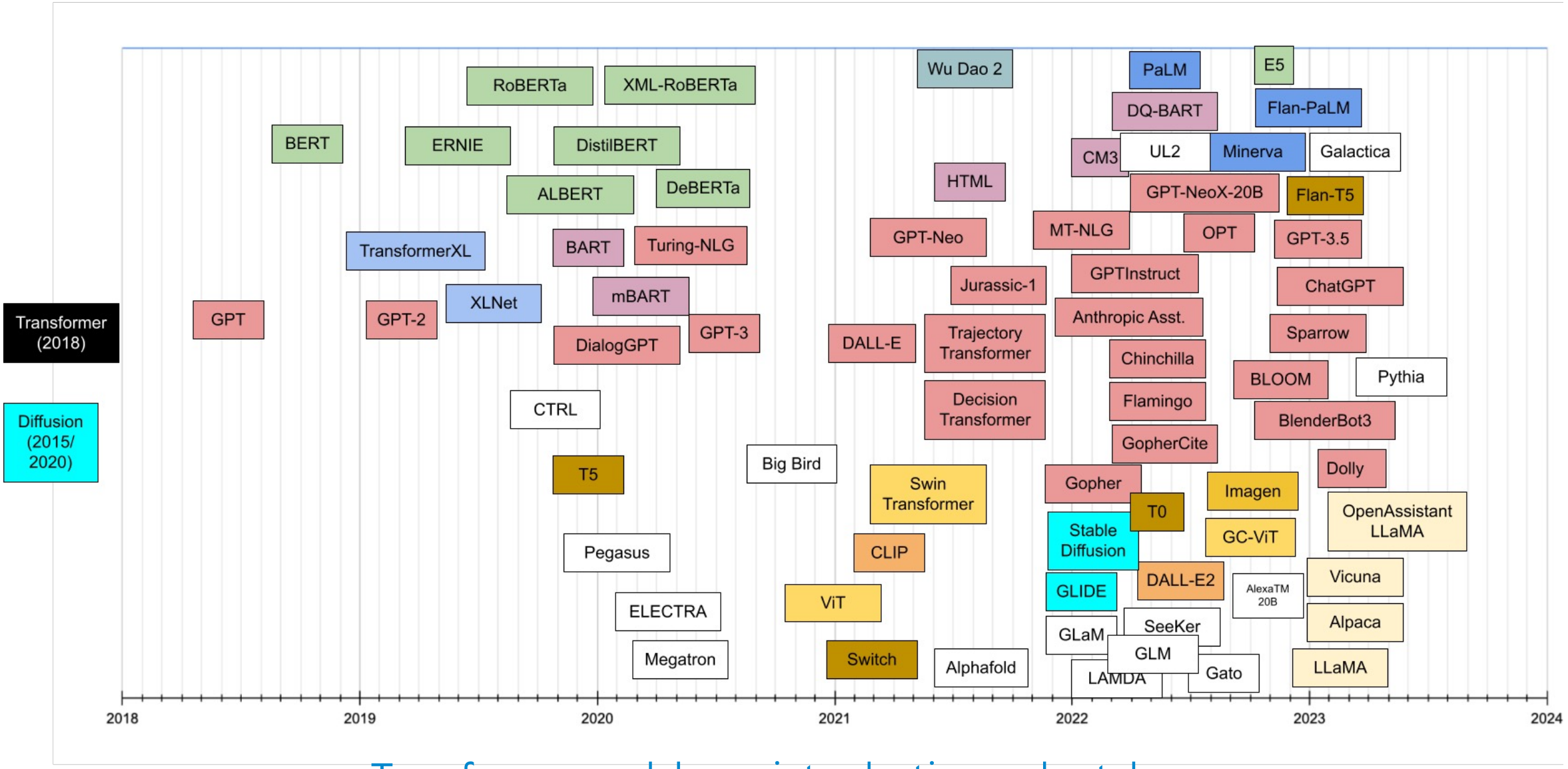
Лекция 12: Large Language Models

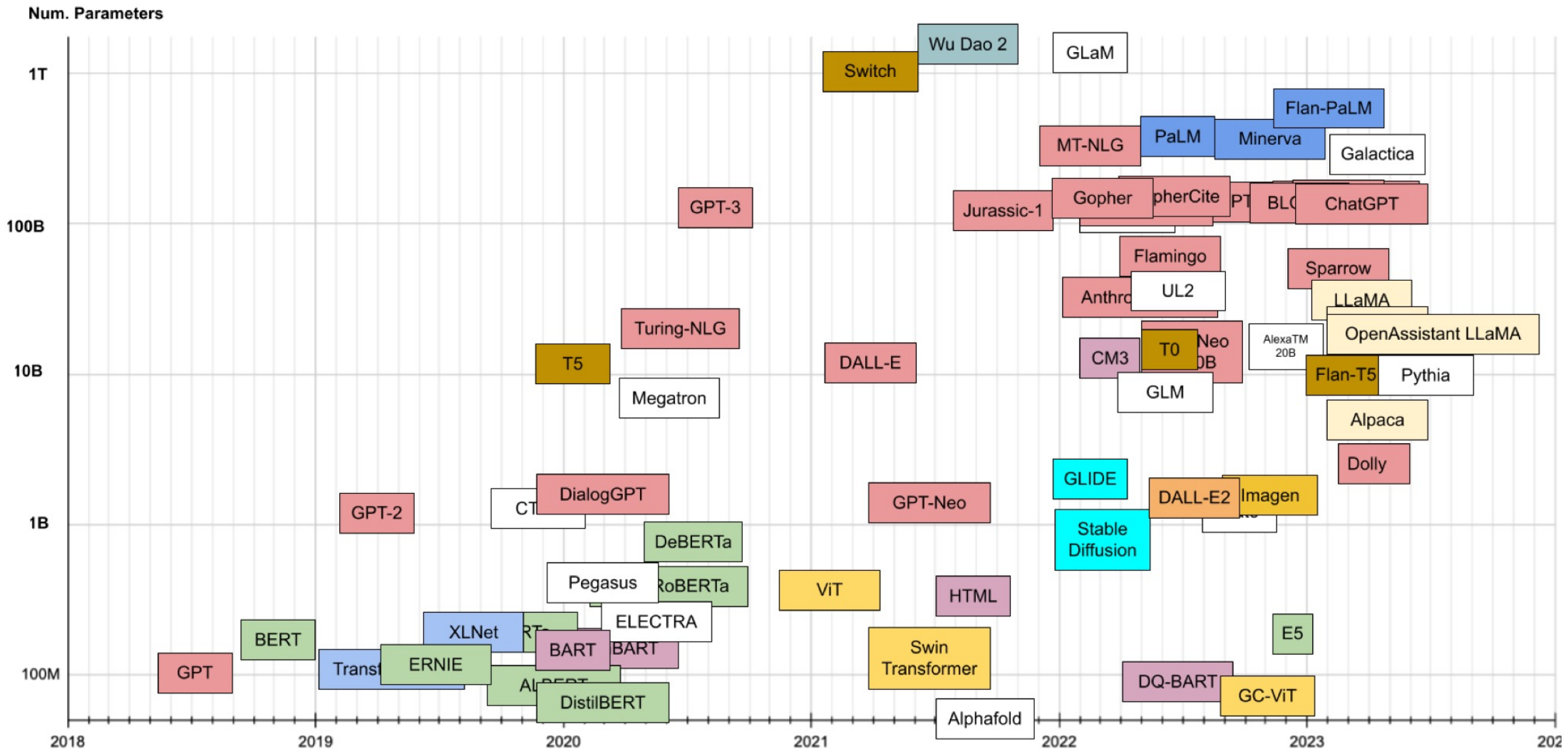
Ольга Цымбой

Sber AI Lab

Апрель 2024



[Transformer models: an introduction and catalog]



[Transformer models: an introduction and catalog]

