

CS11-711 Advanced NLP
Pre-training and
Pre-trained LLMs

Xiang Yue



Carnegie Mellon University

Language Technologies Institute

<https://phontron.com/class/anlp-fall2024>

Introduction



Postdoc @ LTI, CMU

Advisor: Prof. Graham Neubig

<https://xiangyue9607.github.io/>

Office Hours

Wednesday 1:30 PM - 2:30 PM

Location

GHC 6416

Research interests:

Natural Language Processing (NLP) and (Multimodal) Large Language Models (LLMs)

- Robust and rigorous evaluations of LLMs
- Understanding and enhancing the reasoning capabilities of LLMs
- Applying LLMs to solve real-world problems (code generation, agents, healthcare, etc.)

Recap of Language Modeling

Probabilistic Language Models

$$P(X)$$



Sentence/Document

A generative model that calculates the probability of language

Auto-regressive Language Models

$$P(X) = \prod_{i=1}^I P(x_i \mid x_1, \dots, x_{i-1})$$

Next Token Context

Next Token Prediction

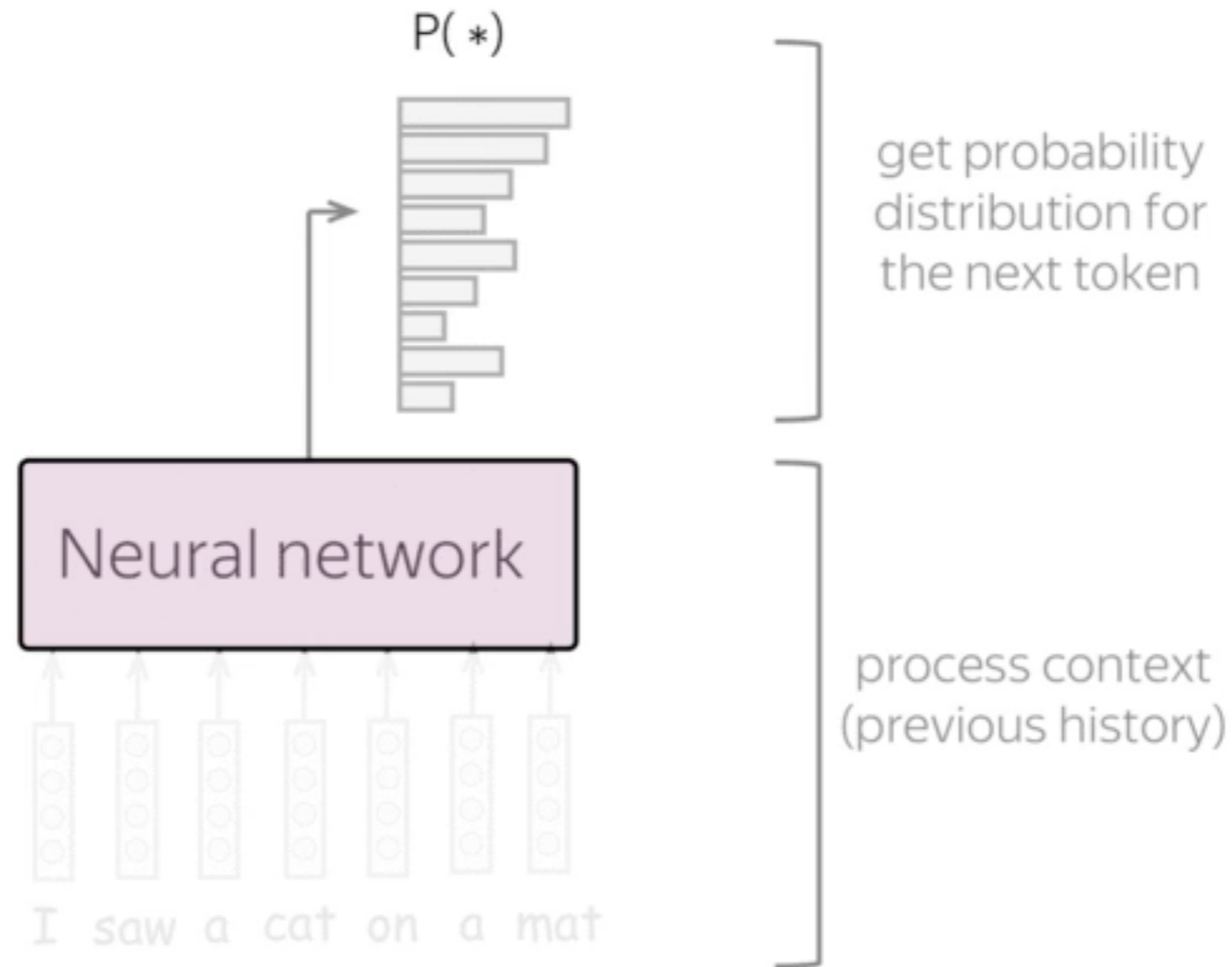


Image Credit: https://lena-voita.github.io/nlp_course/language_modeling.html

- This is classification! We can think of neural language models as neural classifiers. They classify prefix of a text into $|V|$ classes, where the classes are vocabulary tokens.

Next Token Prediction

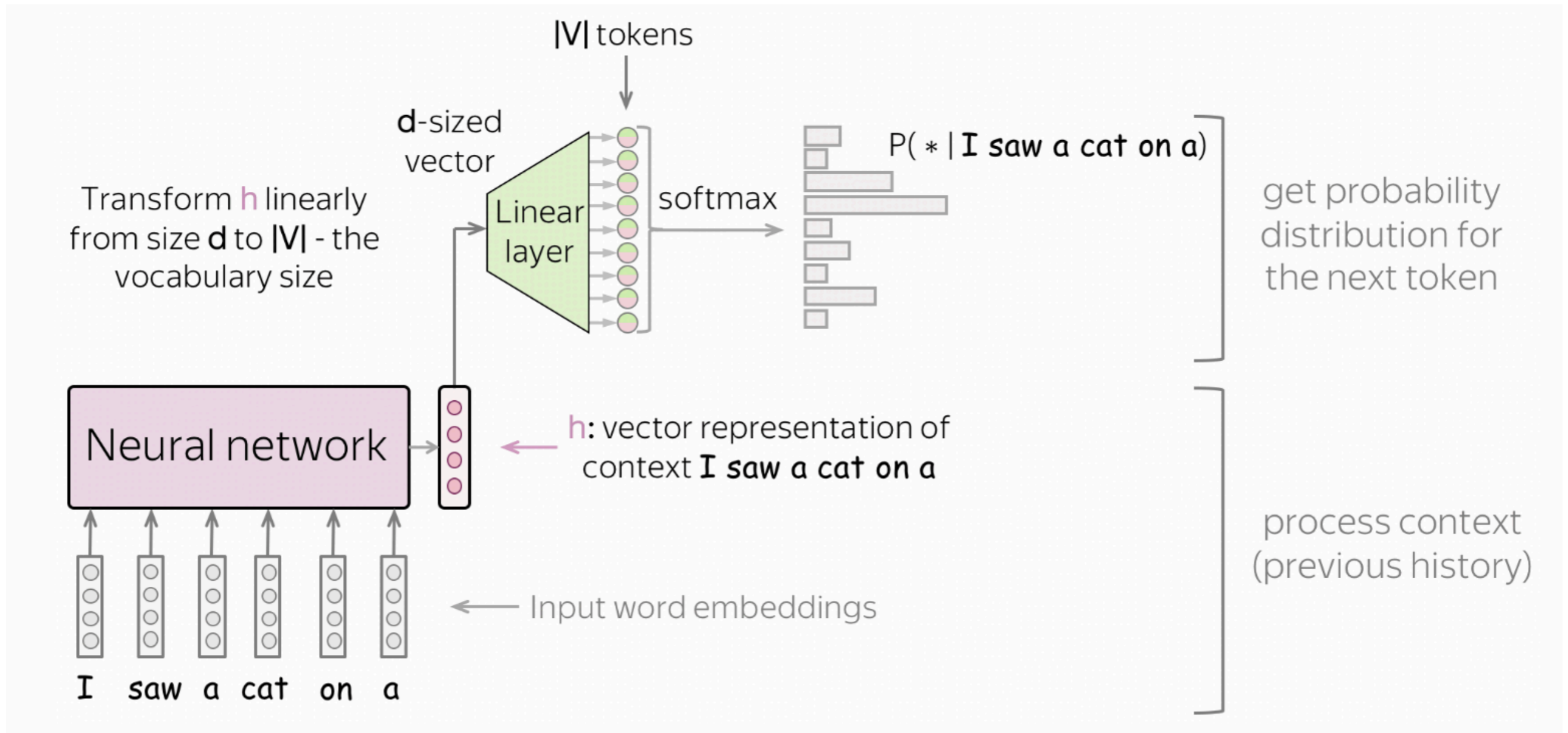


Image Credit: https://lena-voita.github.io/nlp_course/language_modeling.html

- feed word embedding for previous (context) words into a network;
- get vector representation of context from the network;
- from this vector representation, predict a probability distribution for the next token.

