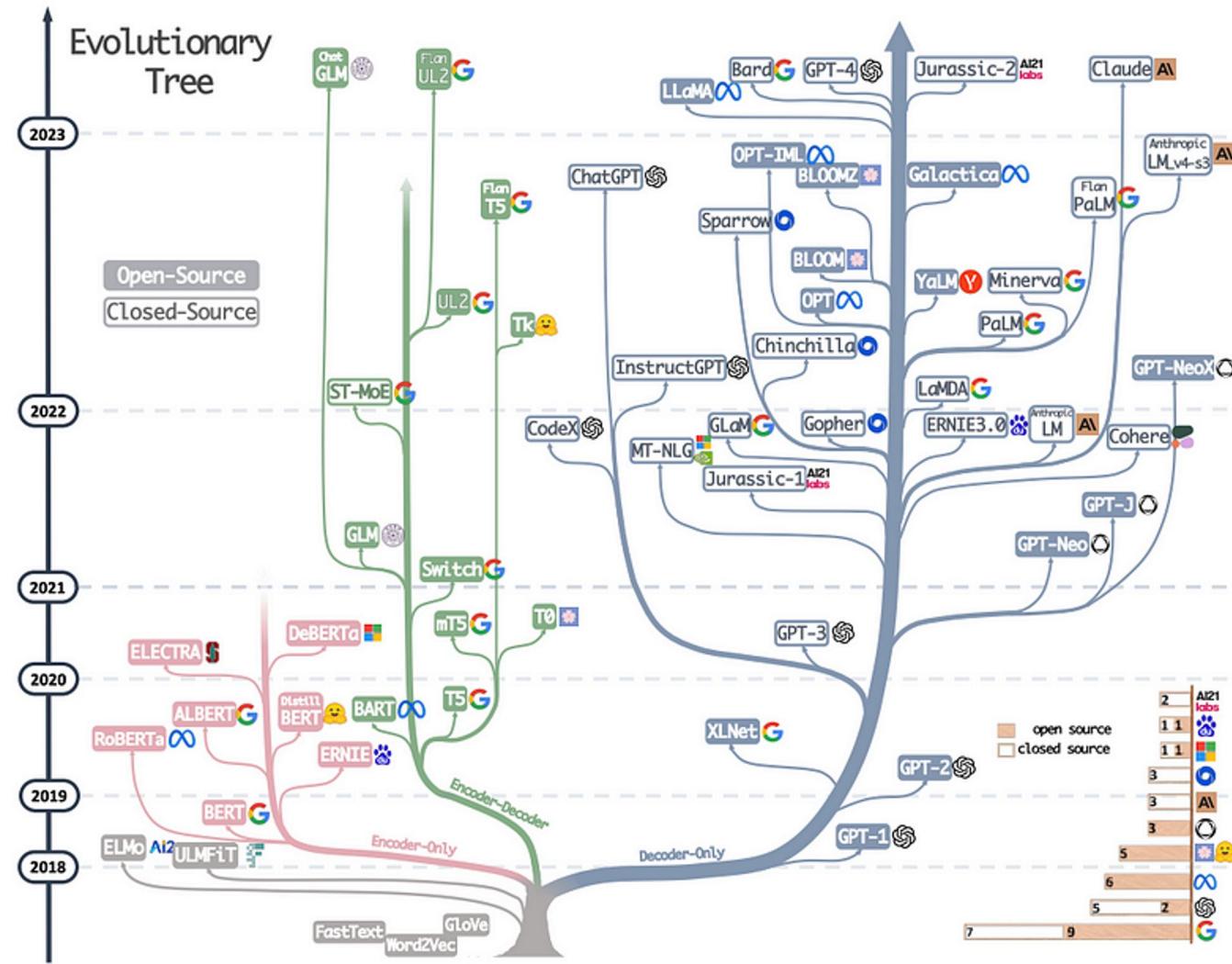


# Large Language Models

## Model Zoo

Sharif University of Technology  
Fall 2023





# Background

# Background

- Tokenization
  - WordPiece
  - BPE
  - UnigramLM
- Attention
  - Self-Attention
  - Cross Attention
  - Full Attention
  - Sparse Attention
  - Flash Attention

# Background

- Layer Normalization
  - LayerNorm
  - RMSNorm
  - Pre-LN
  - Post-LN
- Position Encoding
  - Absolute
  - Relative
    - Alibi
    - RoPE

# Background

- Activation Functions
  - ReLU
  - GeLU
  - GLU variants
- Training
  - Data Parallelism
  - Tensor Parallelism
  - Pipeline Parallelism
  - Model Parallelism
  - 3D Parallelism
  - Optimizer Parallelism

# Background

- PreProcessing
  - Quality Filtering
  - Data Deduplication
  - Privacy Reduction
- Architectures
  - Encoder
  - Causal Decoder
  - Prefix Decoder
- Objectives
  - Full Language Modeling
  - Prefix Language Modeling
  - Masked Language Modeling
  - Unified Language Modeling

# Background

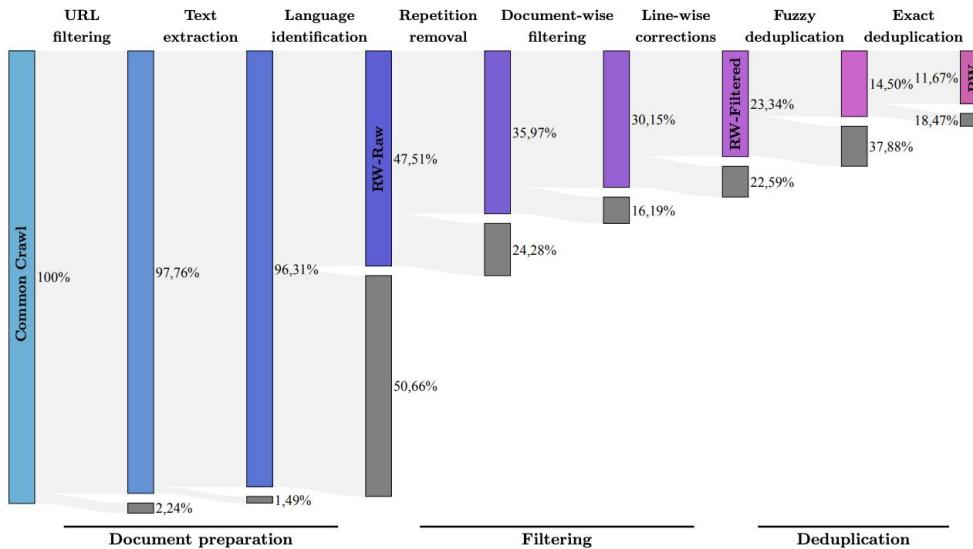
- Adaptation
  - Transfer Learning
  - Parameter Efficient Learning
    - Prompt Tuning
    - Prefix Tuning
    - Adapter Tuning
  - Instruction Finetuning
  - Alignment Tuning
  - In-context Learning
  - Chain-of-thought Prompting
- Dataset
- Model Size

# Dataset

# Dataset (RefinedWeb)

Dataset	Size	Availability	Web	CC Processing	Deduplication
<b>MASSIVE WEB DATASETS</b>					
<b>C4</b>	~ 360GT	Public	100%	Rules + NSFW words blocklist	<b>Exact:</b> spans of 3 sentences
<b>OSCAR-21.09</b>	~ 370GT	Public	100%	Built at the line-level	<b>Exact:</b> per line (~ 55% removed)
<b>OSCAR-22.01</b>	~ 283GT	Public	100%	Line-level rules + optional rules & NSFW URL blocklist	<b>Exact:</b> per line (optional, not used for results in this paper)
<b>CURATED DATASETS</b>					
<b>GPT-3</b>	300GT	Private	60%	Content filter trained on known high-quality sources	<b>Fuzzy:</b> MinHash (~ 10% removed)
<b>The Pile</b>	~ 340GT	Public	18%	jusText for extraction, content filter trained on curated data	<b>Fuzzy:</b> MinHash (~ 26% removed)
<b>PaLM</b>	780GT	Private	27%	Filter trained on HQ data	Unknown
<b>OURS</b>					
<b>REFINEDWEB</b>	~ 5,000GT	Public (600GT)	100%	trafilatura for text extraction, document and line-level rules, NSFW URL blocklist	<b>Exact &amp; fuzzy:</b> exact substring+MinHash (~ 50% removed)

# Dataset (RefinedWeb)



**Figure 2. Subsequent stages of Macrodata Refinement remove nearly 90 % of the documents originally in CommonCrawl.** Notably, filtering and deduplication each result in a halving of the data available: around 50% of documents are discarded for not being English, 24% of remaining for being of insufficient quality, and 12% for being duplicates. We report removal rate (grey) with respect to each previous stage, and kept rate (shade) overall. Rates measured in % of documents in the document preparation phase, then in tokens.

# Dataset (The Pile)

Component	Raw Size	Weight	Epochs	Effective Size	Mean Document Size
Pile-CC	227.12 GiB	18.11%	1.0	227.12 GiB	4.33 KiB
PubMed Central	90.27 GiB	14.40%	2.0	180.55 GiB	30.55 KiB
Books3 <sup>†</sup>	100.96 GiB	12.07%	1.5	151.44 GiB	538.36 KiB
OpenWebText2	62.77 GiB	10.01%	2.0	125.54 GiB	3.85 KiB
ArXiv	56.21 GiB	8.96%	2.0	112.42 GiB	46.61 KiB
Github	95.16 GiB	7.59%	1.0	95.16 GiB	5.25 KiB
FreeLaw	51.15 GiB	6.12%	1.5	76.73 GiB	15.06 KiB
Stack Exchange	32.20 GiB	5.13%	2.0	64.39 GiB	2.16 KiB
USPTO Backgrounds	22.90 GiB	3.65%	2.0	45.81 GiB	4.08 KiB
PubMed Abstracts	19.26 GiB	3.07%	2.0	38.53 GiB	1.30 KiB
Gutenberg (PG-19) <sup>†</sup>	10.88 GiB	2.17%	2.5	27.19 GiB	398.73 KiB
OpenSubtitles <sup>†</sup>	12.98 GiB	1.55%	1.5	19.47 GiB	30.48 KiB
Wikipedia (en) <sup>†</sup>	6.38 GiB	1.53%	3.0	19.13 GiB	1.11 KiB
DM Mathematics <sup>†</sup>	7.75 GiB	1.24%	2.0	15.49 GiB	8.00 KiB
Ubuntu IRC	5.52 GiB	0.88%	2.0	11.03 GiB	545.48 KiB
BookCorpus2	6.30 GiB	0.75%	1.5	9.45 GiB	369.87 KiB
EuroParl <sup>†</sup>	4.59 GiB	0.73%	2.0	9.17 GiB	68.87 KiB
HackerNews	3.90 GiB	0.62%	2.0	7.80 GiB	4.92 KiB
YoutubeSubtitles	3.73 GiB	0.60%	2.0	7.47 GiB	22.55 KiB
PhilPapers	2.38 GiB	0.38%	2.0	4.76 GiB	73.37 KiB
NIH ExPorter	1.89 GiB	0.30%	2.0	3.79 GiB	2.11 KiB
Enron Emails <sup>†</sup>	0.88 GiB	0.14%	2.0	1.76 GiB	1.78 KiB
<b>The Pile</b>	<b>825.18 GiB</b>			<b>1254.20 GiB</b>	<b>5.91 KiB</b>

# Dataset (ROOTS Corpus)

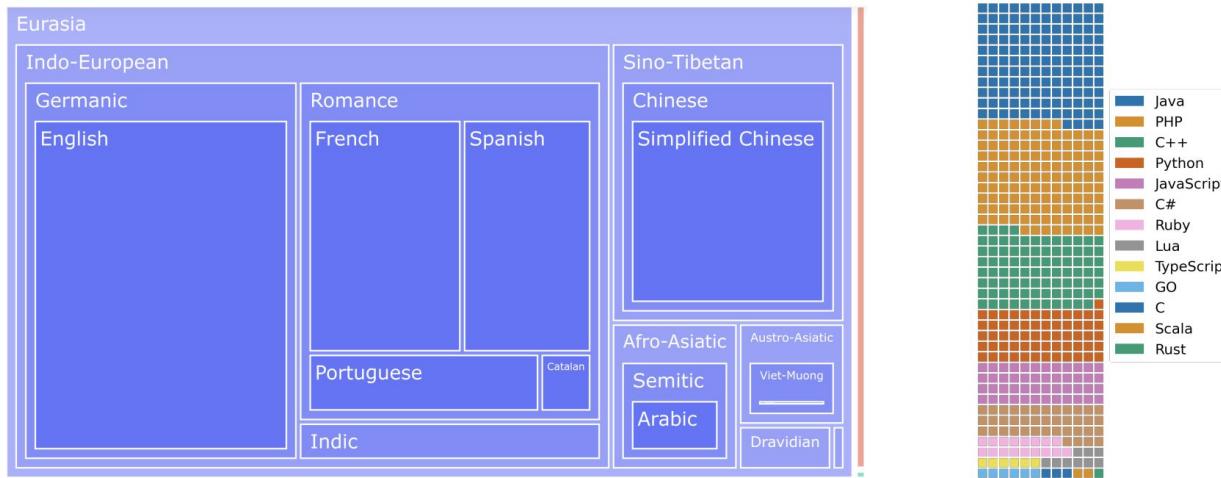


Figure 1: Overview of ROOTS. Left: A treemap of natural language representation in number of bytes by language family. The bulk of the graph is overwhelmed by the 1321.89 GB allotted to Eurasia. The orange rectangle corresponds to the 18GB of Indonesian, the sole representative of the Papunesia macroarea, and the green rectangle to the 0.4GB of the Africa linguistic macroarea. Right: A waffle plot of the distribution of programming languages by number of files. One square corresponds approximately to 30,000 files.

