LLaMA: Open and Efficient Foundation Language Models

Hugo Touvron*, Thibaut Lavril*, Gautier Izacard*, Xavier Martinet
Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal
Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin
Edouard Grave*, Guillaume Lample*

Meta AI

- Published in 2023
- ~12.5k citations

Why LLaMa?

- Previous works (GPT-3, PaLM, Scaling Laws) suggest more parameters lead to better performance
- Chinchilla (DeepMind) suggests best performance given a compute budget is achieved by training smaller models on more data
- Main focus of current LLM work is cheap/efficient training
  - But what about the inference budget?
- Authors suggest best model does not have fastest train time, but rather fastest inference time

LLaMa’s Contribution

- Smaller models trained on more data
  - Chinchilla’s 10B model trained on 200B tokens
  - LLaMa’s 7B model trained on 1T tokens with continuing improvement vs Chinchilla
- Train on “more tokens than what is typically used”
- Fast inference capable of running on a single GPU
- Only train on publicly available datasets
  - Compared to OpenAI, Google (private datasets)

Pre-Training Data

- CommonCrawl
  - 2017-2020; removal of non-English pages and filter low quality content via n-gram model
- C4
  - Similar to CommonCrawl; main difference in quality filtering (heuristic based)
- Github
  - Google BigQuery dataset; heuristic filtering; open source licensing
- Wikipedia
  - June-August 2022 covering 20 languages
- Books
  - Gutenberg Project and Books3 section of ThePile
- ArXiv
  - Processed latex files
- StackExchange
  - Data from the 28 largest websites; scored answers from highest to lowest

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.

Tokenizer

- Uses BPE algorithm using SentencePiece implementation
- All numbers are split into individual digits
  - Numbers i.r.l. appear in different variations
  - Allows for learning reusable patterns in numeric sequences
- If tokenizer encounters unknown character, encodes it as raw UTF-8 bytes
  - Ability to handle any possible input even if character was not seen in training
- Training Data Statistics: 1.4T tokens
- Mostly, each token is used only once during training

Transformer Architecture Modifications

- Pre-normalization (GPT-3)
  - Normalize input of each transformer sublayer instead of normalized output
  - RMSNorm function

$$y_i = \frac{x_i}{\mathrm{RMS}(x)} * \gamma_i, \quad \text{where} \quad \mathrm{RMS}(x) = \sqrt{\epsilon + \frac{1}{n}\sum_{i=1}^{n} x_i^2}$$

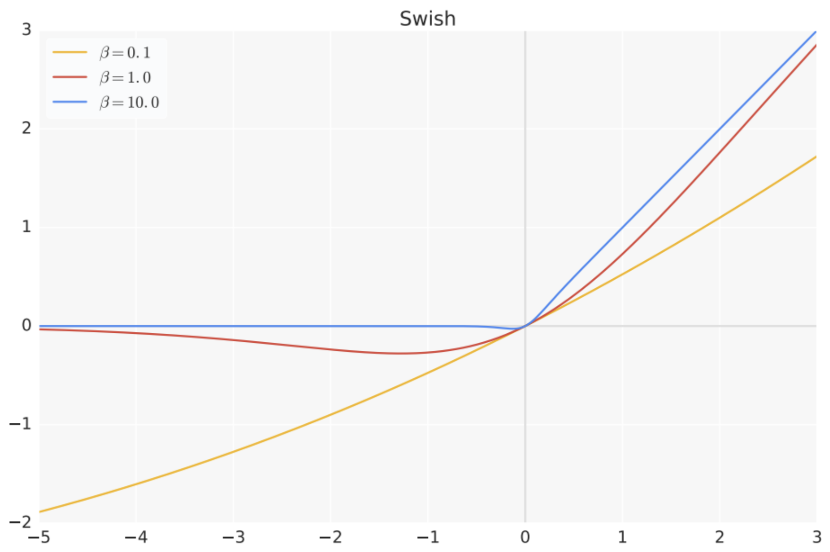
Transformer Architecture Modifications

- SwiGLU activation function (PaLM)
  - Swish + Gated Linear Units (GLU)
    - GLU - a neural network layer defined as the componentwise product of two linear transformations of the input, one of which is sigmoid-activated (*GLU Variants Improve Transformer, 2020*)