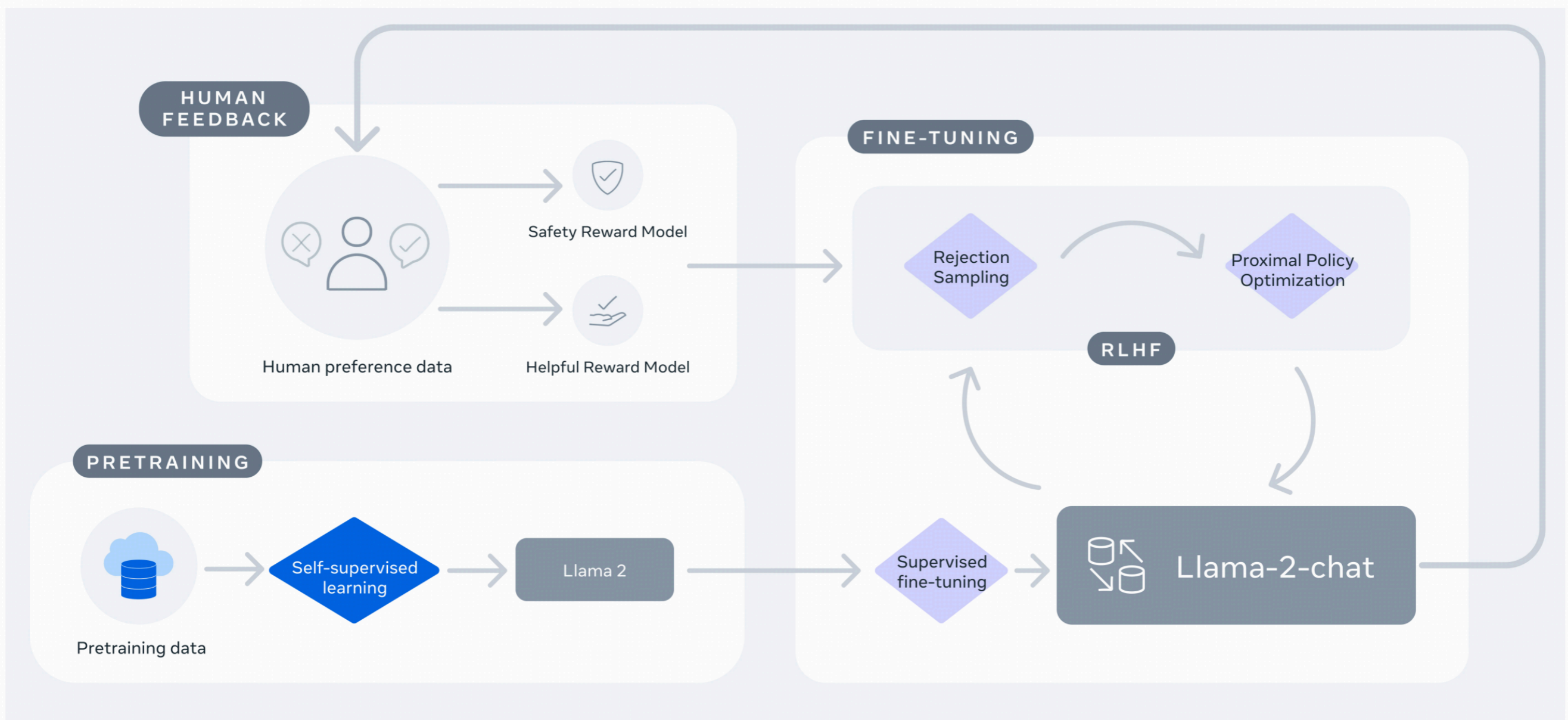
What we will cover in this class

- Overview of LLMs training
- Pretraining Data
- Training Setups
- Open vs Closed Models
- Representative LLMs

Training LLMs - Llama as an example



Overview of LLMs Training



Pre-training -> Supervised Fine-tuning (SFT) -> RLHF

Pre-training -> Post-training

Pre-training -> Mid-training -> Post-training

Why is it called pre-training?

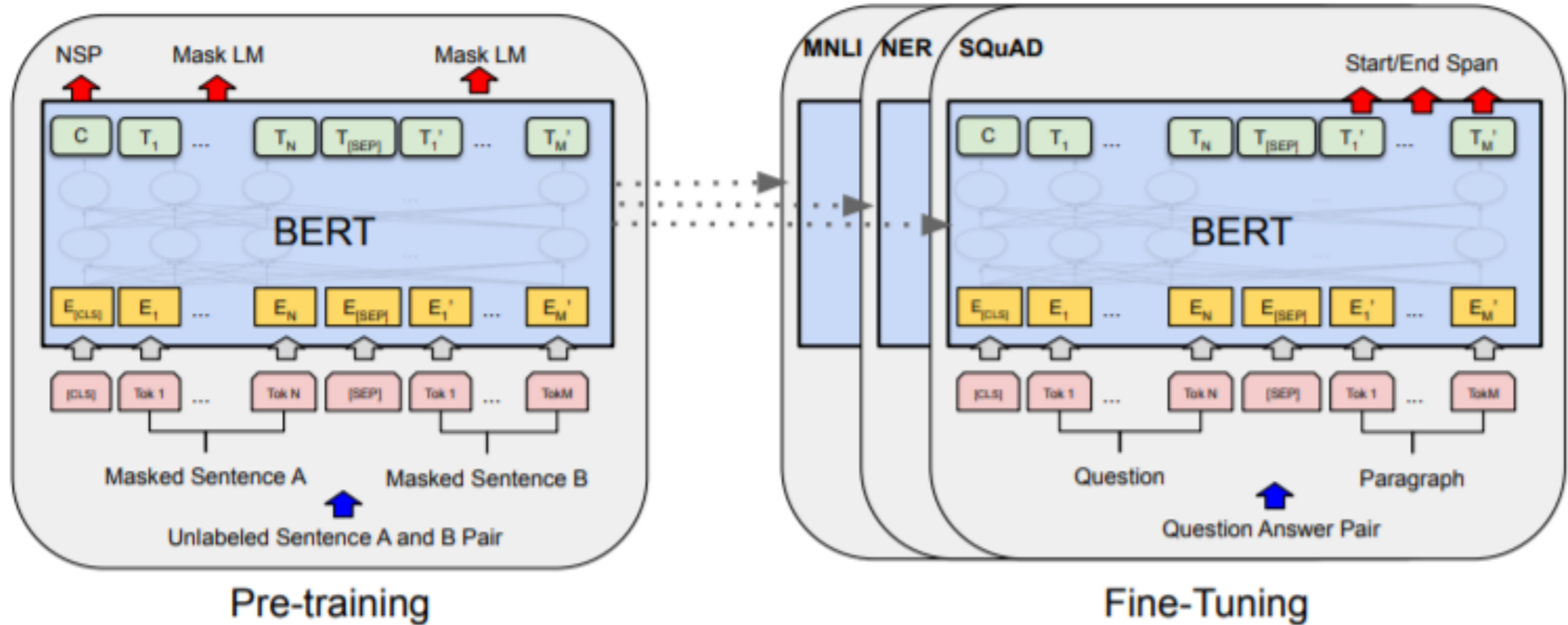


Image Credit: BERT Paper

“Pre-”training happens before training (fine-tuning)!

Pre-training Data

Common
Crawl

Colossal Clean
Crawled Corpus (C4)



arXiv

StackExchange



Example: Llama1

Pre-training Data Mixture

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

1.4 Trillion Tokens!

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**.

Tokenizer

“We tokenize the data with the bytepair encoding (BPE) algorithm (Sennrich et al., 2015), using the implementation from SentencePiece (Kudo and Richardson, 2018). Notably, we split all numbers into individual digits, and fallback to bytes to decompose unknown UTF-8 characters.” (*—from Llama1 Paper*)

- Incrementally combine together the most frequent token pairs

```
{'l o w </w>': 5, 'l o w e r </w>': 2, 'n e w e s t </w>': 6, 'w i d e s t </w>': 3}
```

```
pairs = get_stats(vocab)
```

```
[(('e', 's'), 9), (('s', 't'), 9), (('t', '</w>'), 9), (('w', 'e'), 8), (('l', 'o'), 7), ...]
```

```
vocab = merge_vocab(pairs[0], vocab)
```

```
{'l o w </w>': 5, 'l o w e r </w>': 2, 'n e w e s t </w>': 6, 'w i d e s t </w>': 3}
```