# Dataset (ROOTS Corpus)

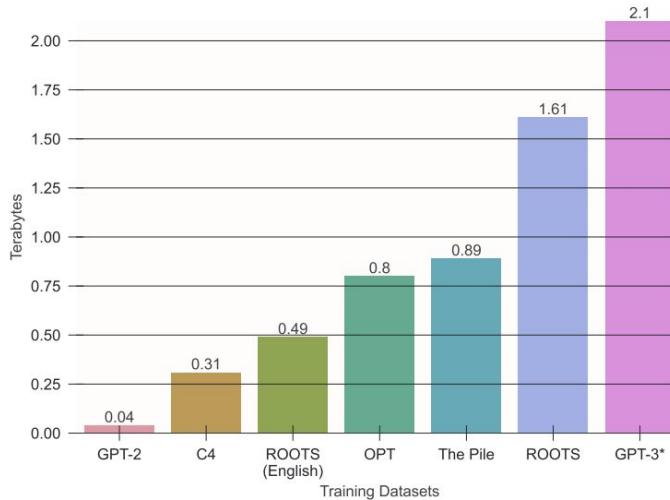


Figure 4: A raw size comparison to other corpora used to train large language models. The asterisk next to GPT-3 indicates the fact that the value in question is an estimate computed using the reported number of tokens and the average number of tokens per byte of text that the GPT-2 tokenizer produces on the Pile-CC, Books3, OWT2, and Wiki-en subsets of the Pile (Gao et al., 2020)

# Dataset (MassiveText)

- MassiveWeb
- Books
- C4
- News
- GitHub
- Wikipedia



# Models

# BigBird

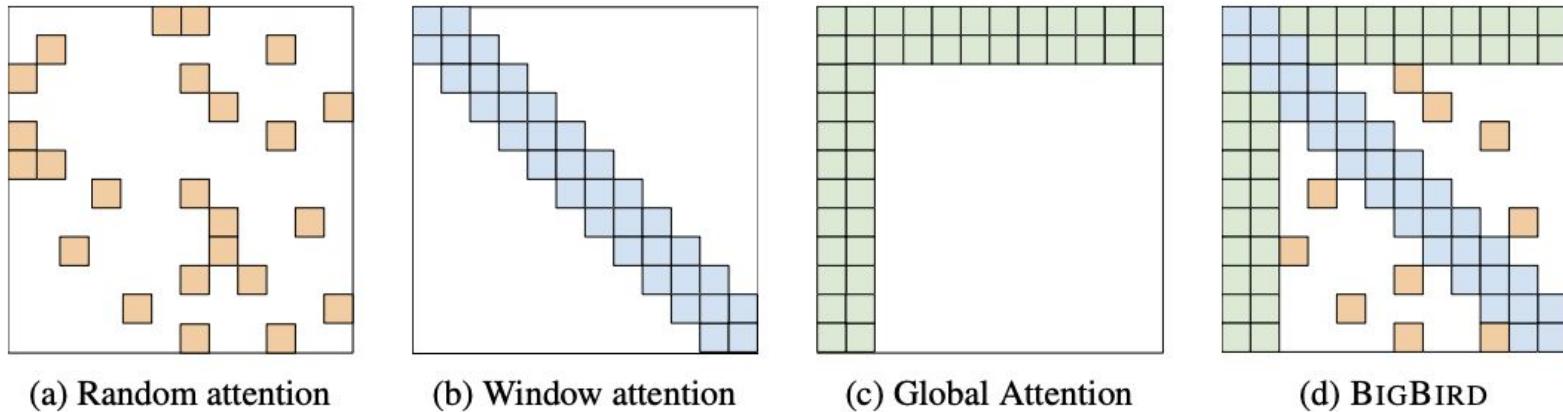


Figure 1: Building blocks of the attention mechanism used in BIGBIRD. White color indicates absence of attention. (a) random attention with  $r = 2$ , (b) sliding window attention with  $w = 3$  (c) global attention with  $g = 2$ . (d) the combined BIGBIRD model.

# T5

- Google
- Encoder-decoder transformer
  - No bias in layer normalization
  - Using relative positional embedding
  - Placing layer normalization outside the residual path
- CC
  - 20 TB each month
- C4
  - 750 GB
- mT5
  - mC4



# T5

- Machine translation
  - WMT
- QA
  - SQuAD
- Summarization
  - NEWS
- Text classification
  - GLUE
- Task prefix



# T5



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Oct-2019	11	1T	Enc-Dec	Span Corruption	SentencePiece	Relative	AdaFactor

# T5

Original text

Thank you ~~for~~ ~~inviting~~ me to your party ~~last~~ week.

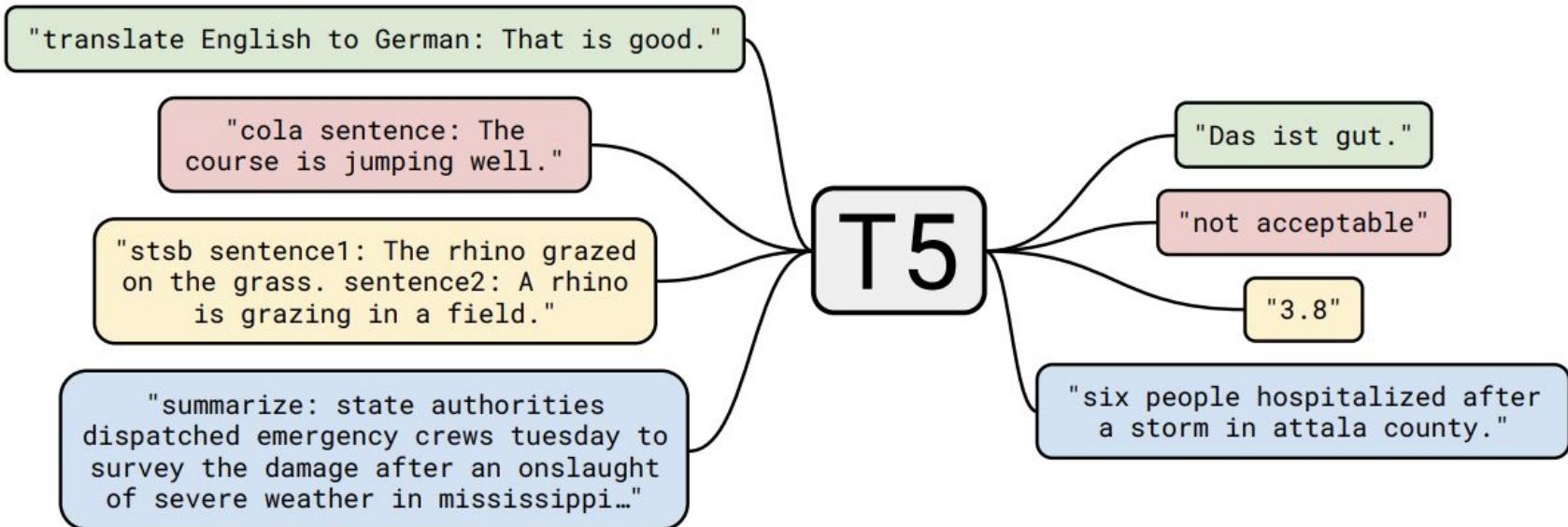
Inputs

Thank you <X> me to your party <Y> week.

Targets

<X> for inviting <Y> last <Z>

# T5



# mT5



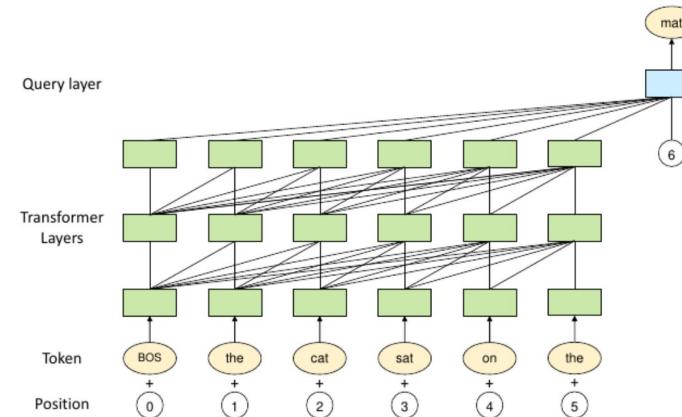
- 101 languages
- mC4
- Data sampling

Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Oct-2020	13	1T	Enc-Dec	Span Corruption	SentencePiece	Relative	AdaFactor

# PanGu- $\alpha$



- Autoregressive model
  - Query layer after stacked transformer layers
- 1.1TB Chinese data
  - Common Crawl, e-Books, encyclopedia, etc.



# PanGu- $\alpha$



盤古

PanGu-Alpha

Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Apr-2021	200	1.1TB	Causal-Dec	Next Token	BPE	-	-



# CodeGen

- Causal-Dec
  - PaLM
- Natural language and programming language data
- Multi-step approach
- Train sequentially
  - PILE
  - BIGQUERY
  - BIGPYTHON

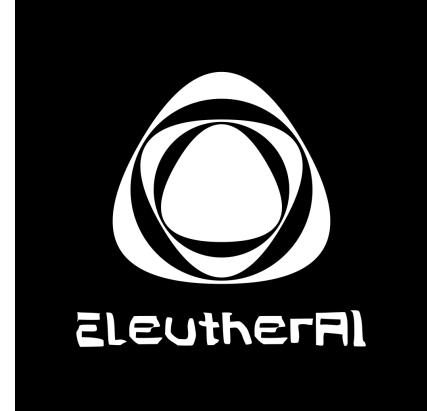
# CodeGen



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Mar-2022	16	577B	Causal-Dec	Next Token	BPE	RoPE	Adam

# GPT-NeoX-20B

- Eleuther AI
- Auto-regressive model
- Trained on the Pile dataset without any data deduplication
- Parallel attention and feed-forward layers
- Rotary positional embedding
  - 25% of embedding vector dimension
- Hyperparameter interpolation
- ZeRO optimizer



# GPT-NeoX-20B



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Apr-2022	20	825GB	Causal-Dec	Next Token	BPE	Rotary	AdamW

# Parallel attention and feed-forward layers

- Standard serialized formulation

$$y = x + \text{MLP}(\text{LayerNorm}(x + \text{Attention}(\text{LayerNorm}(x))))$$

- Parallel formulation in each Transformer block

$$y = x + \text{MLP}(\text{LayerNorm}(x)) + \text{Attention}(\text{LayerNorm}(x))$$

- 15% faster training speed at large scales
- Small quality degradation at 8B scale but no quality degradation at 62B scale

# GPT-NeoX-20B

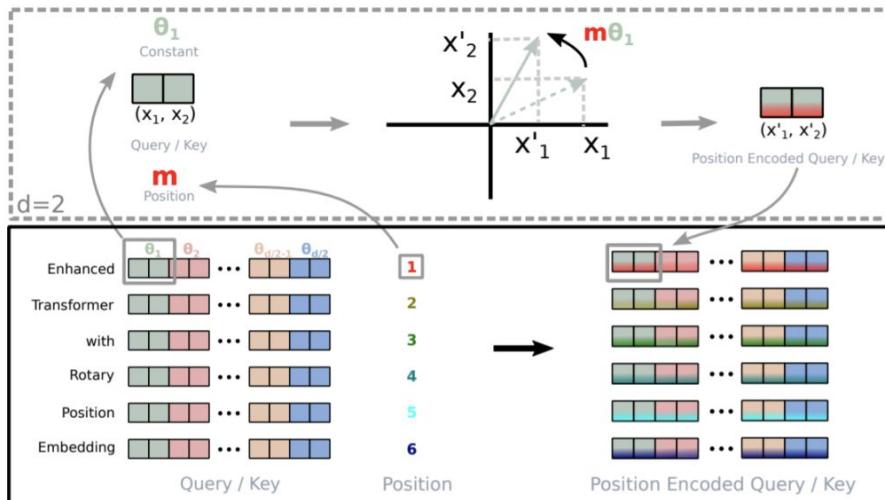


Figure 1: A pictorial representation of rotary embeddings, from Su et al. (2021).

GPT-2

```
def fibRec(n):←
  if n < 2:←
    return n←
  else:←
    return fibRec(n-1) + fibRec(n-2)←
55 tokens
```

GPT-NeoX-20B

```
def fibRec(n):←
  if n < 2:←
    return n←
  else:←
    return fibRec(n-1) + fibRec(n-2)←
39 tokens
```

Figure 3: GPT-2 tokenization vs. GPT-NeoX-20B tokenization. GPT-NeoX-20B tokenization handles whitespace better, which is particularly useful for text such as source code. For more examples, see Appendix F.