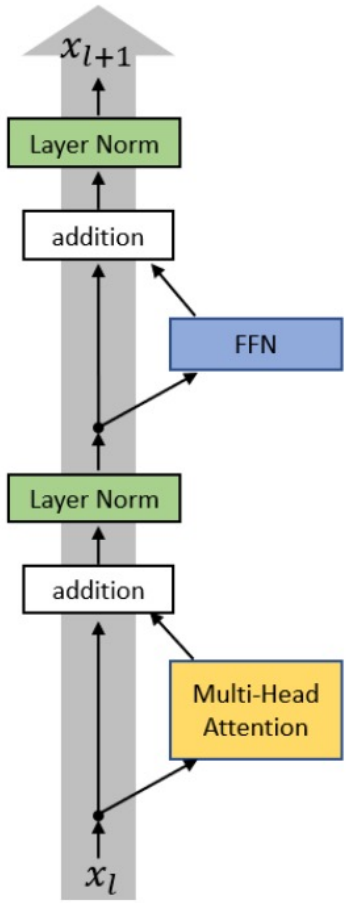
Recap

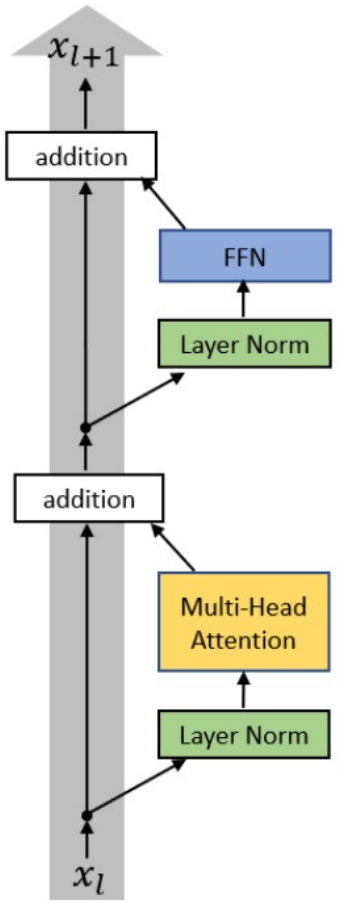
GPT-3: Language Models are Few-Shot Learners

Decoder only model:

- Pre-normalization

GPT





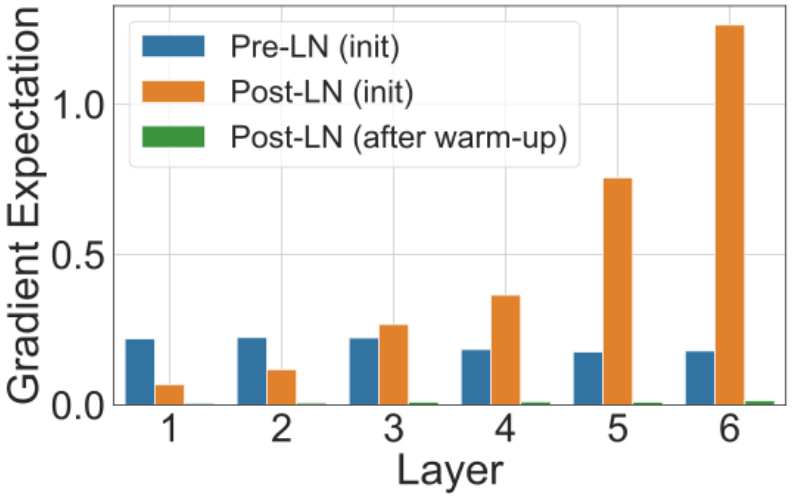
GPT-2
GPT-3

[On Layer Normalization in the Transformer Architecture]