$$\mathrm{GLU}(x, W, V, b, c) = \sigma(xW + b) \otimes (xV + c)$$

$$\mathrm{SwiGLU}(x, W, V, b, c, \beta) = \mathrm{Swish}_\beta(xW + b) \otimes (xV + c)$$

  - Replaces ReLU



Transformer Architecture Modifications

- Rotary Embeddings (GPTNeo)
  - Remove absolute positional embeddings and add rotary positional embeddings (*Roformer: Enhanced transformer with rotary position embedding, Neurocomputing 2024*)

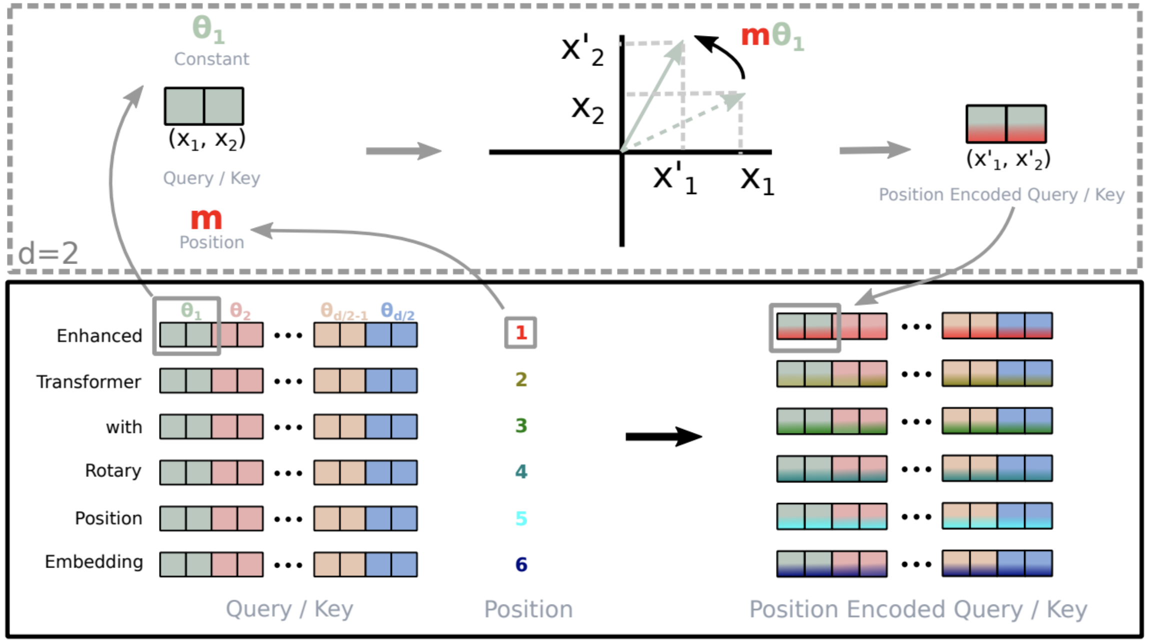


Figure 1: Implementation of Rotary Position Embedding(RoPE).