# GPT-NeoX-20B

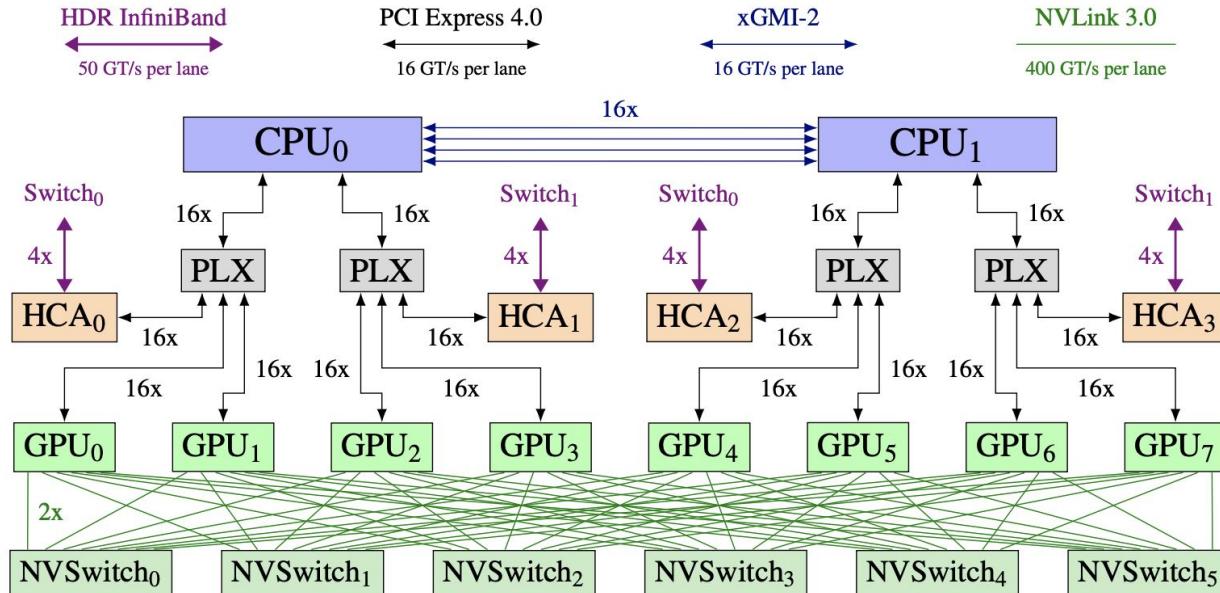
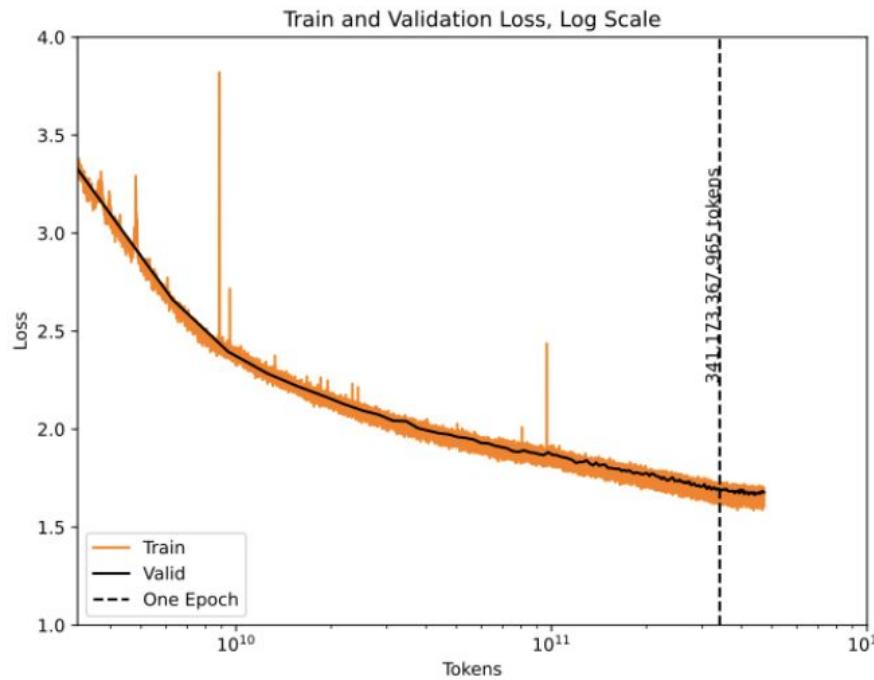


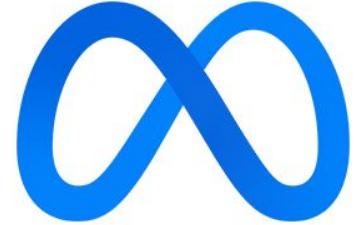
Figure 2: Architecture diagram of a single training node.

# GPT-NeoX-20B

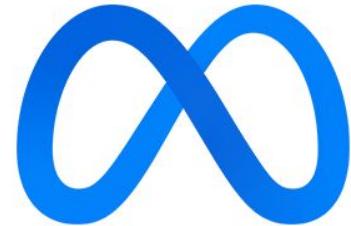


# OPT

- Meta AI
- Auto-regressive model
  - A clone of GPT-3
- De-duplicate The Pile
- Dataset
  - RoBERTa
  - The Pile
  - PushShift.io Reddit

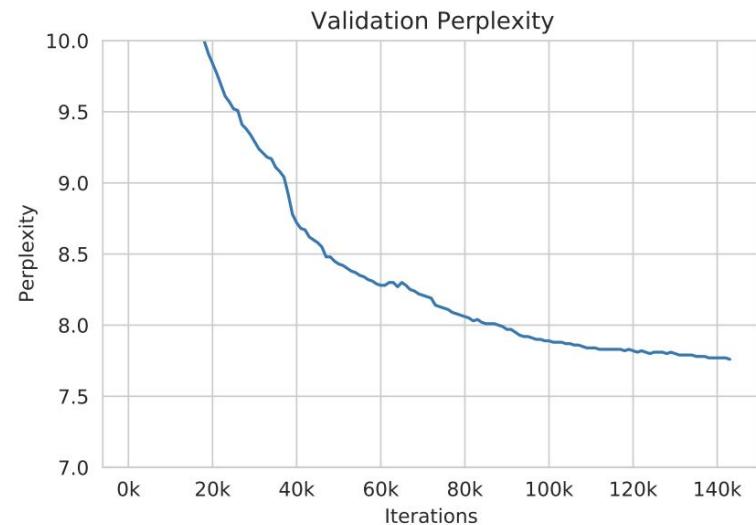
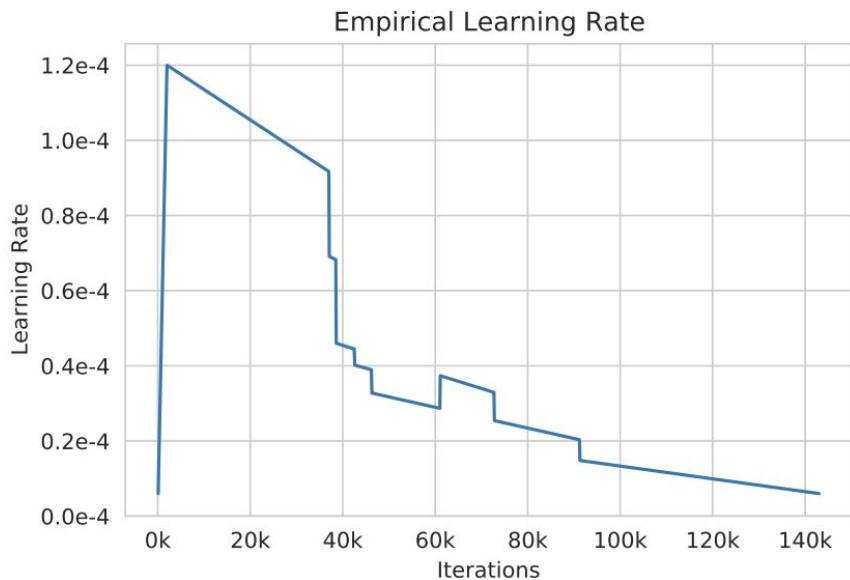


# OPT



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
May-2022	175	180B	Causal-Dec	Next Token	BPE	-	AdamW

# OPT



# OPT

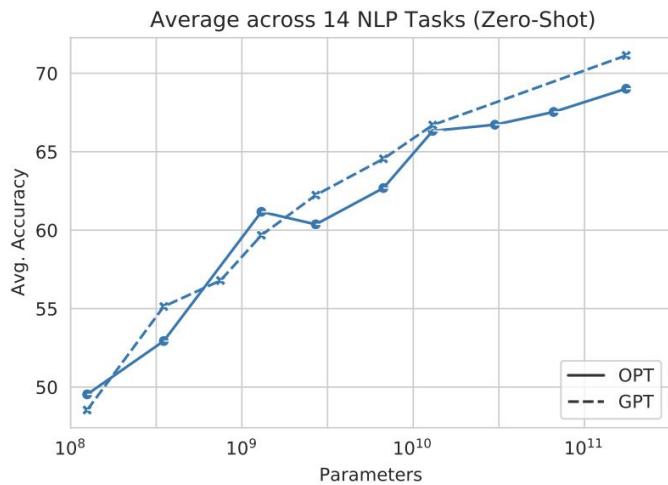


Figure 3: **Zero-shot NLP Evaluation Averages.** Across a variety of tasks and model sizes, OPT largely matches the reported averages of GPT-3. However, performance varies greatly per task: see Appendix A.

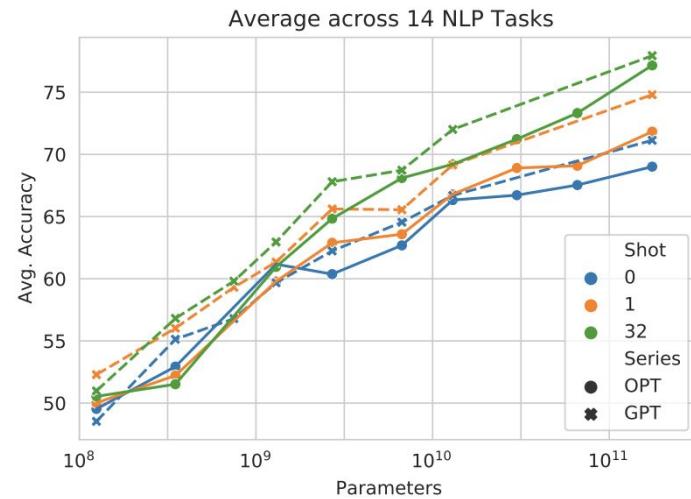


Figure 4: **Multi-shot performance.** OPT performance for one- and few-shot lags behind GPT-3 models, but performance depends heavily per task; see Appendix A.

# OPT

Setup	Davinci	OPT-175B
Zero-shot	.628	<b>.667</b>
One-shot	.616	<b>.713</b>
Few-shot (binary)	.354	<b>.759</b>
Few-shot (multiclass)	.672	<b>.812</b>

Table 3: **Hate speech detection.** F1 scores of detecting hate speech between Davinci and OPT-175B. OPT-175B considerably outperforms Davinci in all settings.

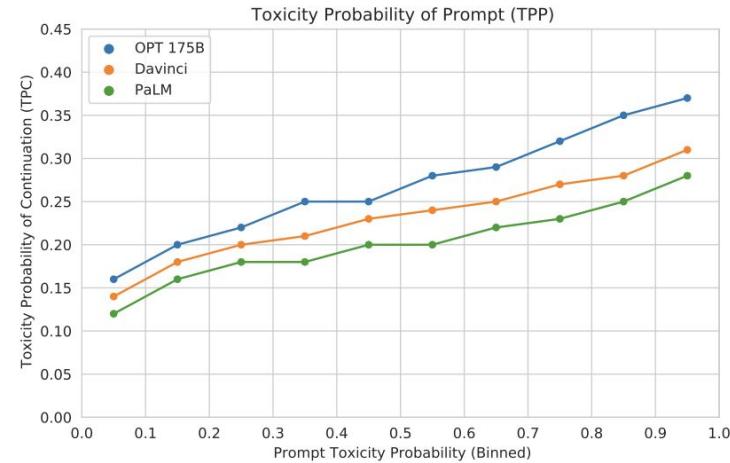


Figure 5: **RealToxicityPompts.** OPT-175B is more likely to generate toxic responses than either Davinci or PaLM. Consistent with prior work, toxicity rates increase as prompt toxicity increases.

# BLOOM

- Causal decoder model
  - ALiBi positional embedding
  - Additional normalization layer after the embedding layer
  - Full attention
- ROOTS corpus
- GPT-3 architecture and hyperparameters
- Diverse data source
- GELU



Activation function	Average EAI Results
GELU	42.79
SwiGLU	<b>42.95</b>

Table 3: **SwiGLU slightly outperforms GELU for zero-shot generalization.** Models trained on The Pile for 112 billion tokens.

Bloom: A 176bparameter open-access multilingual language model (ArXiv 2022)

What Language Model to Train if You Have One Million GPU Hours? (EMNLP Findings 2022)

# BLOOM

- ALiBi positional embeddings significantly outperforms other embeddings for zero-shot generalization
- Adding layer normalization after the embedding layer incurs a significant penalty on zero-shot generalization

Bloom: A 176bparameter open-access multilingual language model (ArXiv 2022)

What Language Model to Train if You Have One Million GPU Hours? (EMNLP Findings 2022)

# BLOOM



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Nov-2022	176	366B	Causal-Dec	Next Token	BPE	ALiBi	Adam

Bloom: A 176bparameter open-access multilingual language model (ArXiv 2022)

What Language Model to Train if You Have One Million GPU Hours? (EMNLP Findings 2022)

# BLOOM

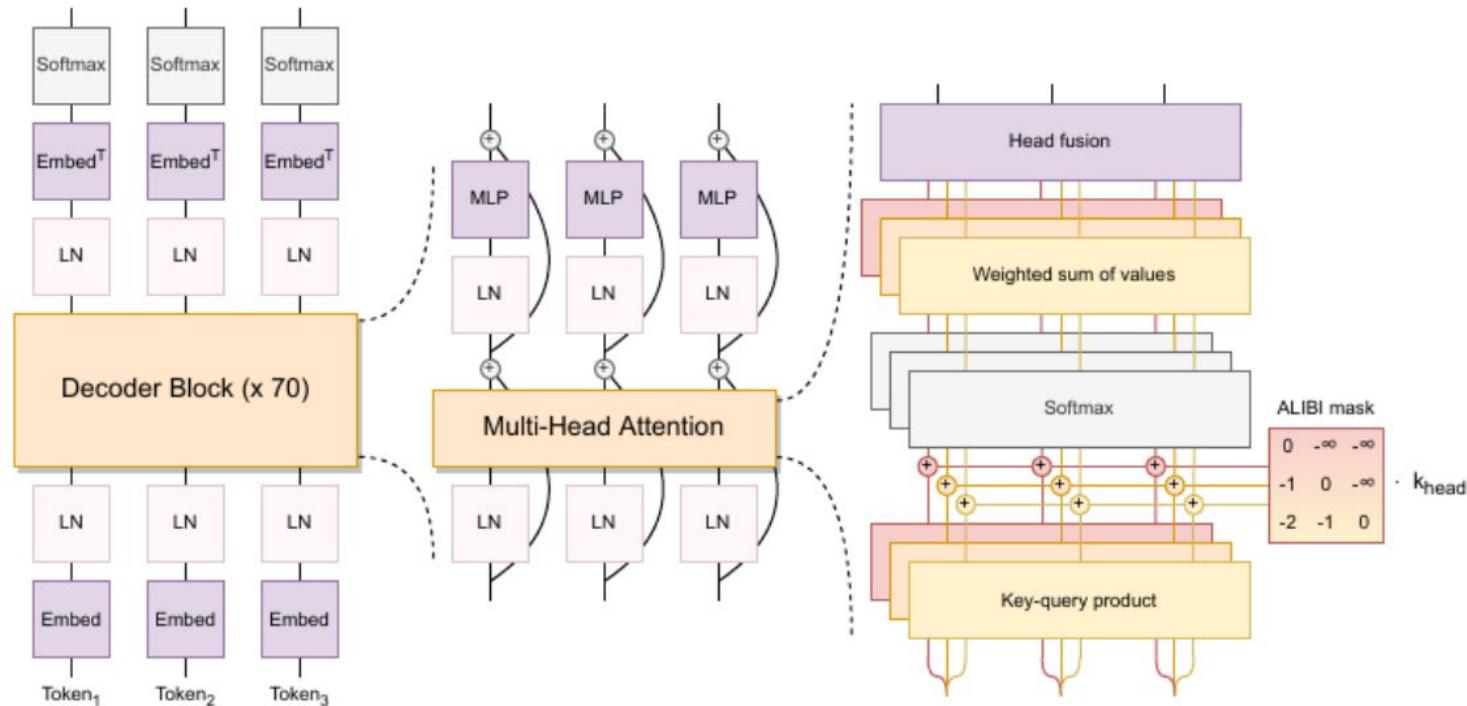
Model	Parameters	Pretraining tokens		
		Dataset	112B	250B
<b>OpenAI</b> — Curie	6.7B			<u>49.28</u>
<b>OpenAI</b> — Babbage	1.3B			<b>45.30</b>
<b>EleutherAI</b> — GPT-Neo	1.3B	The Pile		42.94
<b>Ours</b>	13B	OSCAR v1		47.09
<b>Ours</b>	1.3B	The Pile	<b>42.79</b>	43.12
	1.3B	C4	42.77	
	1.3B	OSCAR v1	41.72	