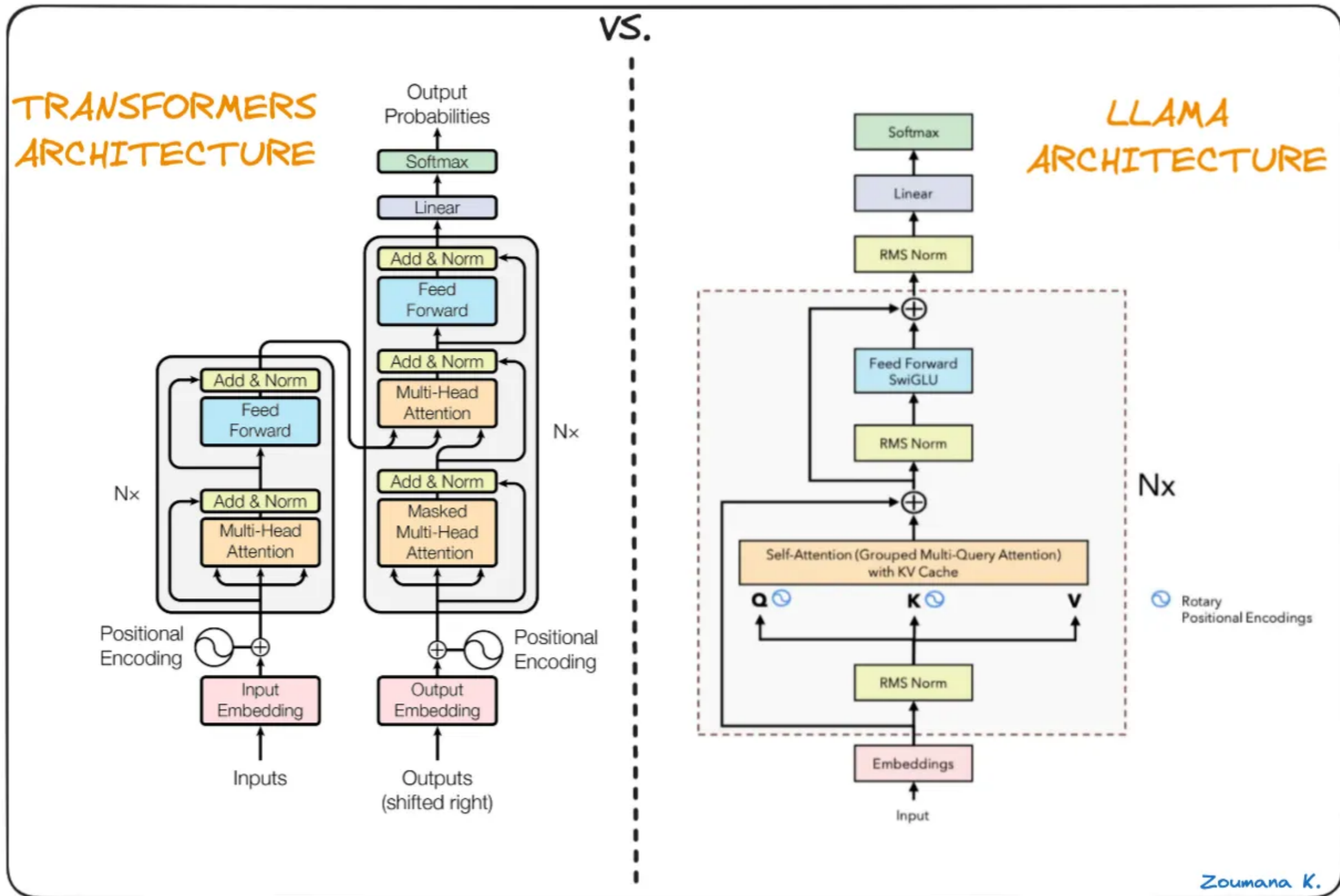
```
pairs = get_stats(vocab)
```

```
[(('es', 't'), 9), (('t', '</w>'), 9), (('l', 'o'), 7), (('o', 'w'), 7), (('n', 'e'), 6)]
```

```
vocab = merge_vocab(pairs[0], vocab)
```

```
{'l o w </w>': 5, 'l o w e r </w>': 2, 'n e w e s t </w>': 6, 'w i d e s t </w>': 3}
```

Architecture (Recap)



Grouped Query Attention (GQA)

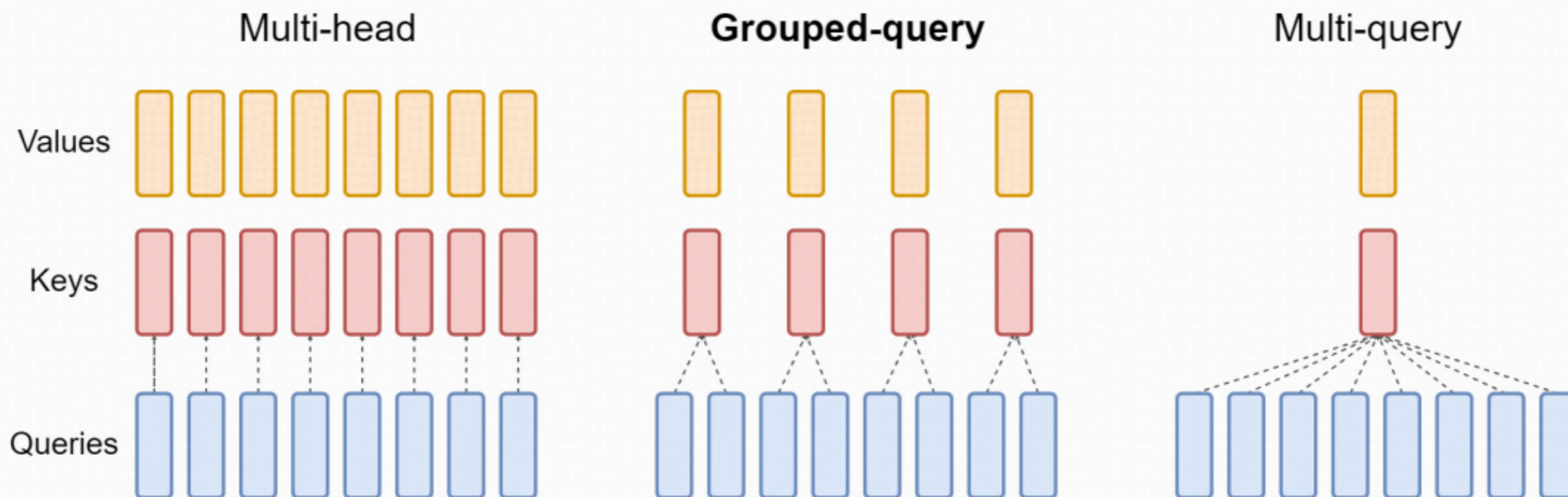


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between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

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.**

<https://arxiv.org/pdf/2302.13971>

Optimizer: AdamW (β_1 : 0.9, β_2 : 0.95)

Learning Rate Schedule: Cosine schedule

Final learning rate: 10% of the maximal learning rate

Weight Decay: 0.1

Gradient Clipping: 1.0

Warmup Steps: 2,000 steps

How to train the model
with a GPU cluster or
multiple GPU nodes?

Training Library

- DeepSpeed is a deep learning optimization library that makes distributed training easy, efficient, and effective. It has been integrated into the Huggingface library.
- Megatron-LM is a large, powerful transformer model framework developed by the Applied Deep Learning Research team at NVIDIA.

Parallelism

- 1. DataParallel (DP)** - the same setup is replicated multiple times, and each being fed a slice of the data. The processing is done in parallel and all setups are synchronized at the end of each training step.
- 2. TensorParallel (TP)** - each tensor is split up into multiple chunks, so instead of having the whole tensor reside on a single GPU, each shard of the tensor resides on its designated GPU. During processing each shard gets processed separately and in parallel on different GPUs and the results are synced at the end of the step. This is what one may call horizontal parallelism, as the splitting happens on a horizontal level.
- 3. PipelineParallel (PP)** - the model is split up vertically (layer-level) across multiple GPUs, so that only one or several layers of the model are placed on a single GPU. Each GPU processes in parallel different stages of the pipeline and works on a small chunk of the batch.
- 4. Zero Redundancy Optimizer (ZeRO)** - also performs sharding of the tensors somewhat similar to TP, except the whole tensor gets reconstructed in time for a forward or backward computation, therefore the model doesn't need to be modified. It also supports various offloading techniques to compensate for limited GPU memory.

Loss Curve

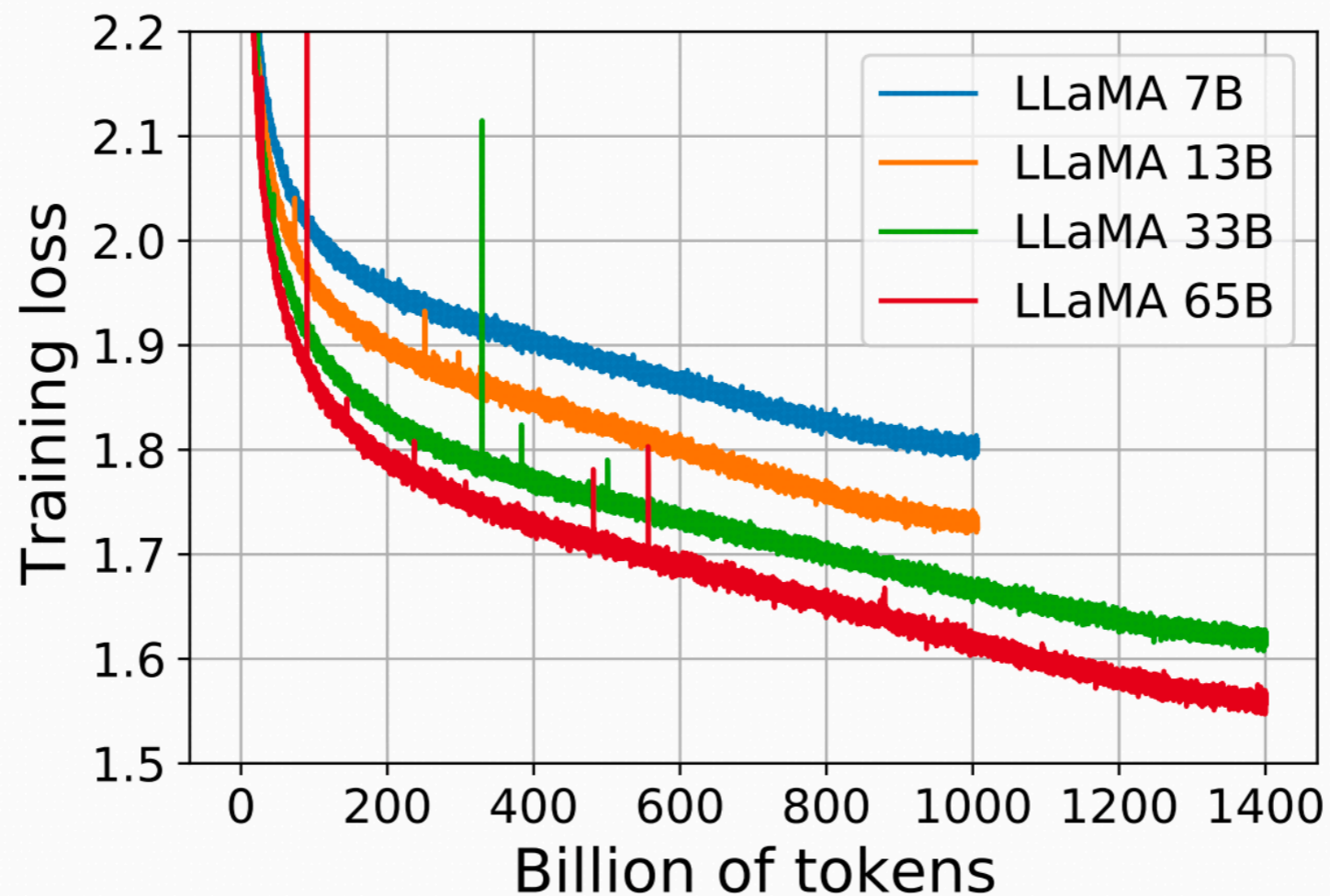


Figure 1: **Training loss over train tokens for the 7B, 13B, 33B, and 65 models.** LLaMA-33B and LLaMA-65B were trained on 1.4T tokens. The smaller models were trained on 1.0T tokens. All models are trained with a batch size of 4M tokens.