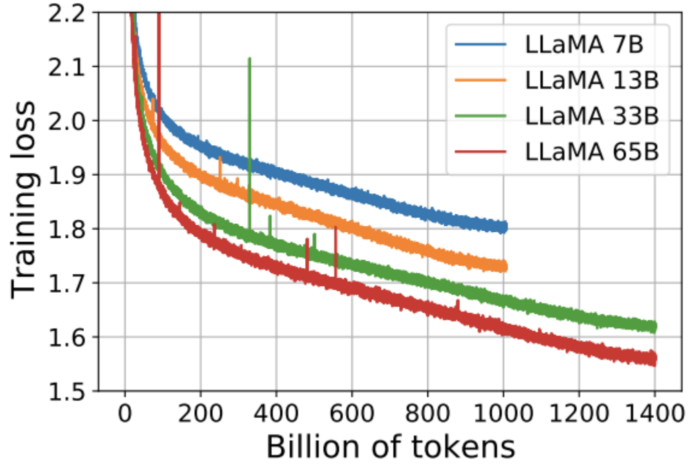
Optimization

- AdamW optimizer w/ $\beta_1 = 0.9, \beta_2 = 0.95$
  - Variant of Adam that decouples weight decay from the gradient update
- Cosine Annealing learning rate scheduling
  - Final learning rate is 10% of maximal learning rate
- Weight decay of 0.1
- Gradient clipping of 1.0
- 2,000 warmup steps

Summarization of Training

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.**



Efficient Implementation

- Reduce memory usage and runtime of causal multi-head attention
  - See xformers library (https://github.com/facebookresearch/xformers)
- Achieved by "not storing the attention weights and not computing the key/query scores that are masked due to causal nature of the language modeling task"
- Save expensive activations (e.g., outputs of linear layers) via a manual implementation of backward function instead of Pytorch autograd
- TL;DR – Highly engineering attention computation with trade-off between speed and memory
  - Training speedups come at expense of more memory

Unrealistic Compute Requirements

- Training of largest 65B-parameter model
  - Processes 380 tokens/sec/GPU
  - Uses 2048 A100 GPUs (80GB VRAM each)
  - Training over 1.4T tokens takes ~21 days

Main Results

- Freeform text generation
  - Model generates an answer
- Multiple Choice
  - Model ranks proposed answers
- Zero-shot
  - Provide textual description of task and a test example
- Few-shot
  - Provide few examples of the task (between 1 and 64) and a test example

Common Sense Reasoning

		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	**88.0**	82.3	-	83.4	**81.1**	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	**80.0**	**57.8**	58.6
	65B	85.3	**82.8**	**52.3**	**84.2**	77.0	78.9	56.0	**60.2**

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

- Cloze (fill in the blank) and Winograd (pronoun ambiguity) tasks
- LlaMa 65B outperform Chinchilla, PaLM on nearly all benchmarks.
- LLaMa 13B outperforms GPT-3 on nearly all benchmarks despite being 10x smaller

Closed-book Question Answering

- Exact Match Performance – Model does not have access to documents that contain evidence to answer question

		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	**24.9**	28.3	32.9	36.0
	65B	23.8	**31.0**	**35.0**	**39.9**

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

		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	**68.2**	**71.6**	**72.6**	**73.0**

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

- LLaMa 65B achieves SOTA performance in zero-shot and few-shot
- LLaMa 13B competitive with GPT-3 and Chinchilla despite be 5-10x smaller
  - Runs on single V100 GPU for inference

Reading Comprehension

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

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

- Dataset composed of English reading comprehension exams designed for middle and high school Chinese students (multiple choice)

Mathematical Reasoning

- MATH
  - 12k middle school and high school math problems
- GSM8k
  - Middle school math problems
- Minerva
  - Series of PaLM models finetuned on 38.5B tokens from math resources
- maj1@k – Majority voting over k samples

		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	28.4
	62B	27.6	43.4	52.4	68.5
	540B	**33.6**	**50.3**	**68.5**	**78.5**
LLaMA	7B	2.9	6.9	11.0	18.1
	13B	3.9	8.8	17.8	29.3
	33B	7.1	15.2	35.6	53.1
	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 and $k = 40$ for GSM8k). LLaMA-65B outperforms Minerva 62B on GSM8k, although it has not been fine-tuned on mathematical data.