Table 1: **Pretraining datasets with diverse cross-domain high-quality data improves zero-shot generalization.** Average accuracy on EAI harness (higher is better) using different pretraining corpora and comparison with baseline models. **Bold is best 1.3B model for amount of tokens seen, underline is best overall.**

Bloom: A 176bparameter open-access multilingual language model (ArXiv 2022)

What Language Model to Train if You Have One Million GPU Hours? (EMNLP Findings 2022)

# BLOOM



Bloom: A 176bparameter open-access multilingual language model (ArXiv 2022)

# Galactica

GALACTICA

- Meta
- Dataset
  - 48 million papers
  - Textbooks
  - Lecture notes
  - Millions of compounds and protein
  - Scientific websites
  - Encyclopedias
  - ...
- Tokenization
  - Markdown
- <work> token
- Prompt pre-training

# Galactica

GALACTICA

Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Nov-2022	120	106B	Causal-Dec	Next Token	BPE	Learned	AdamW

# Galactica

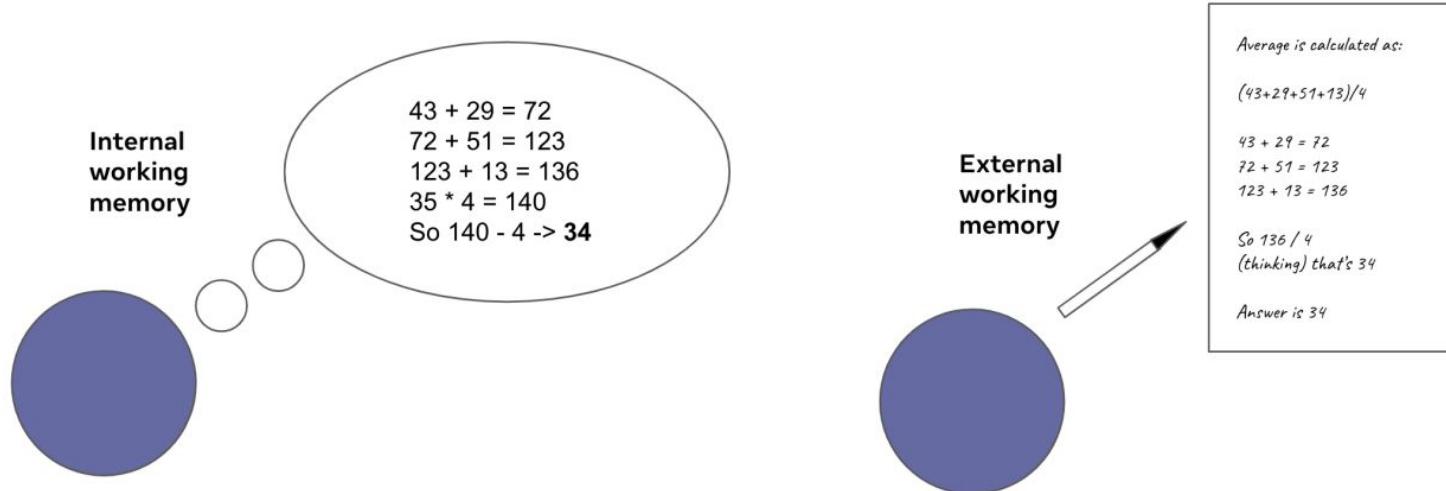
Modality	Entity	Sequence	
Text	Abell 370	Abell 370 is a cluster...	
LATEX	Schwarzschild radius	$r_s = \frac{2GM}{c^2}$	$r_s = \frac{2GM}{c^2}$
Code	Transformer	class Transformer(nn.Module)	
SMILES	Glycine	C(C(=O)O)N	
AA Sequence	Collagen $\alpha$ -1(II) chain	MIRLGAPQTL...	
DNA Sequence	Human genome	CGGTACCCCTC...	

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

# Galactica

1. **Citations:** we wrap citations with special reference tokens [START\_REF] and [END\_REF].
2. **Step-by-Step Reasoning:** we wrap step-by-step reasoning with a working memory token <work>, mimicking an internal working memory context.
3. **Mathematics:** for mathematical content, with or without LaTeX, we split ASCII operations into individual characters. Parentheses are treated like digits. The rest of the operations allow for unsplit repetitions. Operation characters are ! "#\$%&'\*+, -./: ;<=>?\^\_`| and parentheses are () [] {}.
4. **Numbers:** we split digits into individual tokens. For example 737612.62 -> 7,3,7,6,1,2,.,6,2.
5. **SMILES formula:** we wrap sequences with [START\_SMILES] and [END\_SMILES] and apply character-based tokenization. Similarly we use [START\_I\_SMILES] and [END\_I\_SMILES] where isomeric SMILES is denoted. For example, C(C(=O)O)N → C,(,C,(,=,0,),0,),N.
6. **Amino acid sequences:** we wrap sequences with [START\_AMINO] and [END\_AMINO] and apply character-based tokenization, treating each amino acid character as a single token. For example, MIRLGAPQTL -> M,I,R,L,G,A,P,Q,T,L.
7. **DNA sequences:** we also apply a character-based tokenization, treating each nucleotide base as a token, where the start tokens are [START\_DNA] and [END\_DNA]. For example, CGGTACCCCTC -> C, G, G, T, A, C, C, C, T, C.

# Galactica



**Figure 2:** Given a task like "What is the average of 43, 29, 51, 13?" a human can use internal or external working memory. In practice, they will use both symbiotically; meaning that working out that is written down in text is usually "missing" some steps performed internally.

# Galactica

**Question:** A needle 35 mm long rests on a water surface at 20°C. What force over and above the needle's weight is required to lift the needle from contact with the water surface?  $\sigma = 0.0728\text{m}$ .

<work>

$$\sigma = 0.0728 \text{ N/m}$$

$$\sigma = F/L$$

$$0.0728 = F/(2 \times 0.035)$$

$$F = 0.0728(2 \times 0.035)$$

calculate.py

'''

f = 0.0728\*(2\*0.035)

with open("output.txt", "w") as file:  
 file.write(str(round(f, 5)))

'''

«run: "calculate.py"»

«read: "output.txt"»

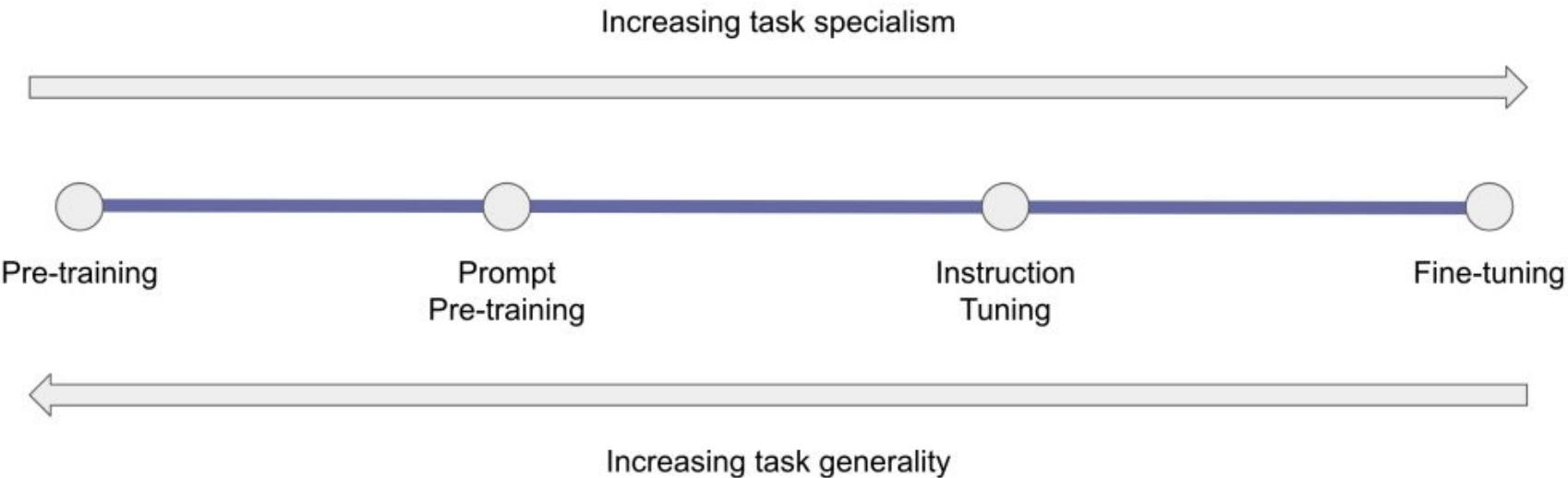
0.0051

</work>

**Answer:**  $F = 0.0051 \text{ N}$

**Figure 3: Model-Machine Symbiosis.** We show an example answer with the <work> working memory token. It performs exact steps for rearranging the equation, and when it reaches a calculation that it cannot solve reliably in a forward-pass, it writes a program, which can then be offloaded to a classical computer.

# Galactica



# Galactica

Total dataset size = 106 billion tokens			
Data source	Documents	Tokens	Token %
Papers	48 million	88 billion	83.0%
Code	2 million	7 billion	6.9%
Reference Material	8 million	7 billion	6.5%
Knowledge Bases	2 million	2 billion	2.0%
Filtered CommonCrawl	0.9 million	1 billion	1.0%
Prompts	1.3 million	0.4 billion	0.3%
Other	0.02 million	0.2 billion	0.2%

**Table 2: The Galactica Corpus.** A full breakdown of these sources is contained in the Appendix.

# GPT-1

- Transformer model (2017, Google)
- GPT-1 (2018)
- Generative Pre-Training
- 12 Layers
- 768 Dimensions
- 117M Parameters
- BPE
- BooksCorpus dataset (1B)
- Adam



# GPT-2

- medium
  - 355M
  - L = 24
  - D = 768
- Large
  - 774M
  - L = 36
  - D = 1280
- XL
  - 1.5B
  - L = 48
  - D = 1600
- BPE
- WebText (40GB)



# GPT-3

- Decode-only transformer
- Scaling up the size of language models
- 175B parameters
- Likes GPT-2
  - Sparse attention
  - larger batch sizes with a lower learning rate
- Dataset
  - CommonCrawl
  - Webtext dataset
  - books corpora
  - English-language Wikipedia
- ICL



# GPT-3



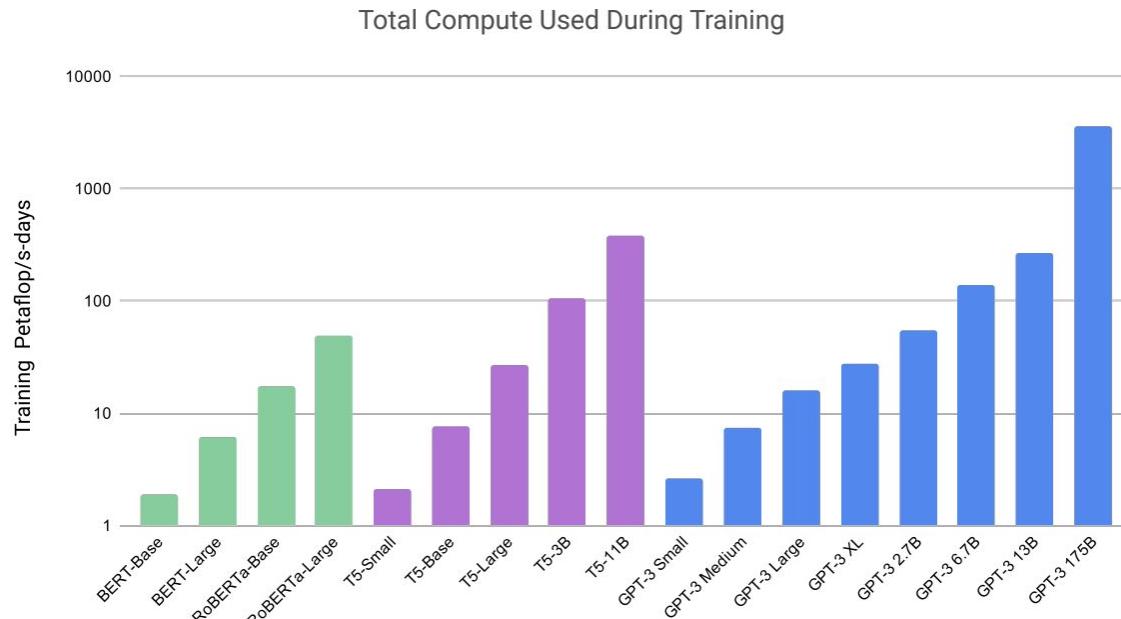
Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
May-2020	175	300B	Causal-Dec	Next Token	-	Learned	Adam

# GPT-3

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

**Table 2.2: Datasets used to train GPT-3.** “Weight in training mix” refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

# GPT-3



**Figure 2.2: Total compute used during training.** Based on the analysis in Scaling Laws For Neural Language Models [KMH<sup>+</sup>20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute during pre-training. Methodology for these calculations can be found in Appendix D.