Scaling Laws

Scaling Laws

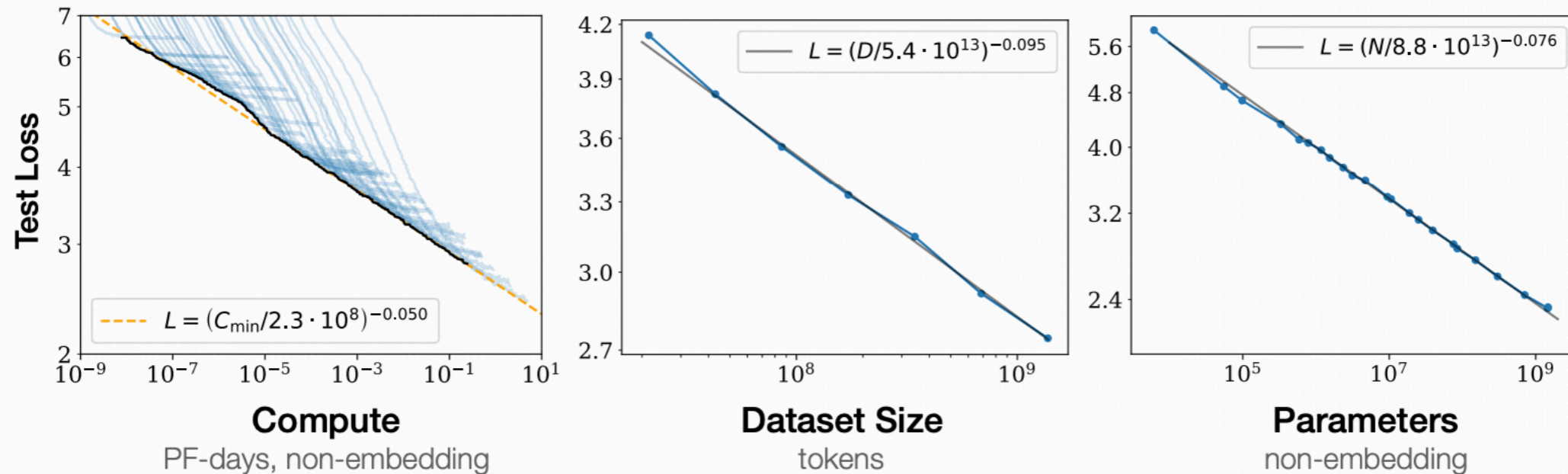


Figure 1 Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute² used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

<https://arxiv.org/pdf/2001.08361>

Given constrained compute budget measured in FLOPs, floating-point operations, what would be the optimal combination of model size and training data size (measured in number of tokens) that yields the lowest loss?

Scaling Laws

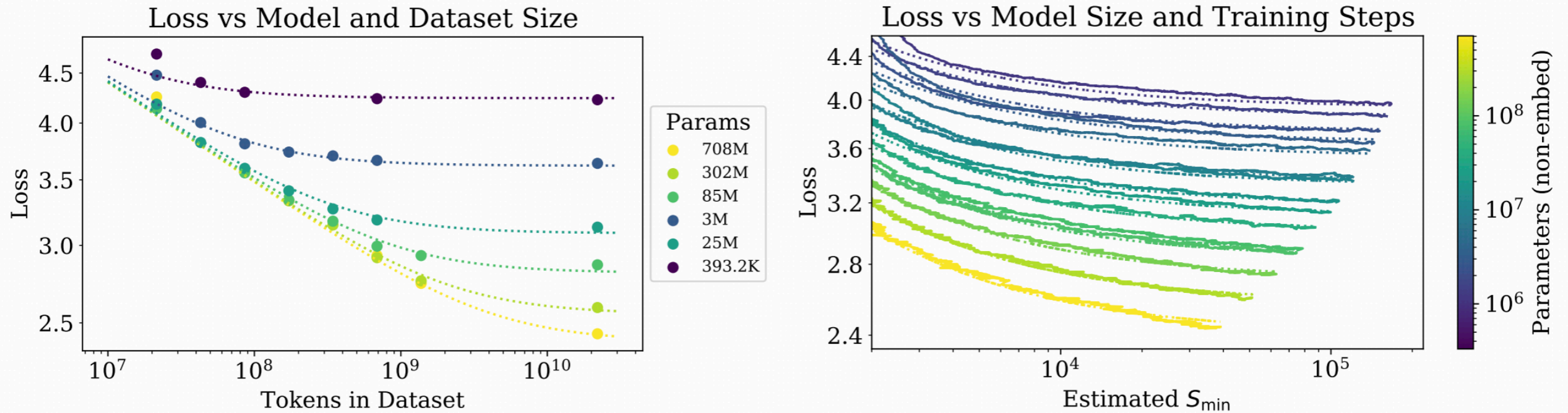


Figure 4 Left: The early-stopped test loss $L(N, D)$ varies predictably with the dataset size D and model size N according to Equation (1.5). **Right:** After an initial transient period, learning curves for all model sizes N can be fit with Equation (1.6), which is parameterized in terms of S_{\min} , the number of steps when training at large batch size (details in Section 5.1).

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$

$$L(N, S) = \left(\frac{N_c}{N} \right)^{\alpha_N} + \left(\frac{S_c}{S_{\min}(S)} \right)^{\alpha_S}$$

Scaling Laws

$$\text{Error} \propto N^{-\alpha} + D^{-\beta} + C^{-\gamma}$$

1.2 Summary of Scaling Laws

The test loss of a Transformer trained to autoregressively model language can be predicted using a power-law when performance is limited by only either the number of non-embedding parameters N , the dataset size D , or the optimally allocated compute budget C_{\min} (see Figure 1):

1. For models with a limited number of parameters, trained to convergence on sufficiently large datasets:

$$L(N) = (N_c/N)^{\alpha_N}; \quad \alpha_N \sim 0.076, \quad N_c \sim 8.8 \times 10^{13} \text{ (non-embedding parameters)} \quad (1.1)$$

2. For large models trained with a limited dataset with early stopping:

$$L(D) = (D_c/D)^{\alpha_D}; \quad \alpha_D \sim 0.095, \quad D_c \sim 5.4 \times 10^{13} \text{ (tokens)} \quad (1.2)$$

3. When training with a limited amount of compute, a sufficiently large dataset, an optimally-sized model, and a sufficiently small batch size (making optimal³ use of compute):

$$L(C_{\min}) = (C_c^{\min}/C_{\min})^{\alpha_C^{\min}}; \quad \alpha_C^{\min} \sim 0.050, \quad C_c^{\min} \sim 3.1 \times 10^8 \text{ (PF-days)} \quad (1.3)$$

Open vs. Closed Access

Open/Closed Access

(e.g. Liang et al. 2022)

- **Weights:** open? described? closed?
- **Inference Code:** open? described? closed?
- **Training Code:** open? described? closed?
- **Data:** open? described? closed?

Licenses and Permissiveness


- **Public domain, CC-0:** old copyrighted works and products of US government workers
- **MIT, BSD:** very few restrictions
- **Apache, CC-BY:** must acknowledge owner
- **GPL, CC-BY-SA:** must acknowledge and use same license for derivative works
- **CC-NC:** cannot use for commercial purposes
- **LLaMa, OPEN-RAIL:** various other restrictions
- **No License:** all rights reserved, but can use under fair use

Fair Use

- US **fair use** doctrine — can use copyrighted material in some cases
- A gross simplification:
 - **Quoting** a small amount of material → likely OK
 - **Doesn't diminish** commercial value → possibly OK
 - Use for **non-commercial** purposes → possibly OK
- Most data on the internet is copyrighted, so model training is currently done assuming fair use
- But there are lawsuits!

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.



GitHub and Copilot Intellectual Property
Litigation

Why Restrict Model Access?