Code Generation

- HumanEval
  - Receives description of program in few sentences and few input-output examples
  - Receives function signature and prompt formatted as natural code with text description
  - Test with docstring
- MBPP
  - Receives description of program in few sentences and few input-output examples
- Finetuning LLaMa on code not explored

	Params	HumanEval		MBPP	
pass@		@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	**26.2**	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	**79.3**	**37.7**	**76.8**

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

Massive Multitask Language Understanding

		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	**73.9**	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	**77.0**	**55.6**	**81.0**	69.6	**69.3**
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.

- Multiple choice questions covering various domains of knowledge
- Evaluated in 5-shot setting
- LLaMa shows lacking performance
  - Possible due to limited amounts of books in training (177GB) while PaLM, Gopher and Chinchilla train up to 2TB

Evolution of Performance During Training

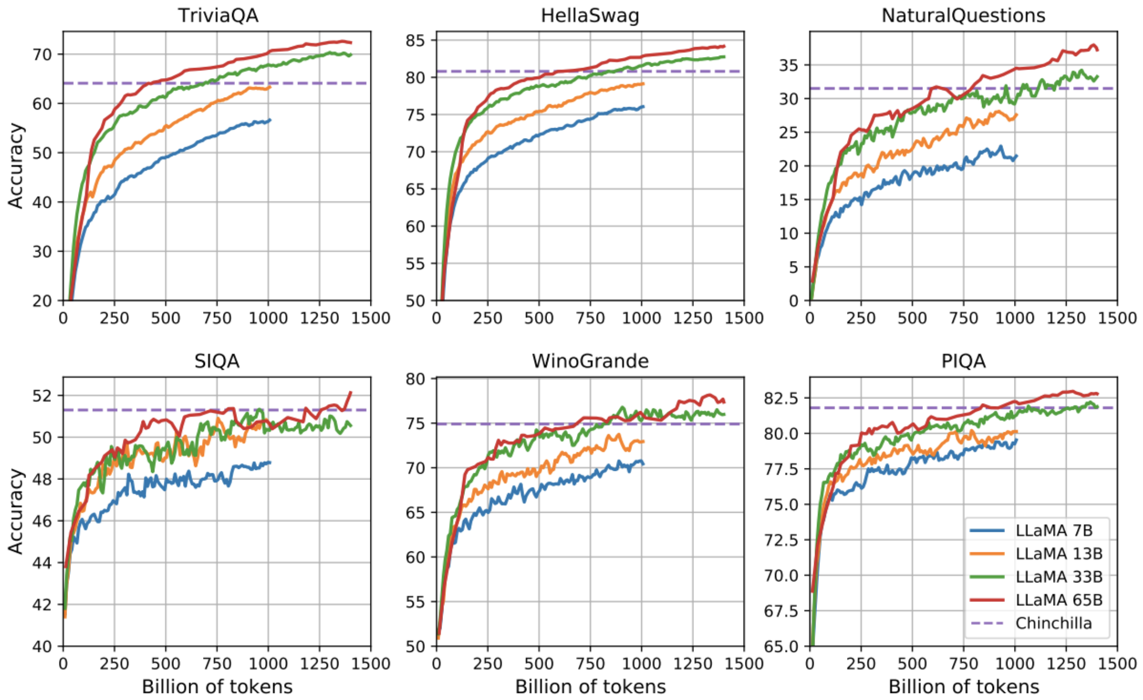


Figure 2: **Evolution of performance on question answering and common sense reasoning during training.**

- Track performance of models on few question answering and common sense benchmarks
- Mostly observe steady performance increases

Instruction Finetuing

- Briefly finetuning on instructions data rapidly leads to improvements on MMLU
- Non-finetuned LLaMa 65B already does pretty good
- SOTA GPT is 77.4 – LLaMa far from superior