# GPT-3

Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

**Table 2.1:** Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

# GPT-3.5

- Training on code data
  - Code-based GPT model (code-davinci-002)
  - Reasoning
- Human alignment
  - RLHF
  - InstructGPT (January 2022)
- Chat GPT
  - InstructGPT
  - Dialogue



# GPT-3.5

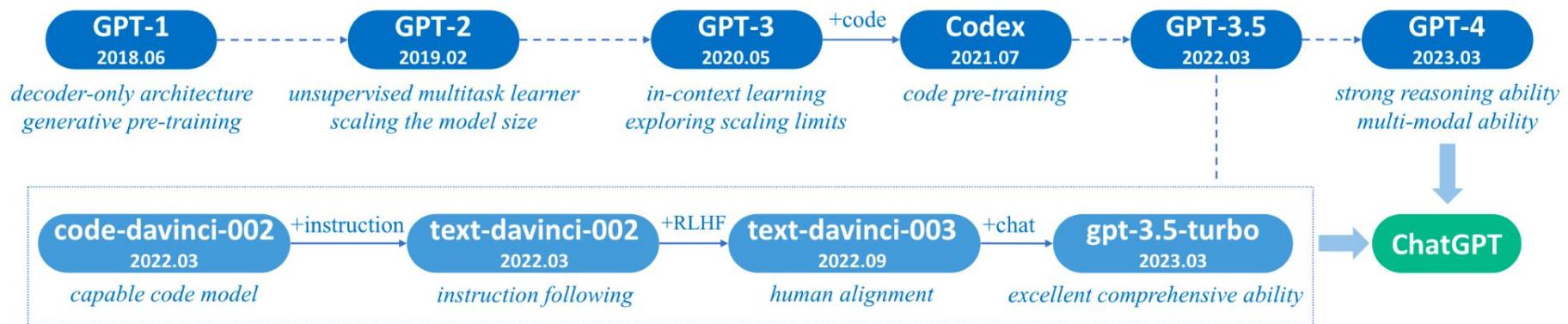


Fig. 3: A brief illustration for the technical evolution of GPT-series models. We plot this figure mainly based on the papers, blog articles and official APIs from OpenAI. Here, *solid lines* denote that there exists an explicit evidence (e.g., the official statement that a new model is developed based on a base model) on the evolution path between two models, while *dashed lines* denote a relatively weaker evolution relation.

# GPT-4

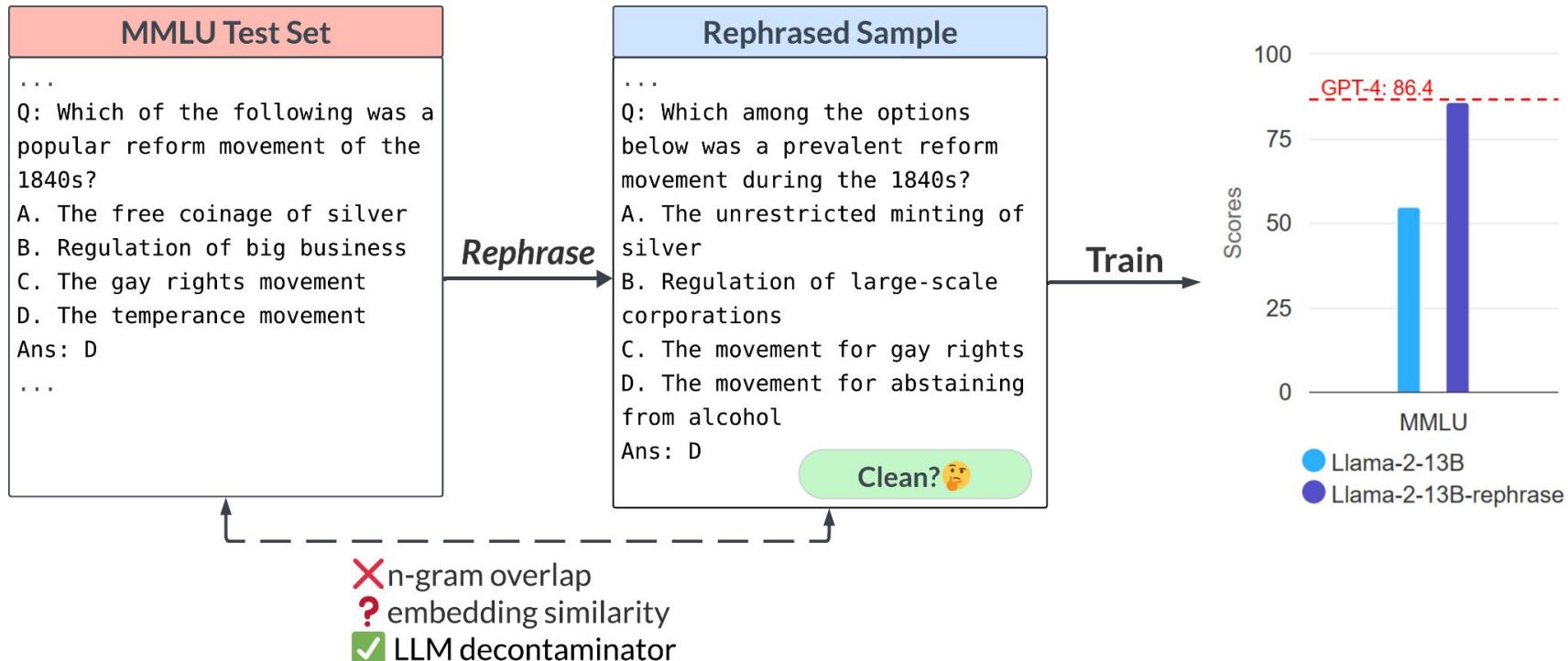
- GPT-4
  - Multimodal
  - Text, Image input-Text output
  - Next token prediction
  - RLHF
  - System prompt
  - Prompt
  - The model's capabilities seem to come primarily from the pre-training process—RLHF does not improve exam performance
  - Data contamination



# Data contamination

- n-gram overlap and embedding similarity search
- Rephrased Samples
  - Paraphrasing
  - Translation
- If such samples are included in the training set, a 13B model can reach drastically high performance (MMLU 85.9)

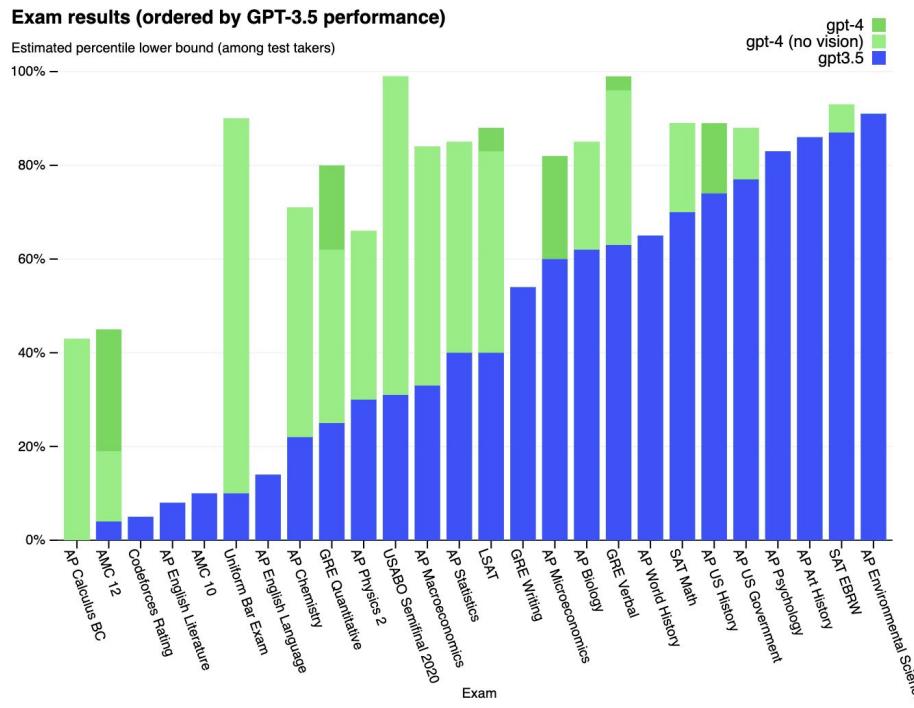
# Data contamination



# GPT-4

Simulated exams	GPT-4 estimated percentile	GPT-4 (no vision) estimated percentile	GPT-3.5 estimated percentile
Uniform Bar Exam (MBE+MEE+MPT) <sup>1</sup>	298/400 ~90th	298/400 ~90th	213/400 ~10th
LSAT	163 ~88th	161 ~83rd	149 ~40th
SAT Evidence-Based Reading & Writing	710/800 ~93rd	710/800 ~93rd	670/800 ~87th
SAT Math	700/800 ~89th	690/800 ~89th	590/800 ~70th
Graduate Record Examination (GRE) Quantitative	163/170 ~80th	157/170 ~62nd	147/170 ~25th
Graduate Record Examination (GRE) Verbal	169/170 ~99th	165/170 ~96th	154/170 ~63rd
Graduate Record Examination (GRE) Writing	4/6 ~54th	4/6 ~54th	4/6 ~54th

# GPT-4



# GPT-4

User

What is funny about this image? Describe it panel by panel.



Source: [hmmm \(Reddit\)](#)

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

# Codex



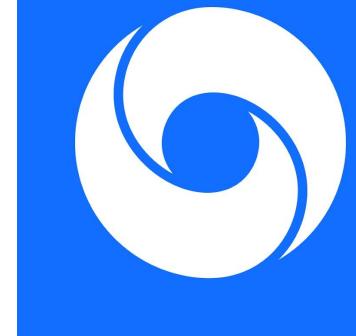
- Fine-tuned GPT
- Dataset
  - Public Python Github repositories
  - 54 million public software repositories hosted on GitHub containing python files under 1MB
- Github Copilot
- Codex generates 100 versions of a program by repetitive sampling

# Codex



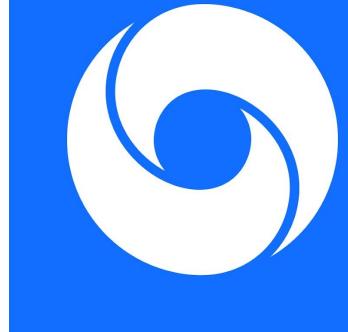
Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Jul-2021	12	100B	Causal-Dec	Next Token	BPE	Learned	Adam

# Gopher



- DeepMind
- Dataset (MassiveText)
  - MassiveWeb
  - Books
  - C4
  - News
  - GitHub
  - Wikipedia
- Models ranges from 44M to 280B parameters in size
- The 280B model beats GPT-3 on 81% of the evaluated tasks

# Gopher



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Dec-2021	280	300B	Causal-Dec	Next Token	SentencePiece	Relative	Adam

# GLaM

- Generalist Language Model
- Mixture-of-experts (MoE)
- Decoder-only Transformer
- $7\times$  larger than GPT-3
- 1.2T parameters (Activates 97B parameters)
- The largest GLaM (64B/64E) model achieves better overall results while consuming only one-third of GPT-3's training energy

# GLaM

Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Dec-2021	1200	600B	MoE-Dec	Next Token	SentencePiece	Relative	Adafactor

# MT-NLG

- Microsoft and NVIDIA
- Causal decoder
- Dataset
  - Common Crawl and Books3
  - OpenWebText2
  - Stack Exchange
  - PubMed Abstracts
  - Wikipedia
  - PG-19
  - BookCorpus2
  - NIH ExPorter
  - PileCC
  - CC-Stories
  - RealNews



# MT-NLG

- 530B model ( $3 \times$  GPT-3)
- This model beats GPT-3 on a number of evaluations
- Training
  - 8-way tensor slicing by Megatron for memory efficiency
  - 35-way pipeline parallelism using DeepSpeed

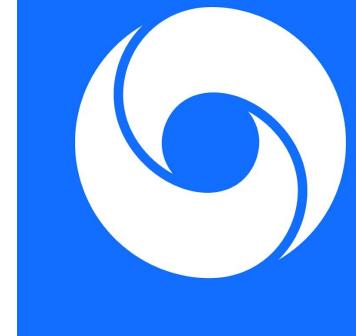


# MT-NLG



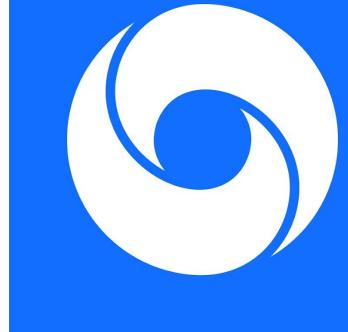
Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Jan-2022	530	270B	Causal-Dec	Next Token	BPE	Learned	Adam

# AlphaCode



- DeepMind
- Encoder-decoder transformer
- From 300M to 41B parameters
- Competition-level code generation
- Multi-query attention
- Dataset
  - Selected GitHub repositories
  - CodeContests
  - Codeforces
  - Description2Code
  - CodeNet
- Fine-tuned on a new competitive programming dataset named CodeContests
- Ranked at top 54.3% among over 5000 competitors

# AlphaCode



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Feb-2022	41	967B	Causal-Dec	Next Token	SentencePiece	-	AdamW

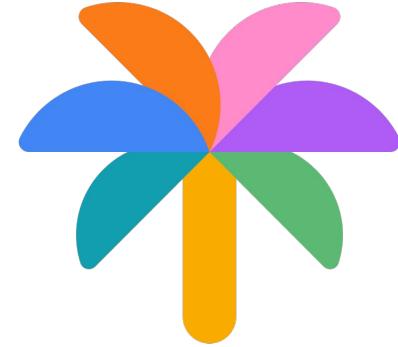
# Chinchilla

- Causal decoder
- MassiveText
- Gopher architecture
  - AdamW
- 400 language models
  - 70 million to over 16 billion parameters
  - 5 to 500 billion tokens

# Chinchilla

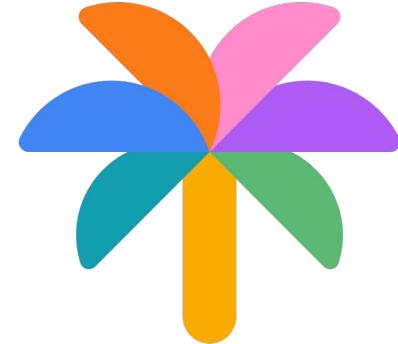
Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Mar-2022	70	1.4T	Causal-Dec	Next Token	SentencePiece NFKC	Relative	AdamW