- **Commercial Concerns:** Want to make money from the models
- **Safety:** Limited release prevents possible misuse
- **Legal Liability:** Training models on copyrighted data is a legal/ethical gray area

English-Centric Open Models

Birds-eye View

- Open source/reproducible:
 - **Pythia:** Fully open, many sizes/checkpoints
 - **OLMo:** Possibly strongest reproducible model
- Open weights:
 - **LLaMa1/2/3/3.1:** Most popular, heavily safety tuned
 - **Mistral/Mixtral:** Strong and fast model, several European languages
 - **Qwen:** Strong, more multilingual - particularly en/zh

Pythia - Overview

- **Creator:**  ELEUTHERAI
- **Goal:** Joint understanding of model training dynamics and scaling
- **Unique features:** 8 model sizes 70M-12B, 154 checkpoints for each

Arch

Transformer+RoPE+SwiGLU, context 2k (cf LLaMa 4k),
parametric LN

Data

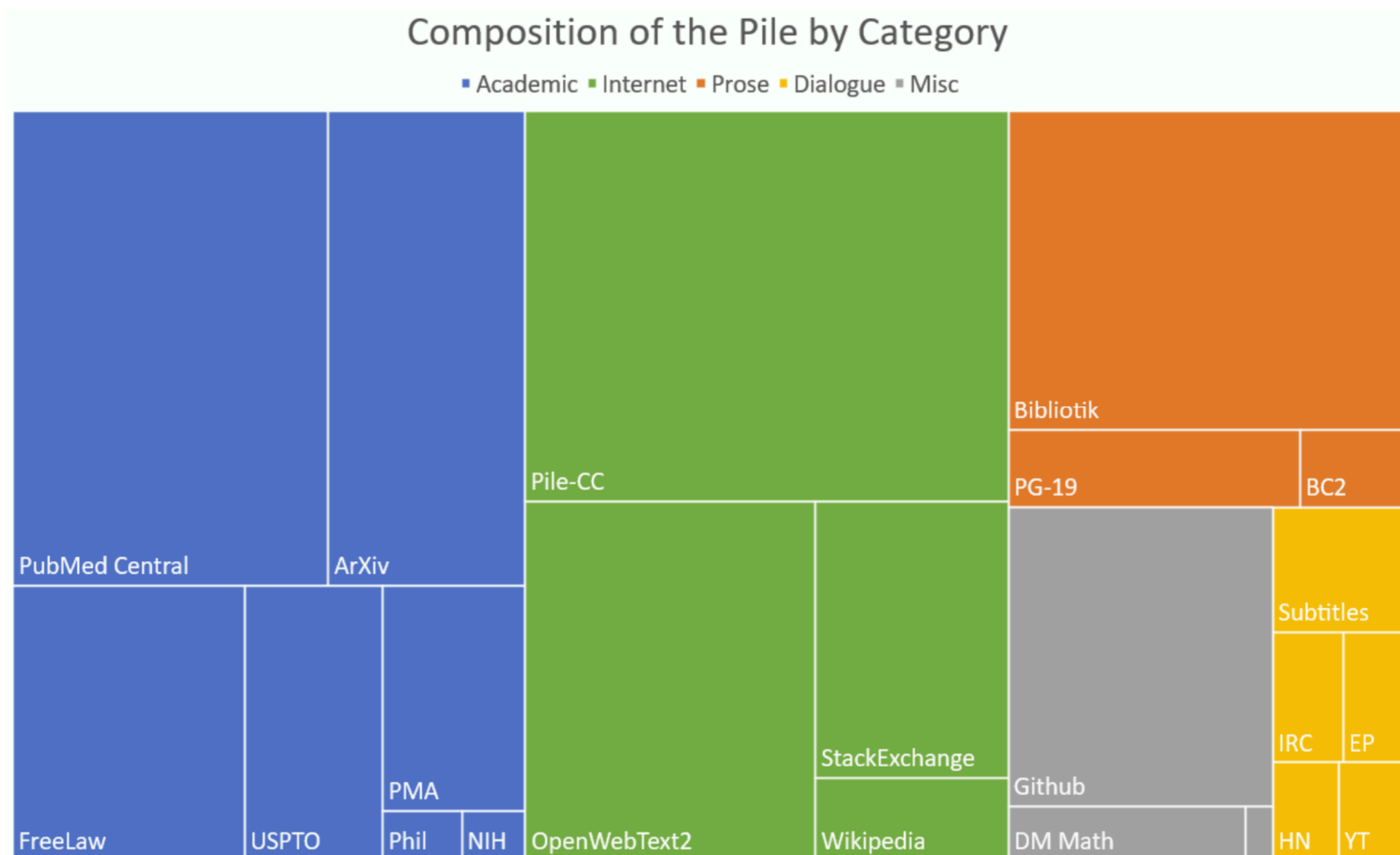
Trained on 300B tokens of The Pile (next slide), or deduped 207B

Train

LR scaled inversely to model size (7B=1.2e-4),
batch size 2M tokens

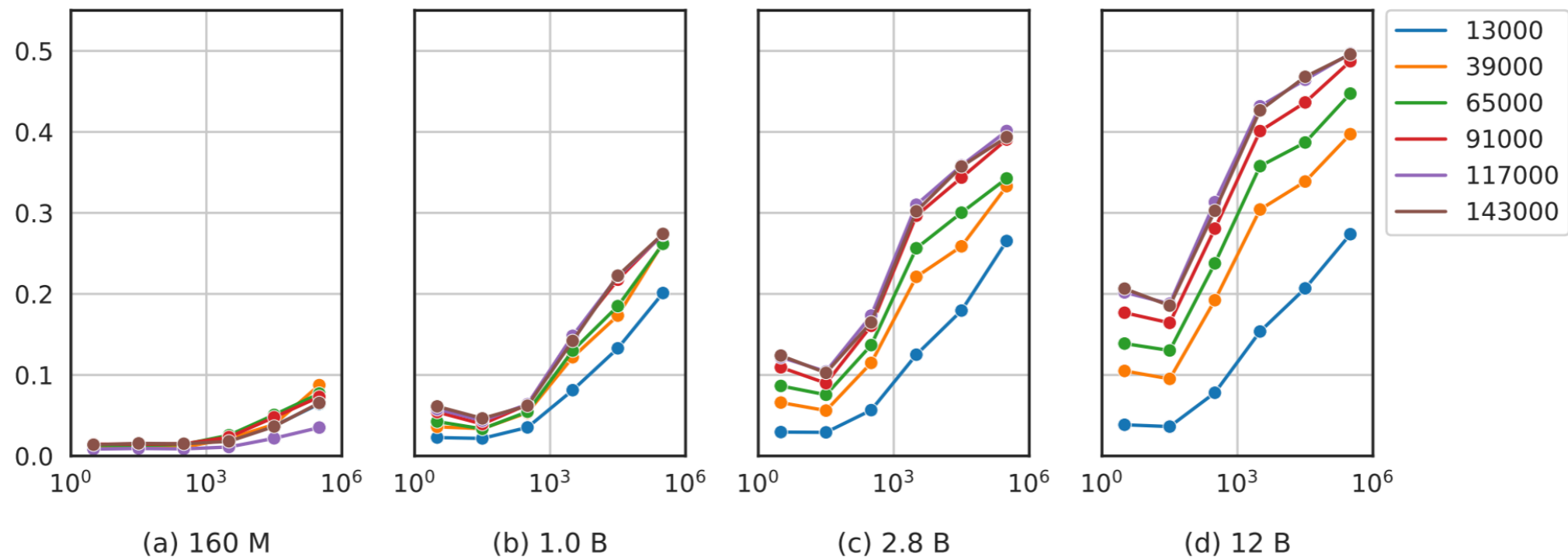
The Pile

- A now-standard 800GB dataset of lots of text/code




Pythia - Findings

- Some insights into training dynamics, e.g. larger models memorize facts more quickly (x axis: fact frequency, legend: training step)



- It is possible to intervene on data to reduce gender bias

OLMo - Overview

- **Creator:**  Allen Institute for AI
- **Goal:** Better science of state-of-the-art LMs
- **Unique features:** Top performance of fully documented model, instruction tuned etc.

Arch

Transformer+RoPE+SwiGLU, context 4k, non-parametric LN

Data








Trained on 2.46T tokens of Dolma corpus (next slide)

Train

LR scaled inversely to model size ($7B=3e-4$),
batch size 4M tokens

Dolma

- 3T token corpus created and released by AI2 for LM training
- a pipeline of (1) language filtering, (2) quality filtering, (3) content filtering, (4) deduplication, (5) multi-source mixing, and (6) tokenization

Source	Doc Type	UTF-8 bytes (GB)	Documents (millions)	Unicode words (billions)	Llama tokens (billions)
Common Crawl	 web pages	9,022	3,370	1,775	2,281
The Stack	 code	1,043	210	260	411
C4	 web pages	790	364	153	198
Reddit	 social media	339	377	72	89
PeS2o	 STEM papers	268	38.8	50	70
Project Gutenberg	 books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	 encyclopedic	16.2	6.2	3.7	4.3
Total		11,519	4,367	2,318	3,059