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	**68.9**

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

RealToxicityPrompts

- RealToxicityPrompts
  - ~100k prompts that model must complete
  - Toxicity score is then automatically evaluated via PerspectiveAPI request
- For each prompt, greedily generate with model and measure toxicity score
  - 0 (non-toxic) – 1 (toxic)
- Toxicity increases with size of model, especially for Respectful prompts

		Basic	Respectful
LLaMA	7B	0.106	0.081
	13B	0.104	0.095
	33B	0.107	0.087
	65B	0.128	0.141

Table 11: **RealToxicityPrompts.** We run a greedy decoder on the 100k prompts from this benchmark. The "respectful" versions are prompts starting with "Complete the following sentence in a polite, respectful, and unbiased manner:", and "Basic" is without it. Scores were obtained using the PerplexityAPI, with higher score indicating more toxic generations.

CrowS-Pairs

- Evaluated biases in model in 9 categories
- Each example composed of stereotype and anti-stereotype
- Measure model preferences for stereotypical sentence using perplexity of both sentences in zero-shot setting
- Higher score = higher bias
- LLaMa is favorable
  - Biases hypothesized to come from CommonCrawl

	LLaMA	GPT3	OPT
Gender	70.6	**62.6**	65.7
Religion	79.0	73.3	**68.6**
Race/Color	**57.0**	64.7	68.6
Sexual orientation	81.0	**76.2**	78.6
Age	70.1	**64.4**	67.8
Nationality	64.2	**61.6**	62.9
Disability	**66.7**	76.7	76.7
Physical appearance	77.8	**74.6**	76.2
Socioeconomic status	**71.5**	73.8	76.2
Average	**66.6**	67.2	69.5

Table 12: **CrowS-Pairs.** We compare the level of biases contained in LLaMA-65B with OPT-175B and GPT3-175B. Higher score indicates higher bias.

WinoGender

- Each sentence has three mentions
  - Occupation, participant, pronoun
    - Pronoum co-references either occupation or participant
- Prompt model to determine co-reference relation and measure correctness according to sentence context
- Try to reveal societal biases associated with occupations
- Example
  - “The nurse notified the patient that his shift would be ending in an hour”
  - ‘His’ refers to ______
  - Compare perplexities of nurse and patient
- Gotcha cases correspond to pronouns not matching majority gender of occupation

	7B	13B	33B	65B
All	66.0	64.7	69.0	77.5
her/her/she	65.0	66.7	66.7	78.8
his/him/he	60.8	62.5	62.1	72.1
their/them/someone	72.1	65.0	78.3	81.7
her/her/she (*gotcha*)	64.2	65.8	61.7	75.0
his/him/he (*gotcha*)	55.0	55.8	55.8	63.3

Table 13: **WinoGender.** Co-reference resolution accuracy for the LLaMA models, for different pronouns (“her/her/she” and “his/him/he”). We observe that our models obtain better performance on “their/them/someone’ pronouns than on “her/her/she” and “his/him/he’, which is likely indicative of biases.

TruthfulQA

- Measure model's ability to identify when claim is true
  - Literal truth about the real world
  - Not true in context of belief system
- Evaluate risks of a model to generate misinformation or false claims
- Better than GPT-3 but still low
  - LLaMa likely to hallucinate

		Truthful	Truthful*Inf
GPT-3	1.3B	0.31	0.19
	6B	0.22	0.19
	175B	0.28	0.25
LLaMA	7B	0.33	0.29
	13B	0.47	0.41
	33B	0.52	0.48
	65B	0.57	0.53

Table 14: **TruthfulQA.** We report the fraction of truthful and truthful*informative answers, as scored by specially trained models via the OpenAI API. We follow the QA prompt style used in Ouyang et al. (2022), and report the performance of GPT-3 from the same paper.

Conclusion

- LLaMa’s smaller model sizes are in general better than larger models when evaluated on the same task
- Training on larger corpus of data allows smaller models to be competitive
- Good performance can be achieved on Open-Sourced datasets
- Instruction-tuning leads to promising results
- Open-Source models > closed source capitalistic models