# PaLM



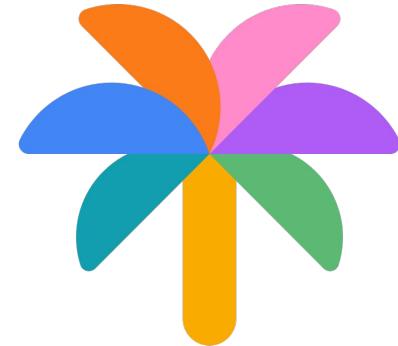
- Google
- Causal decoder
- Dataset
  - 780B tokens
  - Webpages
  - Books
  - Wikipedia
  - News
  - Articles
  - Source code
  - Social media conversations

# PaLM



- Parallel attention and feed-forward layers
- SwiGLU
- RoPE
- Multi-query attention
- Lossless vocab
- shared input-output embeddings
- 540B parameters (Dense model)
- PaLM memorizes around 2.4% of the training data at the 540B model scale

# PaLM



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Apr-2022	540	780B	Causal-Dec	Next Token	SentencePiece	RoPE	Adafactor

# Parallel attention and feed-forward layers

- Standard serialized formulation

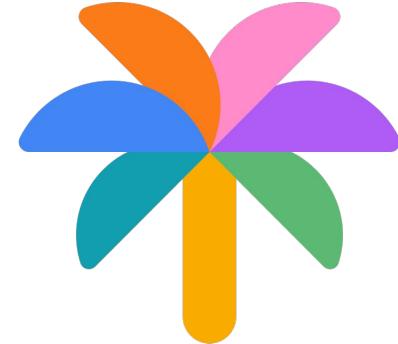
$$y = x + \text{MLP}(\text{LayerNorm}(x + \text{Attention}(\text{LayerNorm}(x))))$$

- Parallel formulation in each Transformer block

$$y = x + \text{MLP}(\text{LayerNorm}(x)) + \text{Attention}(\text{LayerNorm}(x))$$

- 15% faster training speed at large scales
- Small quality degradation at 8B scale but no quality degradation at 62B scale

# PaLM

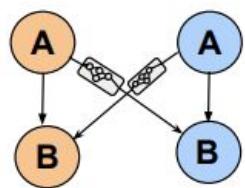


- 6144 TPU chips
- Two TPU v4 pod with 3072 chips

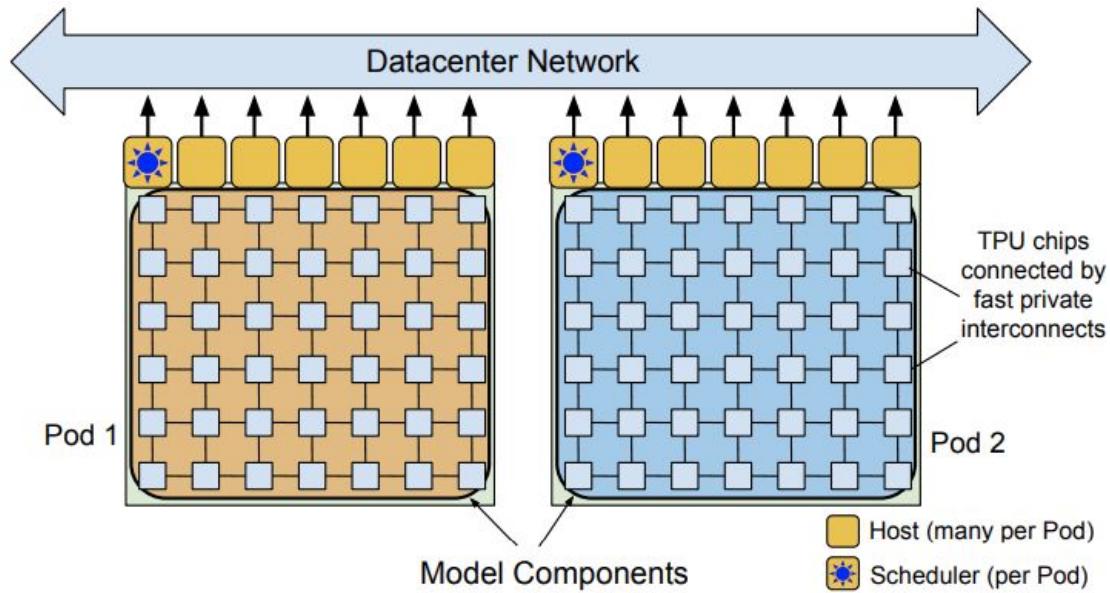
Model	# of Parameters (in billions)	Accelerator chips	Model FLOPS utilization
GPT-3	175B	V100	21.3%
Gopher	280B	4096 TPU v3	32.5%
Megatron-Turing NLG	530B	2240 A100	30.2%
PaLM	540B	6144 TPU v4	46.2%

Table 3: Model FLOPs utilization of PaLM and prior large models. PaLM achieves a notably high MFU because of several optimizations across the model, compiler, and parallelism strategy. The corresponding hardware FLOPs utilization of PaLM is 57.8%. Details of the calculation are in Appendix B.

# PaLM



- Cross-pod transfer gradients
- A** Compute gradients (Forward+backward pass)
- B** Apply gradients
- Pod 1
- Pod 2



# PaLM

---

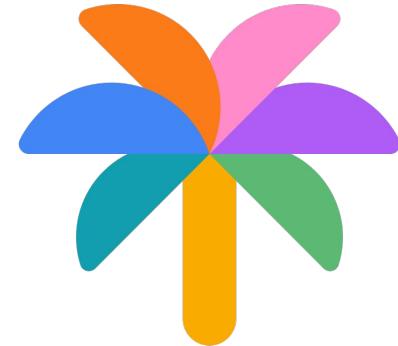
Total dataset size = 780 billion tokens

---

Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

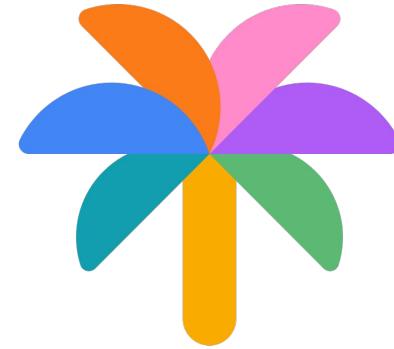
---

# U-PaLM



- Trains PaLM for 0.1% additional compute with UL2 objective
- Same dataset
- Non-causal decoder PaLM
  - Employing 50% sequential denoising, 25% regular denoising, and 25% extreme denoising loss functions

# U-PaLM



task	task / reasoning type	PaLM 540B	U-PaLM 540B
<code>navigate</code>	arithmetic, logical	55.3	<b>67.0 (+21.2%)</b>
<code>strategyqa</code>	multi-step	73.9	<b>78.3 (+6.0%)</b>
<code>crass_ai</code>	commonsense	97.7	<b>100 (+2.4%)</b>
<code>logical_sequence</code>	commonsense	<b>92.3</b>	86.5 (-6.7%)
<code>vitaminc_fact_verification</code>	contextual, commonsense	70.2	<b>73.9 (+5.3%)</b>
<code>understanding_fables</code>	commonsense	75.7	<b>78.4 (+3.6%)</b>
<code>identify_odd_metaphor</code>	analogical	87.2	<b>87.5 (+0.3%)</b>
<code>hyperbaton</code>	contextual QA	54.2	<b>59.9 (+10.5%)</b>
<code>causal_judgment</code>	causal and commonsense	65.3	<b>68.4 (+4.7 %)</b>
<code>english_proverbs</code>	commonsense, contextual QA	<b>91.2</b>	87.5 (-4.2%)
<code>geometric_shapes</code>	algorithmic, visual	44.0	<b>49.3 (+12.0%)</b>
<code>physics_questions</code>	logical, physics, math	7.6	<b>12.5 (+64.5%)</b>
<code>snarks</code>	commonsense	69.1	<b>86.1 (+24.6%)</b>
<code>analogical_similarity</code>	analogical	36.5	<b>37.5 (+2.7%)</b>
<code>international_phonetic_alphabet_nli</code>	reading comprehension	65.9	<b>68.0 (+3.2%)</b>
<code>movie_dialog_same_or_different</code>	commonsense, reading compre.	64.8	<b>68.8 (+6.2%)</b>
<code>timedial</code>	commonsense, logical	78.3	<b>81.2 (+3.7%)</b>
<code>question_selection</code>	reading comprehension	54.8	<b>59.8 (+9.1%)</b>
<code>logical_fallacy_detection</code>	logical reasoning	80.3	<b>81.4 (+1.4%)</b>
<code>unit_interpretation</code>	arithmetic, logical	47.0	<b>51.0 (+8.5%)</b>
<code>language_identification</code>	multilingual	36.0	<b>38.9 (+8.1%)</b>
average (21 tasks)	-	64.3	<b>67.7 (+5.3%)</b>

Table 1: List of challenging tasks in the BigBench emergent suite (BBES) and corresponding scores of PaLM 540B and U-PaLM 540B. All results are reported with standard 5-shot prompting.

# UL2

- Mixture of denoisers
- 20B model

UL2

# UL2

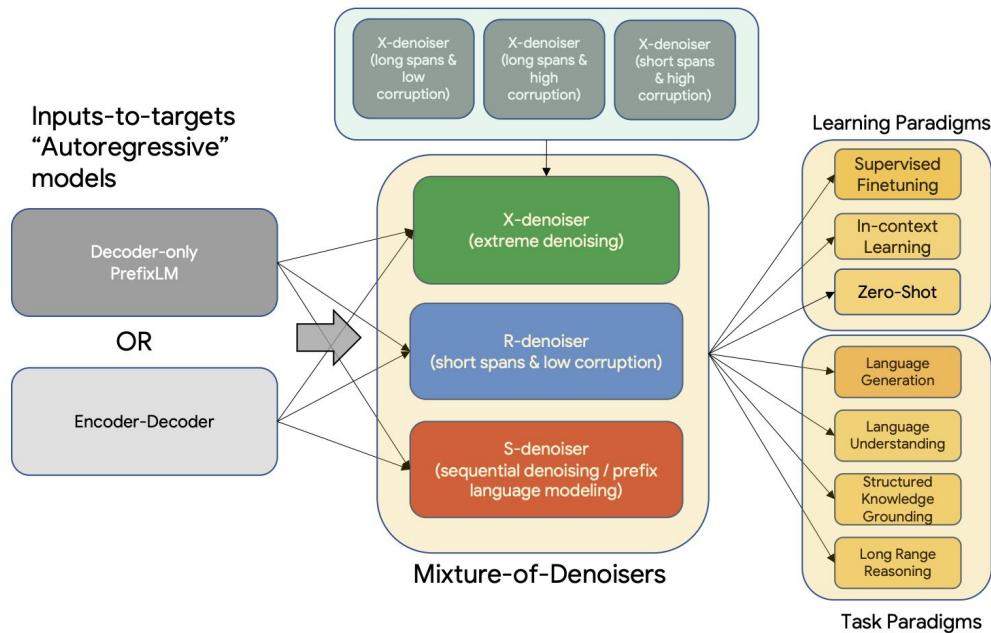


Figure 2: An overview of UL2 pretraining paradigm. UL2 proposes a new pretraining objective that works well on a diverse suite of downstream tasks.

# UL2

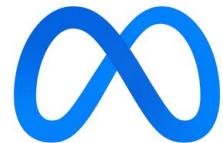
Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
May-2022	20	1T	Enc-Dec	MoD	SentencePiece	-	-

# LLaMA



- Decoder-only transformer
- A set of foundation language models varying from 7B to 65B parameters
- LLaMA-13B outperforms GPT-3 on most benchmarks, despite being 10× smaller
- 65B-parameter model is also competitive with the best large language models such as Chinchilla or PaLM-540B
- BPE
- Pre-normalization [GPT-3]
- SwiGLU [PaLM]
- Rotary Embeddings [GPTNeo]
- AdamW

# LLaMA



Release Time	Size (B)	Tokens	Category	Objective	Tokenizer	PE	Optimizer
Feb-2023	65	1.4T	Causal-Dec	Next Token	BPE	RoPE	-



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

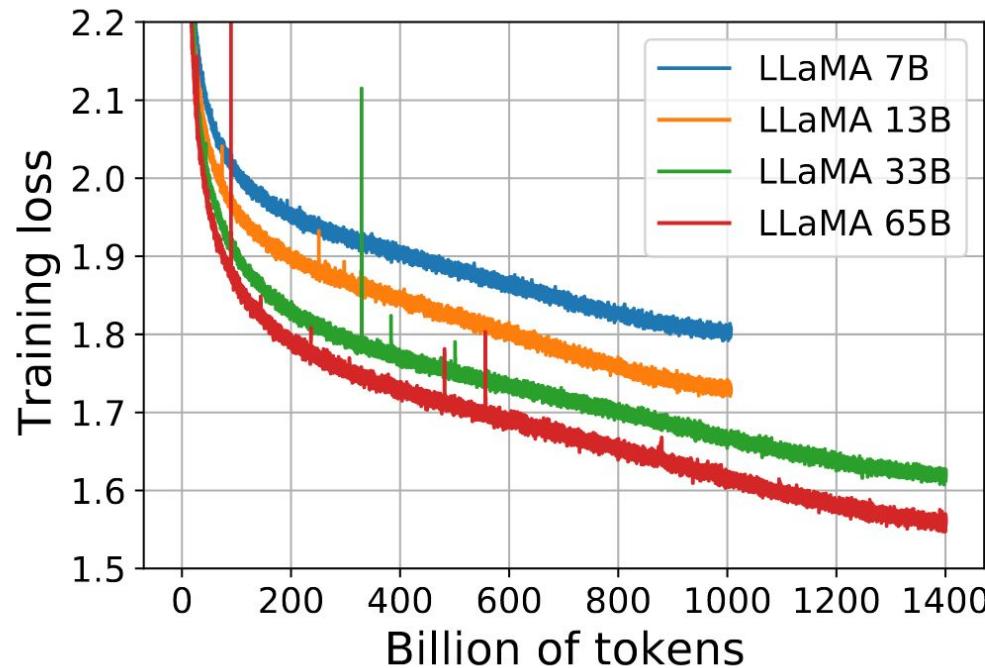
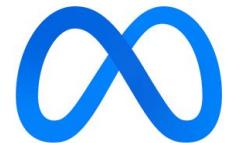
Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.