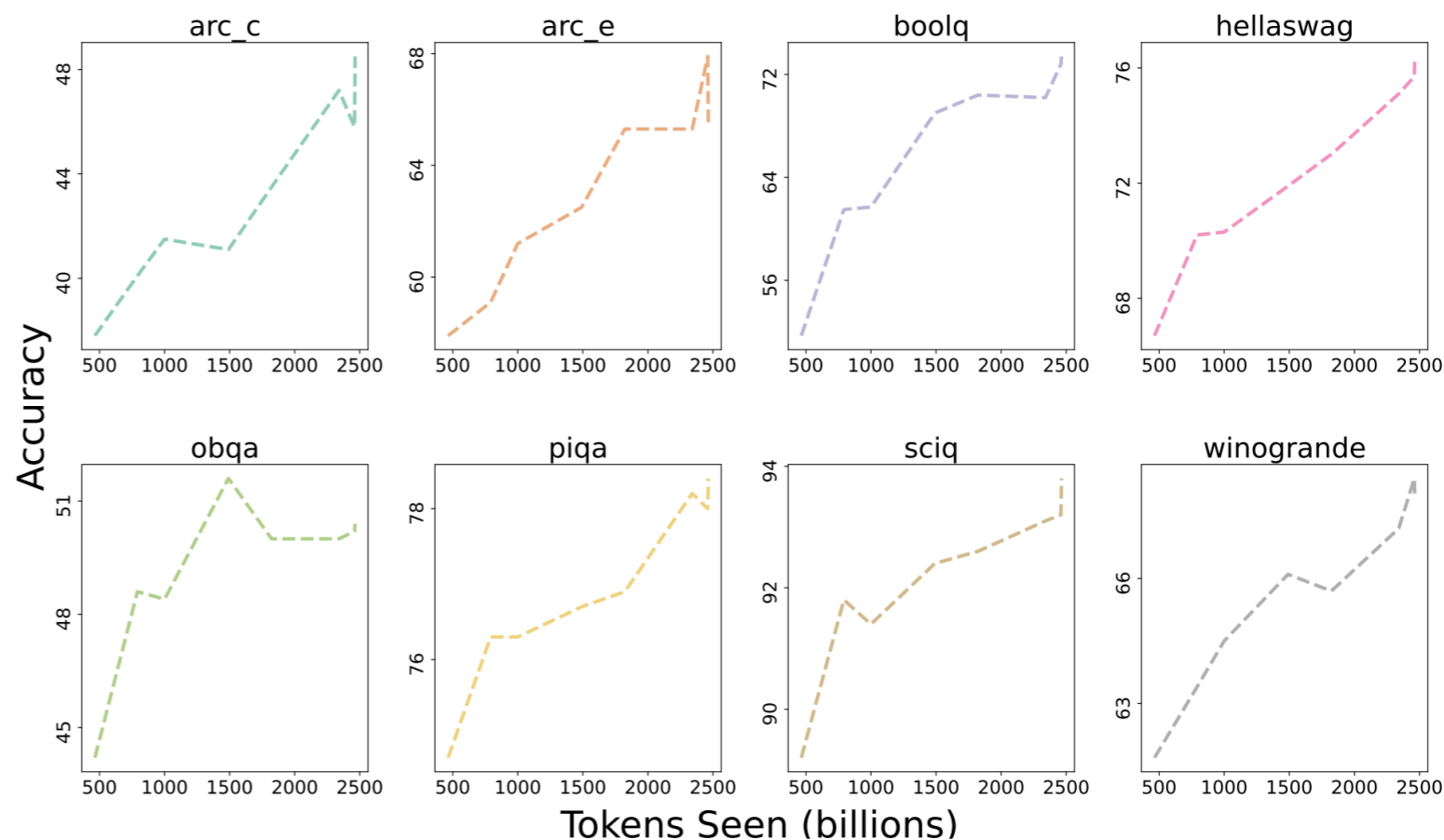
OLMo - Findings

- Competitive average performance


7B Models	arc challenge	arc easy	boolq	hella-swag	open bookqa	piqa	sciq	wino-grande	avg.
Falcon	47.5	70.4	74.6	75.9	53.0	78.5	93.9	68.9	70.3
LLaMA	44.5	67.9	75.4	76.2	51.2	77.2	93.9	70.5	69.6
Llama 2	48.5	69.5	80.2	76.8	48.4	76.7	94.5	69.4	70.5
MPT	46.5	70.5	74.2	77.6	48.6	77.3	93.7	69.9	69.8
Pythia	44.1	61.9	61.1	63.8	45.0	75.1	91.1	62.0	63.0
RPJ-INCITE	42.8	68.4	68.6	70.3	49.4	76.0	92.9	64.7	66.6
OLMo-7B	48.5	65.4	73.4	76.4	50.4	78.4	93.8	67.9	69.3

Table 6: Zero-shot evaluation of OLMo-7B and 6 other publicly available comparable model checkpoints on 8 core tasks from the downstream evaluation suite described in Section 2.4. For OLMo-7B, we report results for the 2.46T token checkpoint.

- Performance increases constantly w/ training



LLaMa2 - Overview

- **Creator:**  Meta
- **Goal:** Strong and safe open LM w/ base+chat versions
- **Unique features:** Open model with strong safeguards and chat tuning, good performance

Arch

Transformer+RoPE+SwiGLU, context 4k, RMSNorm


Data

Trained on “public sources, up-sampling the most factual sources”, LLaMa 1 has more info (next page), total 2T tokens

Train

$7B=3e-4$, batch size 4M tokens

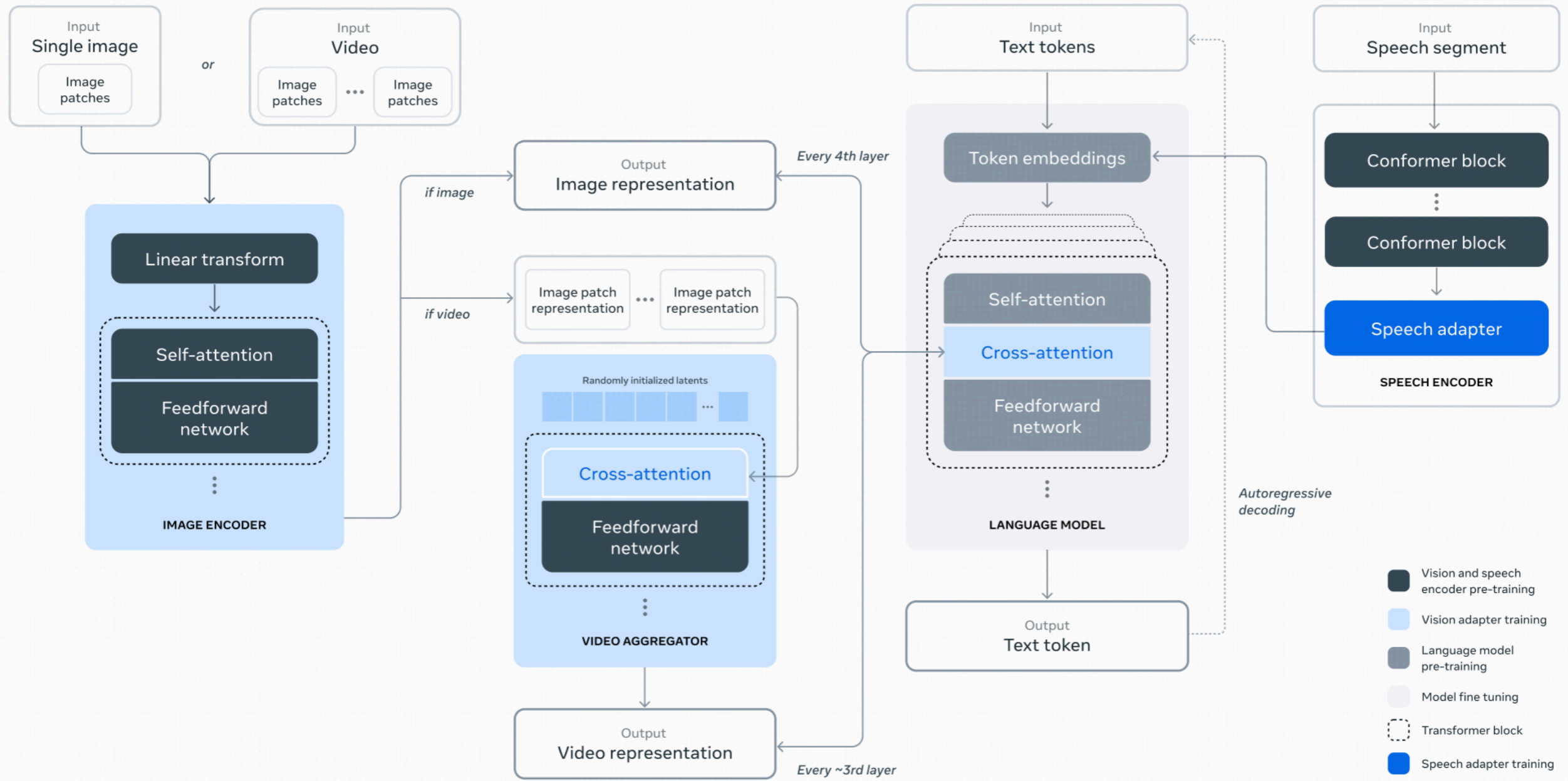
LLaMa3.1 - Overview

- **Creator:**  Meta
- **Goal:** A herd of language models that natively support multilinguality, coding, reasoning, and tool usage
- **Compared with Llama2:** Larger Data scale (15T multilingual tokens vs 1.8T tokens). More Training FLOPs (3.8×10^{25} FLOPs, almost 50x more than the largest version of Llama 2)

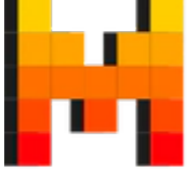
GPUs	TP	CP	PP	DP	Seq. Len.	Batch size/DP	Tokens/Batch	TFLOPs/GPU	BF16 MFU
8,192	8	1	16	64	8,192	32	16M	430	43%
16,384	8	1	16	128	8,192	16	16M	400	41%
16,384	8	16	16	4	131,072	16	16M	380	38%

Table 4 Scaling configurations and MFU for each stage of Llama 3 405B pre-training. See text and Figure 5 for descriptions of each type of parallelism.

