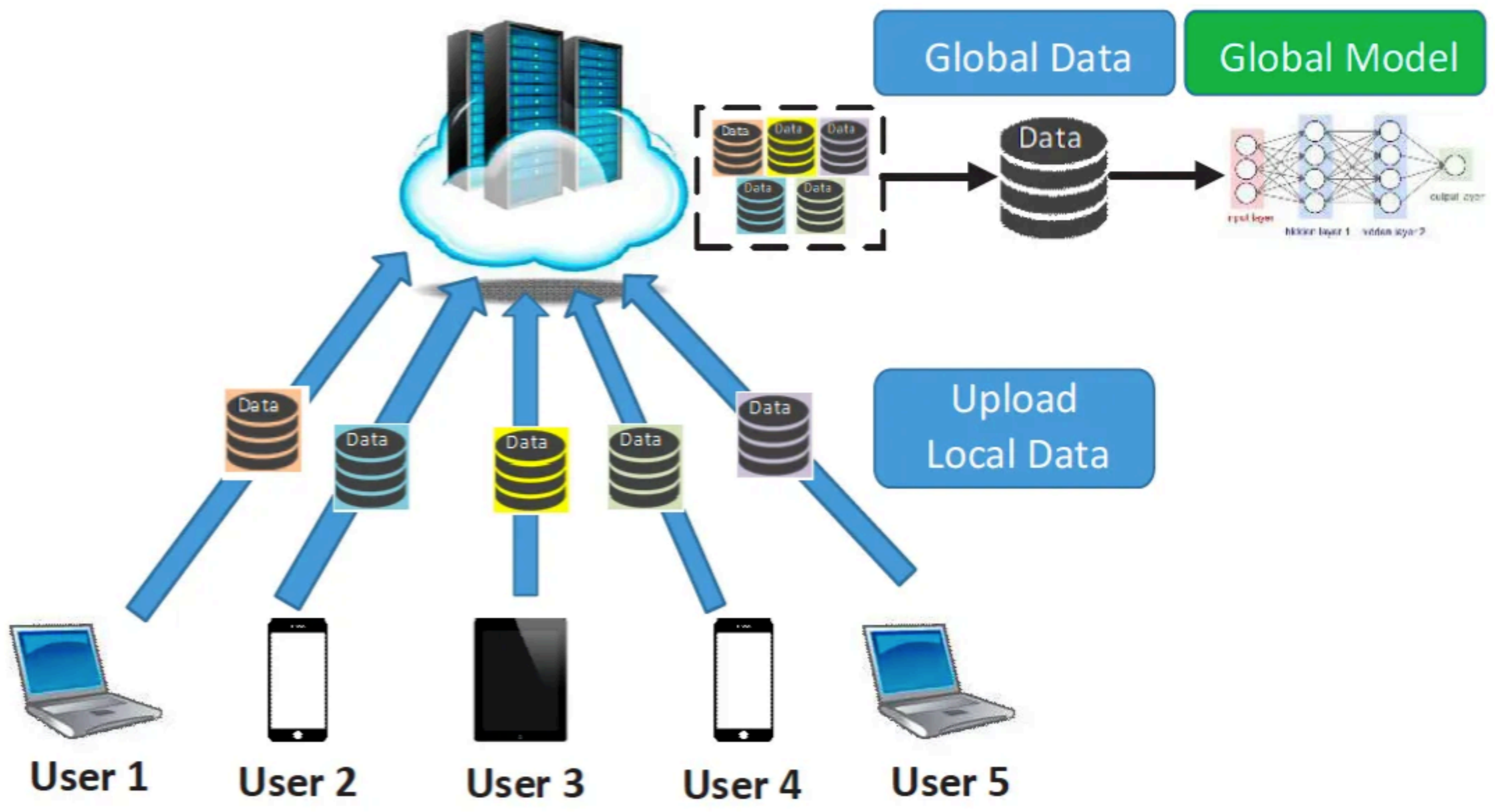
Terms of Service pages have imposed more anti-crawling and now anti-AI restrictions

~~Limits on Data:~~ Synthetic Data



Limits on Compute: Pretraining is Centralized

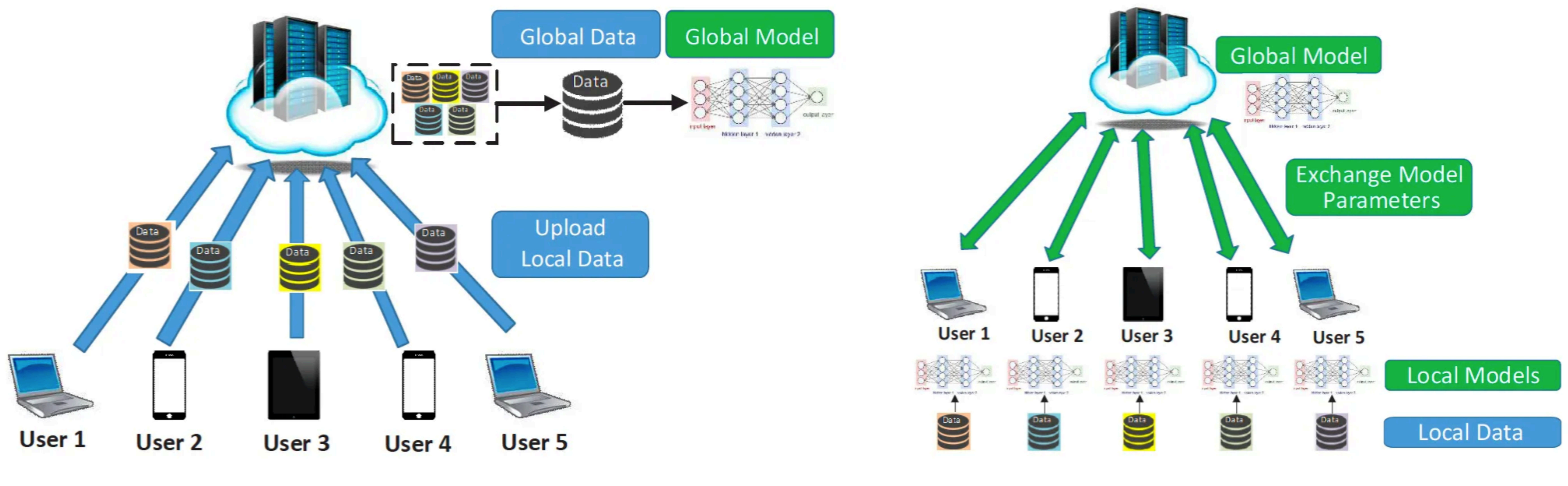
- Current pretraining requires GPUs' communications
- But one data center can hold fixed amount of GPUs



(a) centralized learning (can be outsourced learning).

~~Limits on Compute~~ : Decentralized Training

- Current pretraining requires GPUs' communications
- But one data center can hold fixed amount of GPUs



(a) centralized learning (can be outsourced learning).

(b) distributed learning

Questions?