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



# LLaMA

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	<b>88.0</b>	82.3	-	83.4	<b>81.1</b>	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	<b>80.0</b>	<b>57.8</b>	58.6
	65B	85.3	<b>82.8</b>	<b>52.3</b>	<b>84.2</b>	77.0	78.9	56.0	<b>60.2</b>

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

# LLaMA

		0-shot	1-shot	5-shot	64-shot
GPT-3	175B	14.6	23.0	-	29.9
Gopher	280B	10.1	-	24.5	28.2
Chinchilla	70B	16.6	-	31.5	35.5
PaLM	8B	8.4	10.6	-	14.6
	62B	18.1	26.5	-	27.6
	540B	21.2	29.3	-	39.6
LLaMA	7B	16.8	18.7	22.0	26.1
	13B	20.1	23.4	28.1	31.9
	33B	<b>24.9</b>	28.3	32.9	36.0
	65B	23.8	<b>31.0</b>	<b>35.0</b>	<b>39.9</b>

Table 4: **NaturalQuestions.** Exact match performance.

		RACE-middle	RACE-high
GPT-3	175B	58.4	45.5
	8B	57.9	42.3
	62B	64.3	47.5
PaLM	540B	<b>68.1</b>	49.1
	7B	61.1	46.9
	13B	61.6	47.2
LLaMA	33B	64.1	48.3
	65B	67.9	<b>51.6</b>

Table 6: **Reading Comprehension.** Zero-shot accuracy.

# LLaMA

		0-shot	1-shot	5-shot	64-shot
Gopher	280B	43.5	-	57.0	57.2
Chinchilla	70B	55.4	-	64.1	64.6
LLaMA	7B	50.0	53.4	56.3	57.6
	13B	56.6	60.5	63.1	64.0
	33B	65.1	67.9	69.9	70.4
	65B	<b>68.2</b>	<b>71.6</b>	<b>72.6</b>	<b>73.0</b>

Table 5: **TriviaQA.** Zero-shot and few-shot exact match performance on the filtered dev set.

# LLaMA

	MATH +maj1@k		GSM8k +maj1@k	
PaLM	8B	1.5	-	4.1
	62B	4.4	-	33.0
	540B	8.8	-	56.5
Minerva	8B	14.1	25.4	16.2
	62B	27.6	43.4	52.4
	540B	<b>33.6</b>	<b>50.3</b>	<b>68.5</b>
LLaMA	7B	2.9	6.9	11.0
	13B	3.9	8.8	17.8
	33B	7.1	15.2	35.6
	65B	10.6	20.5	50.9
				69.7

Table 7: **Model performance on quantitative reasoning datasets.** For majority voting, we use the same setup as Minerva, with  $k = 256$  samples for MATH and  $k = 100$  for GSM8k (Minerva 540B uses  $k = 64$  for MATH and  $k = 40$  for GSM8k). LLaMA-65B outperforms Minerva 62B on GSM8k, although it has not been fine-tuned on mathematical data.

pass@	Params	HumanEval		MBPP	
		@1	@100	@1	@80
LaMDA	137B	14.0	47.3	14.8	62.4
PaLM	8B	3.6*	18.7*	5.0*	35.7*
PaLM	62B	15.9	46.3*	21.4	63.2*
PaLM-cont	62B	23.7	-	31.2	-
PaLM	540B	<b>26.2</b>	76.2	36.8	75.0
LLaMA	7B	10.5	36.5	17.7	56.2
	13B	15.8	52.5	22.0	64.0
	33B	21.7	70.7	30.2	73.4
	65B	23.7	<b>79.3</b>	<b>37.7</b>	<b>76.8</b>

Table 8: **Model performance for code generation.** We report the pass@ score on HumanEval and MBPP. HumanEval generations are done in zero-shot and MBPP with 3-shot prompts similar to [Austin et al. \(2021\)](#). The values marked with \* are read from figures in [Chowdhery et al. \(2022\)](#).

# LLaMA

		Humanities	STEM	Social Sciences	Other	Average
GPT-NeoX	20B	29.8	34.9	33.7	37.7	33.6
GPT-3	175B	40.8	36.7	50.4	48.8	43.9
Gopher	280B	56.2	47.4	71.9	66.1	60.0
Chinchilla	70B	63.6	54.9	79.3	<b>73.9</b>	67.5
PaLM	8B	25.6	23.8	24.1	27.8	25.4
	62B	59.5	41.9	62.7	55.8	53.7
	540B	<b>77.0</b>	<b>55.6</b>	<b>81.0</b>	69.6	<b>69.3</b>
LLaMA	7B	34.0	30.5	38.3	38.1	35.1
	13B	45.0	35.8	53.8	53.3	46.9
	33B	55.8	46.0	66.7	63.4	57.8
	65B	61.8	51.7	72.9	67.4	63.4

Table 9: **Massive Multitask Language Understanding (MMLU).** Five-shot accuracy.

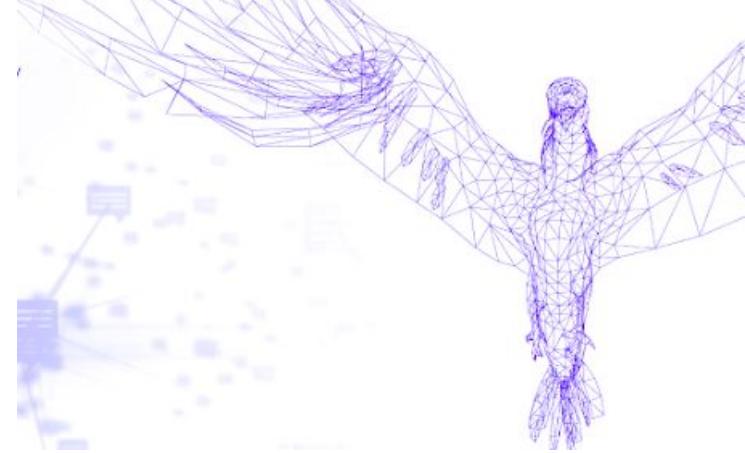
# LLaMA

OPT	30B	26.1
GLM	120B	44.8
PaLM	62B	55.1
PaLM-cont	62B	62.8
Chinchilla	70B	67.5
LLaMA	65B	63.4
OPT-IML-Max	30B	43.2
Flan-T5-XXL	11B	55.1
Flan-PaLM	62B	59.6
Flan-PaLM-cont	62B	66.1
LLaMA-I	65B	<b>68.9</b>

Table 10: **Instruction finetuning – MMLU (5-shot).**  
Comparison of models of moderate size with and without instruction finetuning on MMLU.

# Falcon

- 180B, 40B, 7.5B, 1.3B parameters
- REFINEDWEB
- Open source
- Falcon 40B
  - 1 trillion tokens
- Multilingual Falcon 40B
  - English, German, Spanish, French, Italian, Portuguese, Polish, Dutch, Romanian, Czech, and Swedish languages
- Falcon 180B
  - 3.5 trillion tokens



# Mistral

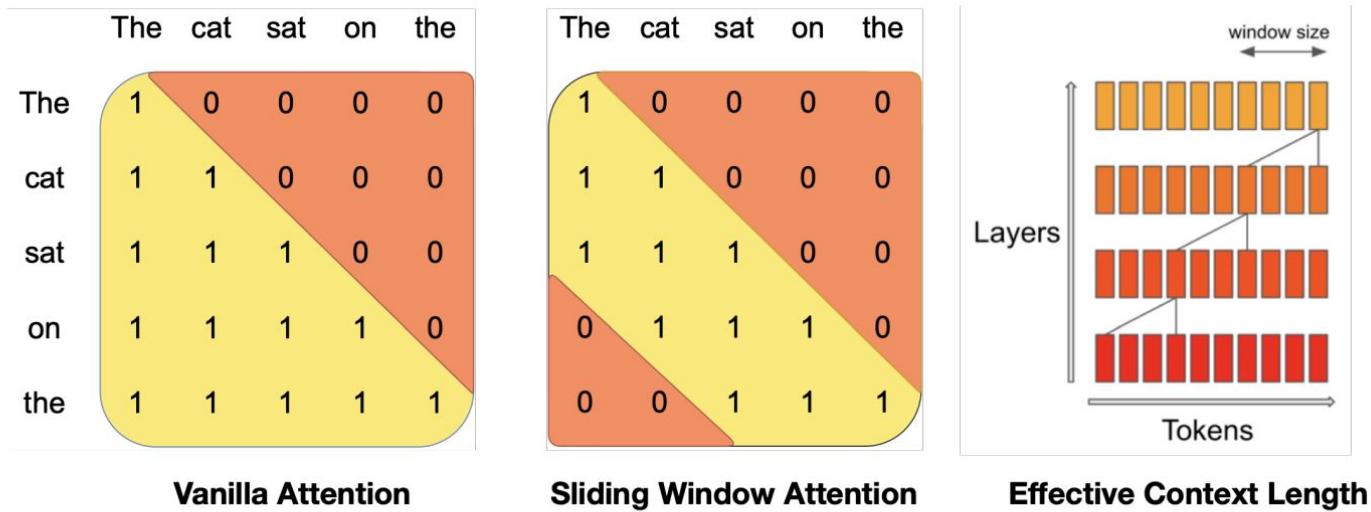
- Decoder-only transformer
- 7-billion-parameter language model
- Mistral 7B – Instruct
- Grouped-query attention (GQA)
- Sliding window attention (SWA)
- Rolling Buffer Cache



Parameter	Value
dim	4096
n_layers	32
head_dim	128
hidden_dim	14336
n_heads	32
n_kv_heads	8
window_size	4096
context_len	8192
vocab_size	32000

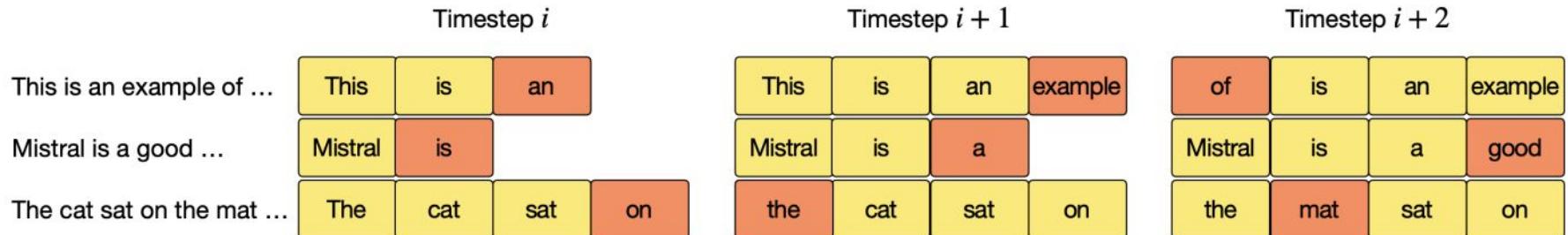
**Table 1: Model architecture.**

# Mistral



**Figure 1: Sliding Window Attention.** The number of operations in vanilla attention is quadratic in the sequence length, and the memory increases linearly with the number of tokens. At inference time, this incurs higher latency and smaller throughput due to reduced cache availability. To alleviate this issue, we use sliding window attention: each token can attend to at most  $W$  tokens from the previous layer (here,  $W = 3$ ). Note that tokens outside the sliding window still influence next word prediction. At each attention layer, information can move forward by  $W$  tokens. Hence, after  $k$  attention layers, information can move forward by up to  $k \times W$  tokens.

# Mistral



**Figure 2: Rolling buffer cache.** The cache has a fixed size of  $W = 4$ . Keys and values for position  $i$  are stored in position  $i \bmod W$  of the cache. When the position  $i$  is larger than  $W$ , past values in the cache are overwritten. The hidden state corresponding to the latest generated tokens are colored in orange.

# Mistral

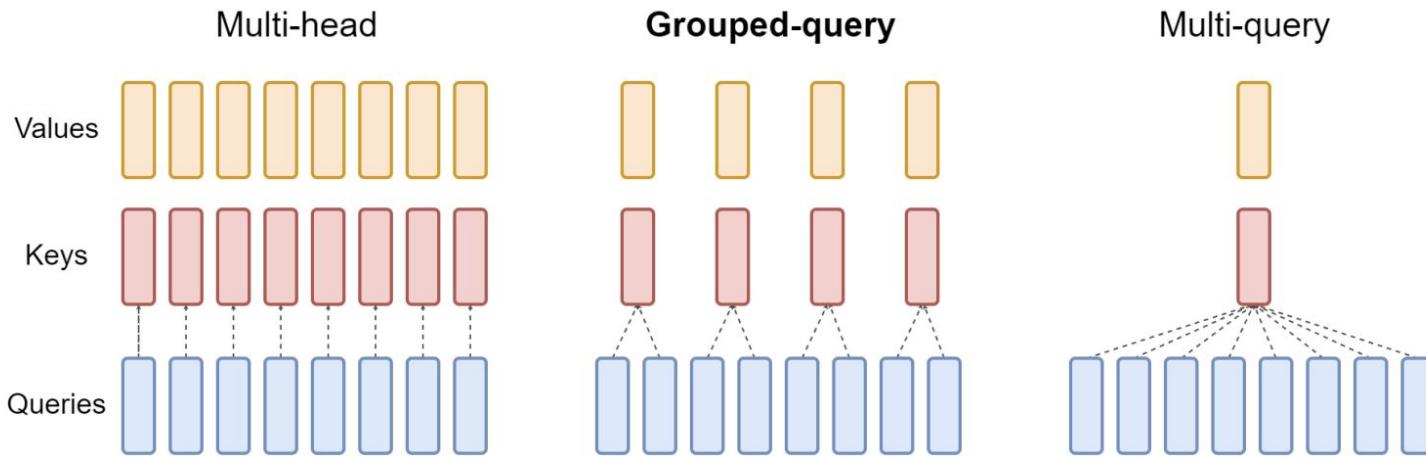
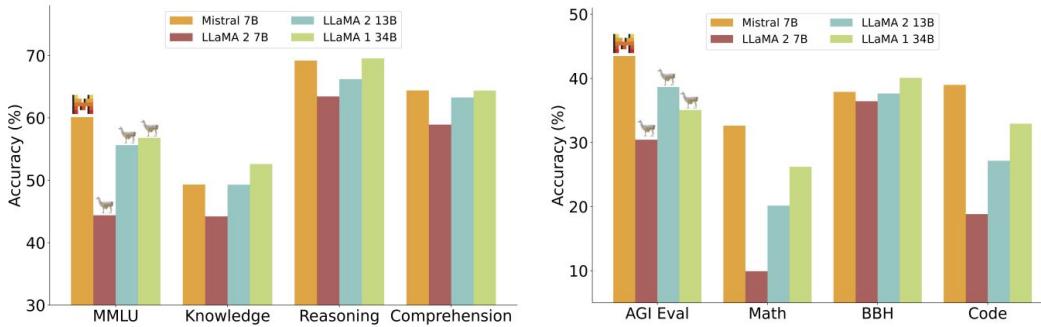