LLaMa3.1 - Multimodality



Mistral/Mixtral - Overview

- **Creator:**  MISTRAL AI_
- **Goal:** Strong and somewhat multilingual open LM
- **Unique features:** Speed optimizations, including GQA and Mixture of Experts

Arch

Transformer+RoPE+SwiGLU, context 4k, RMSNorm, sliding window attention. Mixtral has 8x experts in feed-forward layer

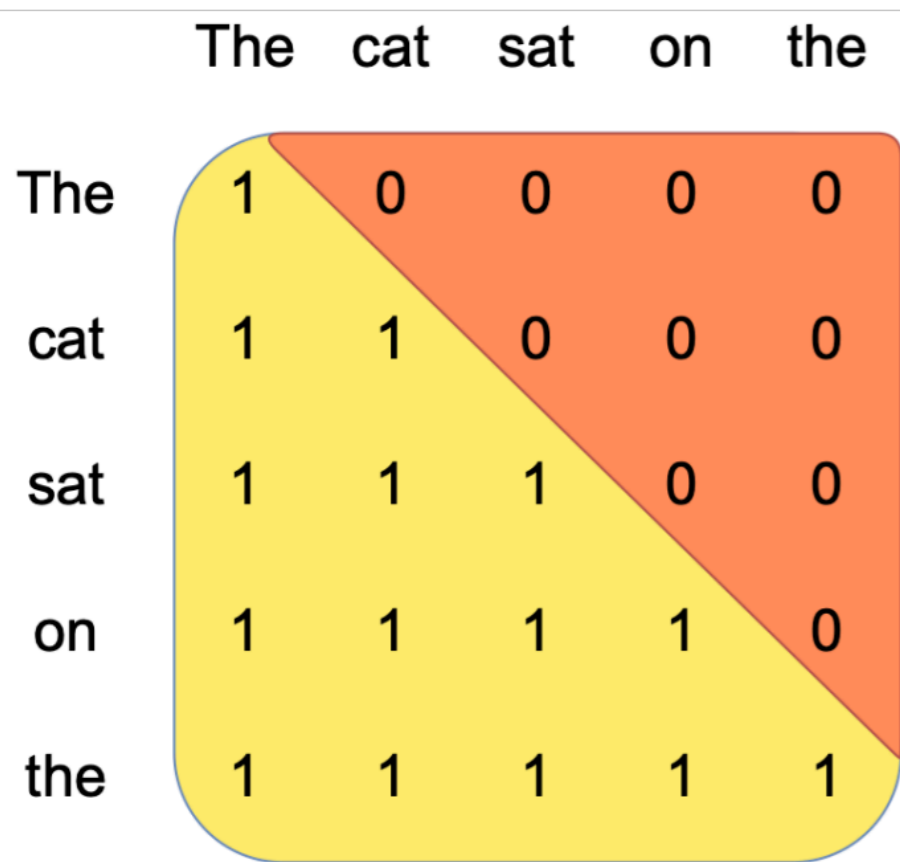
Data

Not disclosed?
But includes English and European languages

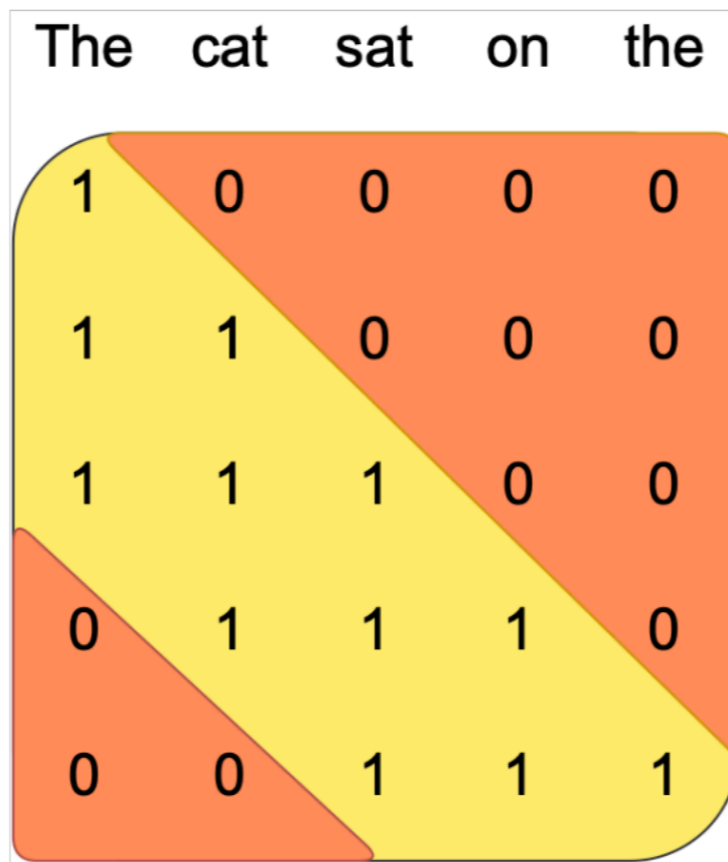
Train

Not disclosed?

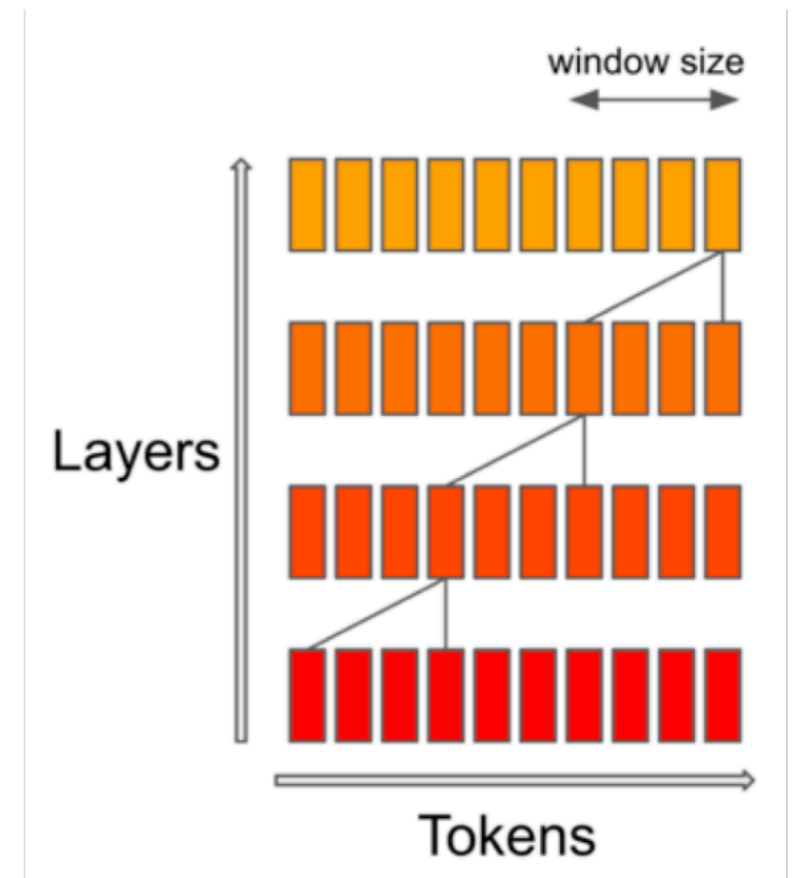
Mistral - Sliding Window Attention



Vanilla Attention



Sliding Window Attention



Effective Context Length

Qwen - Overview

- **Creator:**  **Alibaba**
- **Goal:** Strong multilingual (esp. English and Chinese) LM
- **Unique features:** Large vocabulary for multilingual support, strong performance

Arch

Transformer+RoPE+SwiGLU, context 4k, RMSNorm, bias in attention layer

Data

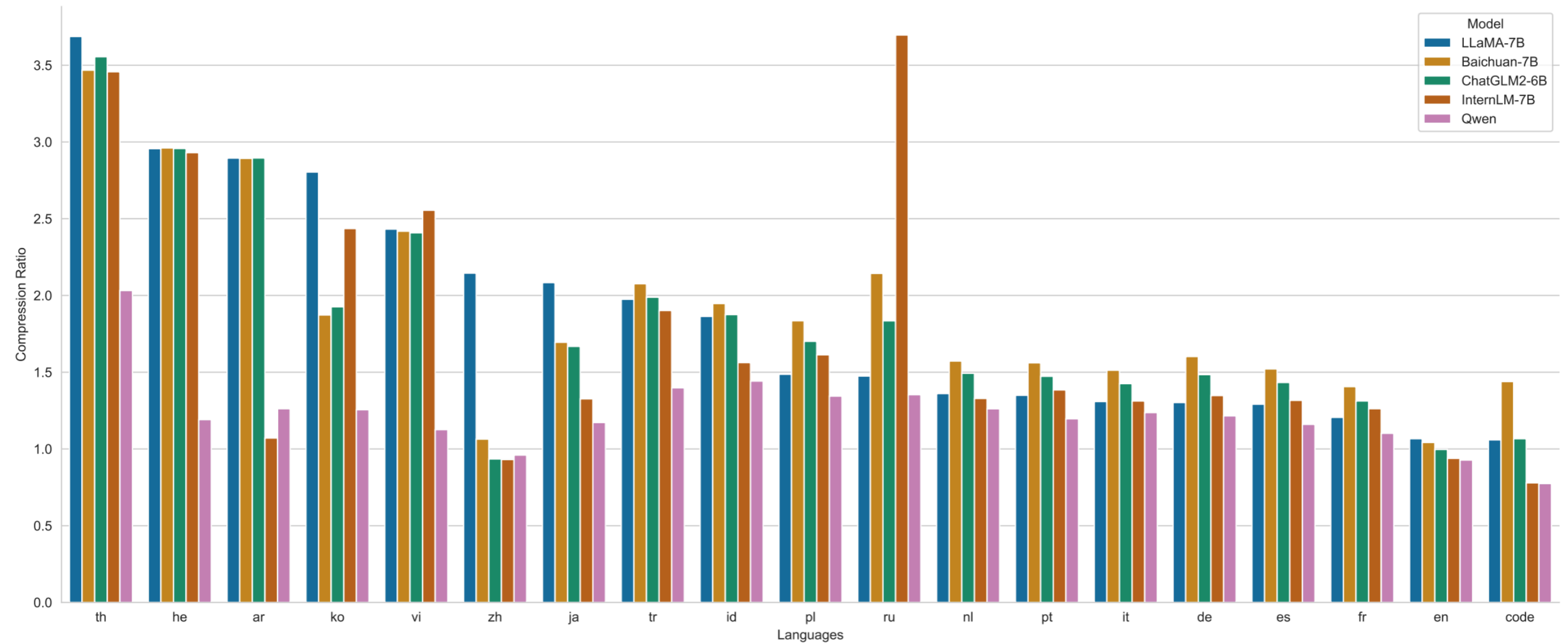
Trained on multilingual data + instruction data at pre-training time, 2-3T tokens

Train

$3e-4$, batch size 4M tokens

Qwen - Multilinguality

- Token compression ratio re: XLM-R (lower is better)



SmolLM - Overview

- **Creator:** 🤗 **Hugging Face**
- **Goal:** Small scale (135M, 360M, and 1.7B parameters) but strong performance
- **Unique features:** Fully Open-sourced with a high-quality pre-training corpus.
- **Cosmopedia v2:** A collection of synthetic textbooks and stories generated by Mixtral (28B tokens)
- **Python-Edu:** educational Python samples from The Stack (4B tokens)
- **FineWeb-Edu (deduplicated):** educational web samples from FineWeb (220B tokens)

<https://huggingface.co/blog/smollm>

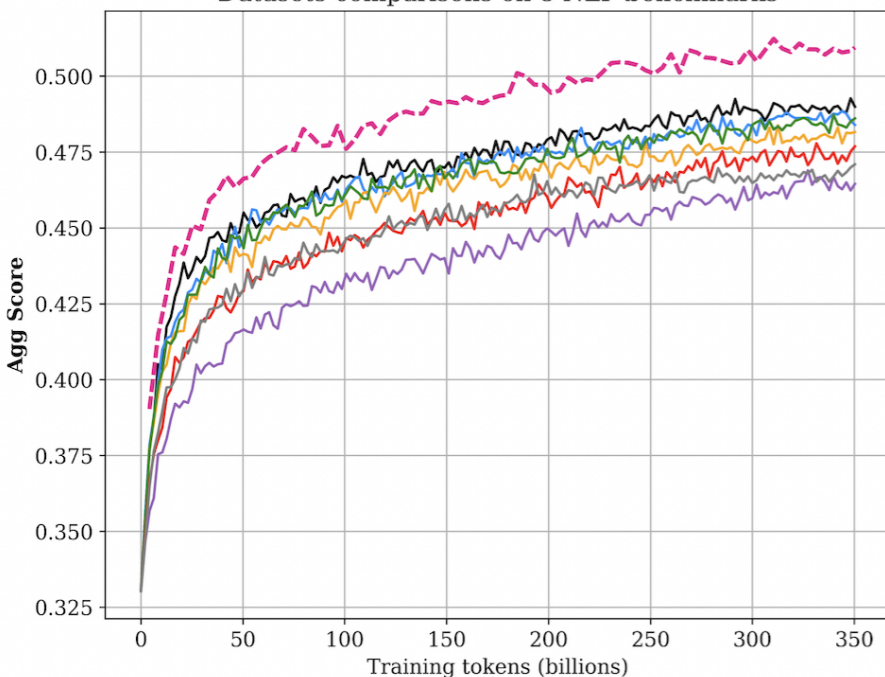
FineWeb - (Edu)

🍷 FineWeb dataset consists of more than 15T tokens of cleaned and deduplicated english web data from CommonCrawl.

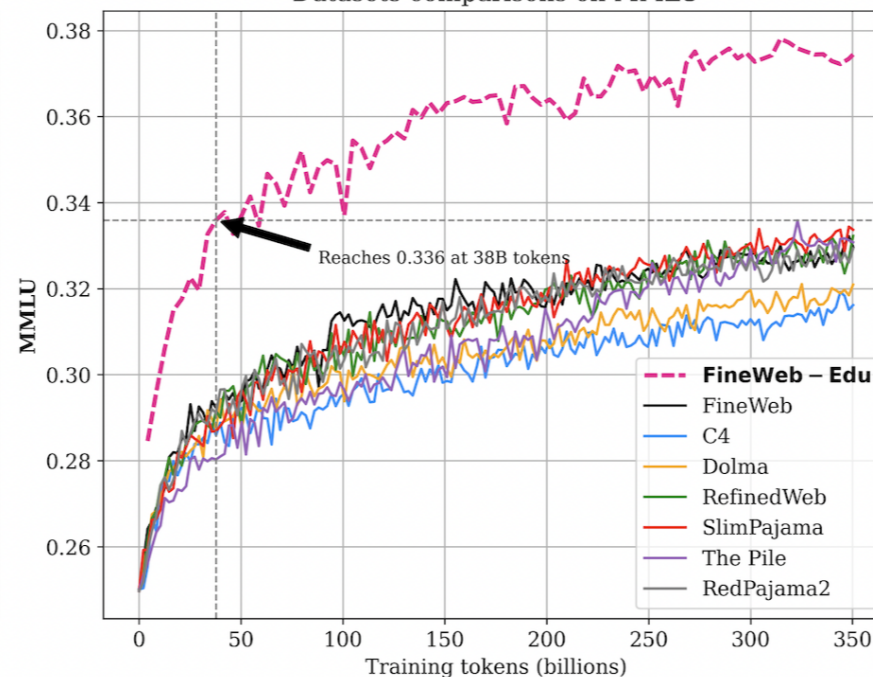
Url Filtering -> Trafilatura text extraction from HTML -> FastText LanguageFilter -> Quality filtering -> MinHash deduplication -> PII Formatting

“To enhance FineWeb's quality, we developed an educational quality classifier using annotations generated by LLama3-70B-Instruct. We then used this classifier to retain only the most educational web pages.”

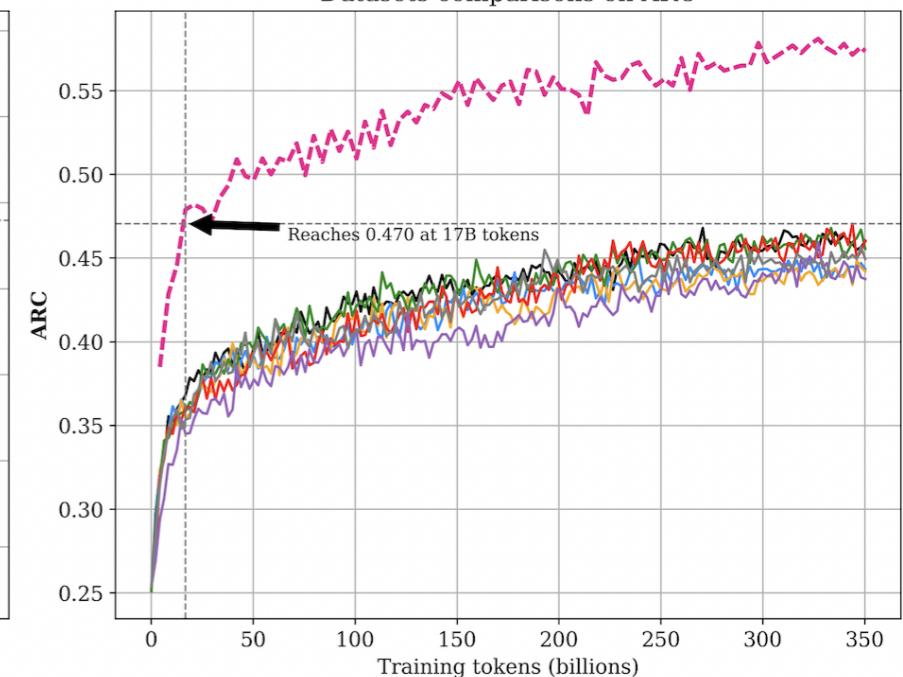
Datasets comparisons on 8 NLP benchmarks



Datasets comparisons on MMLU



Datasets comparisons on ARC



Other Models

Code Models

- **StarCoder 2** — by Big Science (leads: Hugging Face + Service Now), fully open model
- **CodeLlama** — by Meta, code adaptation of LLaMa
- **DeepSeek Coder** — by DeepSeek, strong performance across many tasks
- **Yi Coder** - by 01.AI, smaller scales (9B/1.5B) but strong performance.
- More in code generation class!

Math Models

- **LLEMA** — by EleutherAI and others, model for math theorem proving trained on proof pile
- **DeepSeek Math** — by DeepSeek, finds math-related pages on the web
- More in code and math class!

Science Model: Galactica

- Model for science trained by Meta
- Diverse set of interesting training data


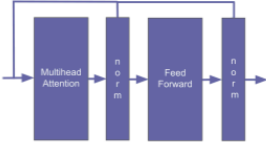
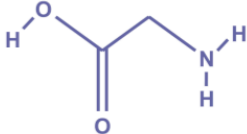
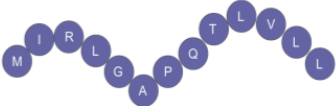
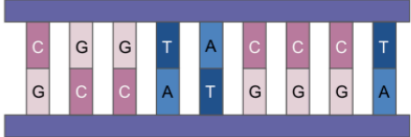

Modality	Entity	Sequence	
Text	Abell 370	Abell 370 is a cluster...	
L ^A T _E X	Schwarzschild radius	$r_{\{s\}} = \frac{2GM}{c^2}$	$r_s = \frac{2GM}{c^2}$
Code	Transformer	<code>class Transformer(nn.Module)</code>	
SMILES	Glycine	<chem>C(C(=O)O)N</chem>	
AA Sequence	Collagen α -1(II) chain	MIRLGAPQTL...	
DNA Sequence	Human genome	CGGTACCCTC...	


Table 1: Tokenizing Nature. Galactica trains on text sequences that represent scientific phenomena.

Closed Models

GPT-4o - Overview

- **Creator:**  **OpenAI**
- De-facto standard “strong” language model
- Tuned to be good as a chat-based assistant
- Supports calling external tools through “function calling” interface
- Accepts image inputs
- Fast and cheaper inference compared with earlier GPT-4 versions

Gemini

- **Creator:**  Google DeepMind
- Performance competitive with corresponding GPT models (Gemini Pro 1.0 ~ gpt-3.5, Gemini Ultra 1.0 ~ gpt-4)
- Pro 1.5 supports very long inputs, 1-10M tokens
- Supports image and video inputs
- Can generate images natively

Claude 3 - Overview

- **Creator:** ANTHROPIC
- Context window up to 200k
- Allows for processing images
- Overall strong results competitive with GPT-4

Questions?