Figure 2: Overview of grouped-query method. Multi-head attention has  $H$  query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between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

# Mistral

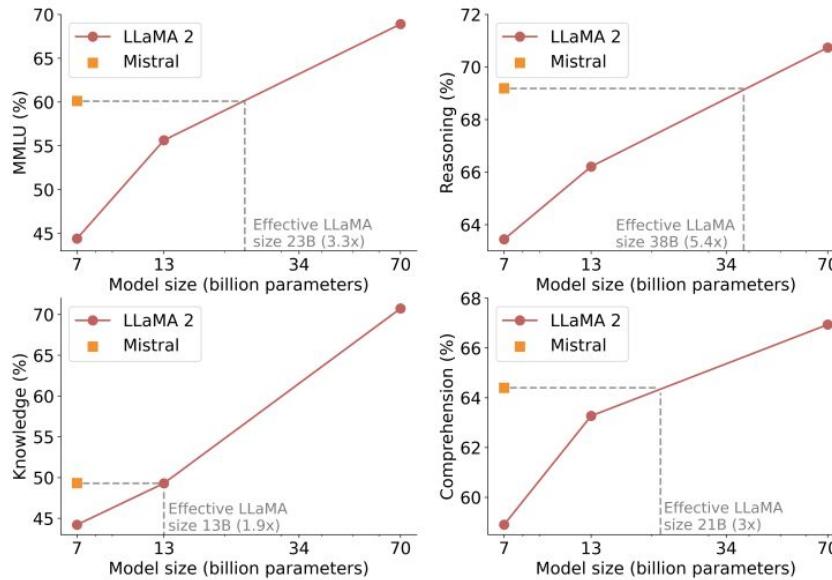


**Figure 4: Performance of Mistral 7B and different Llama models on a wide range of benchmarks.** All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mistral 7B significantly outperforms Llama 2 7B and Llama 2 13B on all benchmarks. It is also vastly superior to Llama 1 34B in mathematics, code generation, and reasoning benchmarks.

Model	Modality	MMLU	HellaSwag	WinoG	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	<b>80.7%</b>	72.9%	80.8%	75.2%	48.8%	<b>29.0%</b>	<b>69.6%</b>	18.9%	35.4%	6.0%	34.3%
Code-Llama 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	<b>31.1%</b>	<b>52.5%</b>	5.2%	20.8%
Mistral 7B	Pretrained	<b>60.1%</b>	<b>81.3%</b>	<b>75.3%</b>	<b>83.0%</b>	<b>80.0%</b>	<b>55.5%</b>	<b>28.8%</b>	<b>69.9%</b>	<b>30.5%</b>	47.5%	<b>13.1%</b>	<b>52.2%</b>

**Table 2: Comparison of Mistral 7B with Llama.** Mistral 7B outperforms Llama 2 13B on all metrics, and approaches the code performance of Code-Llama 7B without sacrificing performance on non-code benchmarks.

# Mistral



**Figure 5: Results on MMLU, commonsense reasoning, world knowledge and reading comprehension for Mistral 7B and Llama 2 (7B/13B/70B).** Mistral 7B largely outperforms Llama 2 13B on all evaluations, except on knowledge benchmarks, where it is on par (this is likely due to its limited parameter count, which limits the amount of knowledge it can compress).

# Pythia

- 16 LLMs all trained on public data seen in the exact same order and ranging in size from 70M to 12B parameters
- EleutherAI
- Decoder-only autoregressive language models
- All models were trained on the same data in the same order
- The data and intermediate checkpoints are publicly available for study
- Pile
- BPE tokenizer

# Pythia

- Flash Attention
- Rotary embeddings
- Adam
- ZeRO

# RWKV

- RNN+Large model size
- Parallelizable RNN with Transformer-level LLM Performance
- RNN problems
  - Training time
- Train like a transformer
- Inference like a RNN
- R: Receptance
- W: Weight
- K: Key
- V : Value



# RWKV

	params	LAMBADA	AVERAGE	LAMBADA	PIQA	StoryCloze16	Hellaswag	WinoGrande	arc_challenge	arc_easy	headQA	openbookQA	sciq	triviaQA	ReCoRD	COPA
	B	ppl	%	acc	acc	acc	acc_norm	acc	acc_norm	acc	acc_norm	acc_norm	acc	acc	em	acc
RWKV-4	0.17	29.33	44.13%	32.99%	65.07%	58.79%	32.26%	50.83%	24.15%	47.47%	25.78%	29.60%	77.50%	1.26%	62.03%	66.00%
Pythia	0.16	24.38	44.14%	38.97%	62.68%	58.47%	31.63%	52.01%	23.81%	45.12%	25.82%	29.20%	76.50%	1.31%	66.32%	62.00%
GPT-Neo	0.16	30.27	43.42%	37.36%	63.06%	58.26%	30.42%	50.43%	23.12%	43.73%	25.16%	26.20%	76.60%	1.18%	64.92%	64.00%
RWKV-4b	0.17	22.02	45.42%	38.56%	64.04%	59.91%	33.33%	53.20%	24.57%	48.15%	25.93%	28.80%	77.80%	1.83%	67.39%	67.00%
	params	LAMBADA	AVERAGE	LAMBADA	PIQA	StoryCloze16	Hellaswag	WinoGrande	arc_challenge	arc_easy	headQA	openbookQA	sciq	triviaQA	ReCoRD	COPA
RWKV-4	0.43	13.04	48.04%	45.16%	67.52%	63.87%	40.90%	51.14%	25.17%	52.86%	27.32%	32.40%	80.30%	2.35%	70.48%	65.00%
Pythia	0.4	11.58	48.39%	50.44%	66.70%	62.64%	39.10%	53.35%	25.77%	50.38%	25.09%	30.00%	81.50%	2.03%	75.05%	67.00%
GPT-Neo	0.4	13.88	47.25%	47.29%	65.07%	61.04%	37.64%	51.14%	25.34%	48.91%	26.00%	30.60%	81.10%	1.38%	73.79%	65.00%
RWKV-4b	0.44	10.48	49.24%	51.35%	68.06%	63.17%	42.09%	54.14%	24.66%	52.36%	27.94%	31.00%	82.20%	3.92%	74.25%	65.00%
	params	LAMBADA	AVERAGE	LAMBADA	PIQA	StoryCloze16	Hellaswag	WinoGrande	arc_challenge	arc_easy	headQA	openbookQA	sciq	triviaQA	ReCoRD	COPA
RWKV-4	1.5	7.04	53.91%	56.43%	72.36%	68.73%	52.48%	54.62%	29.44%	60.48%	27.64%	34.00%	85.00%	5.65%	76.97%	77.00%
Pythia	1.4	6.58	53.55%	60.43%	71.11%	67.66%	50.82%	56.51%	28.58%	57.74%	27.02%	30.80%	85.50%	5.52%	81.43%	73.00%
GPT-Neo	1.4	7.50	52.64%	57.25%	71.16%	67.72%	48.94%	54.93%	25.85%	56.19%	27.86%	33.60%	86.00%	5.24%	80.62%	69.00%
RWKV-4b	1.5	5.82	55.24%	62.35%	72.52%	68.89%	54.32%	57.70%	29.27%	60.44%	28.92%	33.80%	85.10%	7.03%	81.74%	76.00%
	params	LAMBADA	AVERAGE	LAMBADA	PIQA	StoryCloze16	Hellaswag	WinoGrande	arc_challenge	arc_easy	headQA	openbookQA	sciq	triviaQA	ReCoRD	COPA
RWKV-4	3	5.24	57.52%	63.94%	73.72%	70.28%	59.63%	59.43%	31.83%	64.27%	28.74%	37.60%	85.70%	11.07%	80.56%	81.00%
RWKV-4	3, ctx4k	5.25	57.93%	63.96%	74.16%	70.71%	59.89%	59.59%	33.11%	65.19%	28.45%	37.00%	86.50%	11.68%	80.87%	82.00%
Pythia	2.8	4.93	57.64%	65.36%	73.83%	70.71%	59.46%	61.25%	32.25%	62.84%	28.96%	35.20%	87.70%	9.63%	85.10%	77.00%
GPT-Neo	2.8	5.63	55.92%	62.22%	72.14%	69.54%	55.82%	57.62%	30.20%	61.07%	27.17%	33.20%	89.30%	4.82%	83.80%	80.00%
RWKV-4b	3	4.82	58.31%	65.83%	73.94%	72.31%	60.90%	61.88%	32.85%	62.37%	28.48%	36.80%	86.60%	12.53%	83.57%	80.00%
	params	LAMBADA	AVERAGE	LAMBADA	PIQA	StoryCloze16	Hellaswag	WinoGrande	arc_challenge	arc_easy	headQA	openbookQA	sciq	triviaQA	ReCoRD	COPA
RWKV-4	7.4	4.38	61.20%	67.18%	76.06%	73.44%	65.51%	61.01%	37.46%	67.80%	31.22%	40.20%	88.80%	18.30%	83.68%	85.00%
Pythia	6.9	4.30	60.44%	67.98%	74.54%	72.96%	63.92%	61.01%	35.07%	66.79%	28.59%	38.00%	90.00%	15.42%	86.44%	85.00%
GPT-J	6.1	4.10	61.34%	68.31%	75.41%	74.02%	66.25%	64.09%	36.60%	66.92%	28.67%	38.20%	91.50%	16.74%	87.71%	83.00%
	params	LAMBADA	AVERAGE	LAMBADA	PIQA	StoryCloze16	Hellaswag	WinoGrande	arc_challenge	arc_easy	headQA	openbookQA	sciq	triviaQA	ReCoRD	COPA
RWKV-4	14.2	3.86	63.71%	70.83%	77.48%	76.06%	70.65%	63.85%	38.99%	70.24%	32.64%	41.80%	90.40%	24.58%	85.67%	85.00%
GPT-level	14.2	3.81	63.11%	70.94%	76.49%	74.97%	68.72%	65.14%	37.99%	70.77%	31.03%	39.27%	92.20%	22.37%	87.89%	82.66%
Pythia	11.8	3.89	62.38%	70.44%	75.90%	74.40%	67.38%	64.72%	36.77%	69.82%	30.74%	38.80%	91.80%	20.57%	87.58%	82.00%
GPT-NeoX	20.6	3.64	64.58%	71.94%	77.69%	76.11%	71.42%	65.98%	40.44%	72.69%	31.62%	40.20%	93.00%	25.99%	88.52%	84.00%

# RWKV

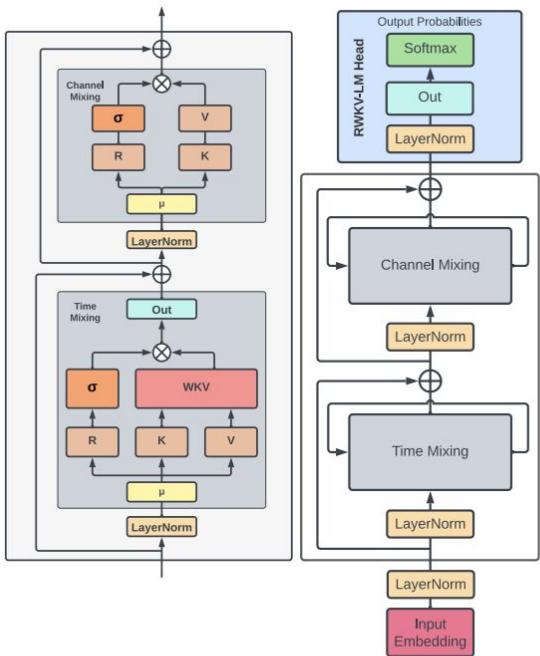


Figure 2: RWKV block elements (left) and RWKV residual block with a final head for language modeling (right) architectures.

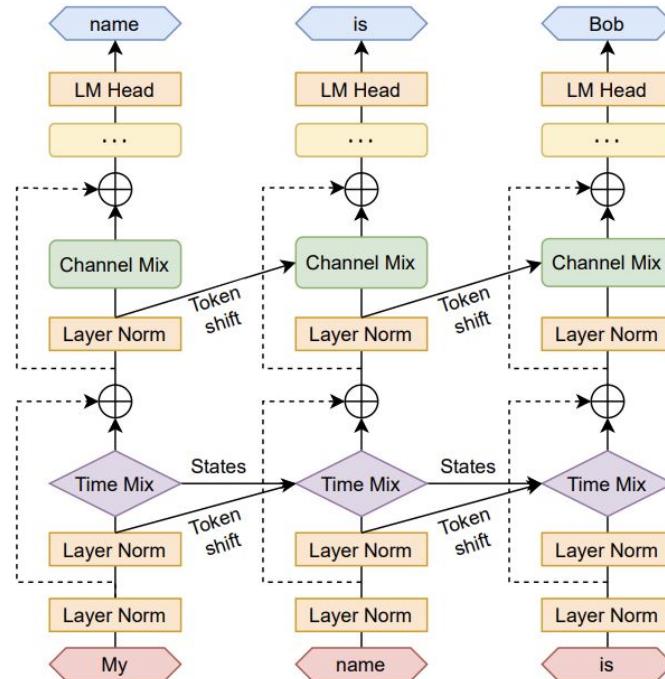


Figure 3: RWKV architecture for language modelling.