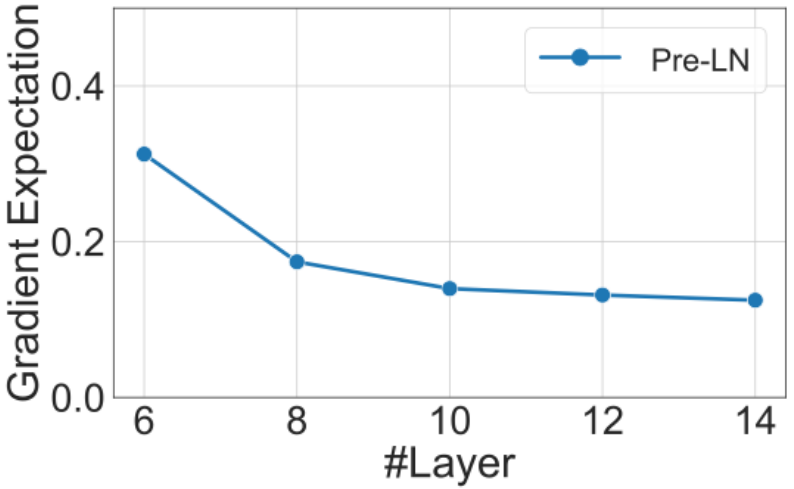
GPT-3: Language Models are Few-Shot Learners

Decoder only model:

- Pre-normalization: стабилизирует обучение



Зависимость нормы градиентов FFN слоя



Зависимость нормы градиентов последнего FFN слоя

[On Layer Normalization in the Transformer Architecture]

GPT-3: Language Models are Few-Shot Learners

Decoder only model:

- Pre-normalization
- Измененная инициализация residual layers: $W * 1/\sqrt{N}$, N – число residual слоев
- Sparse attention

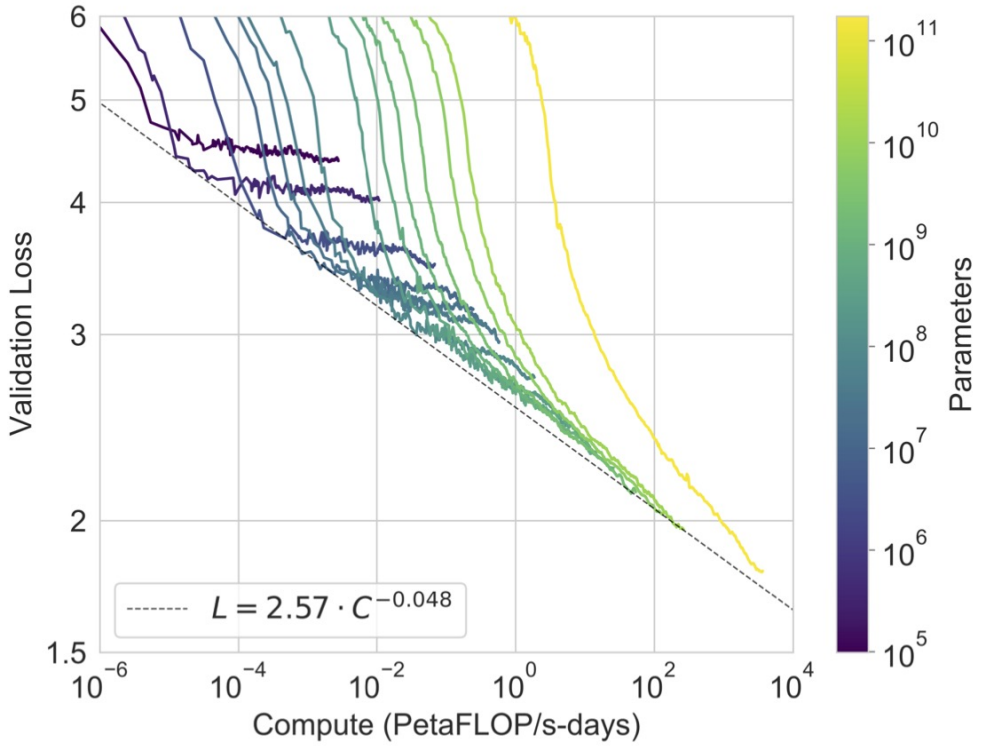
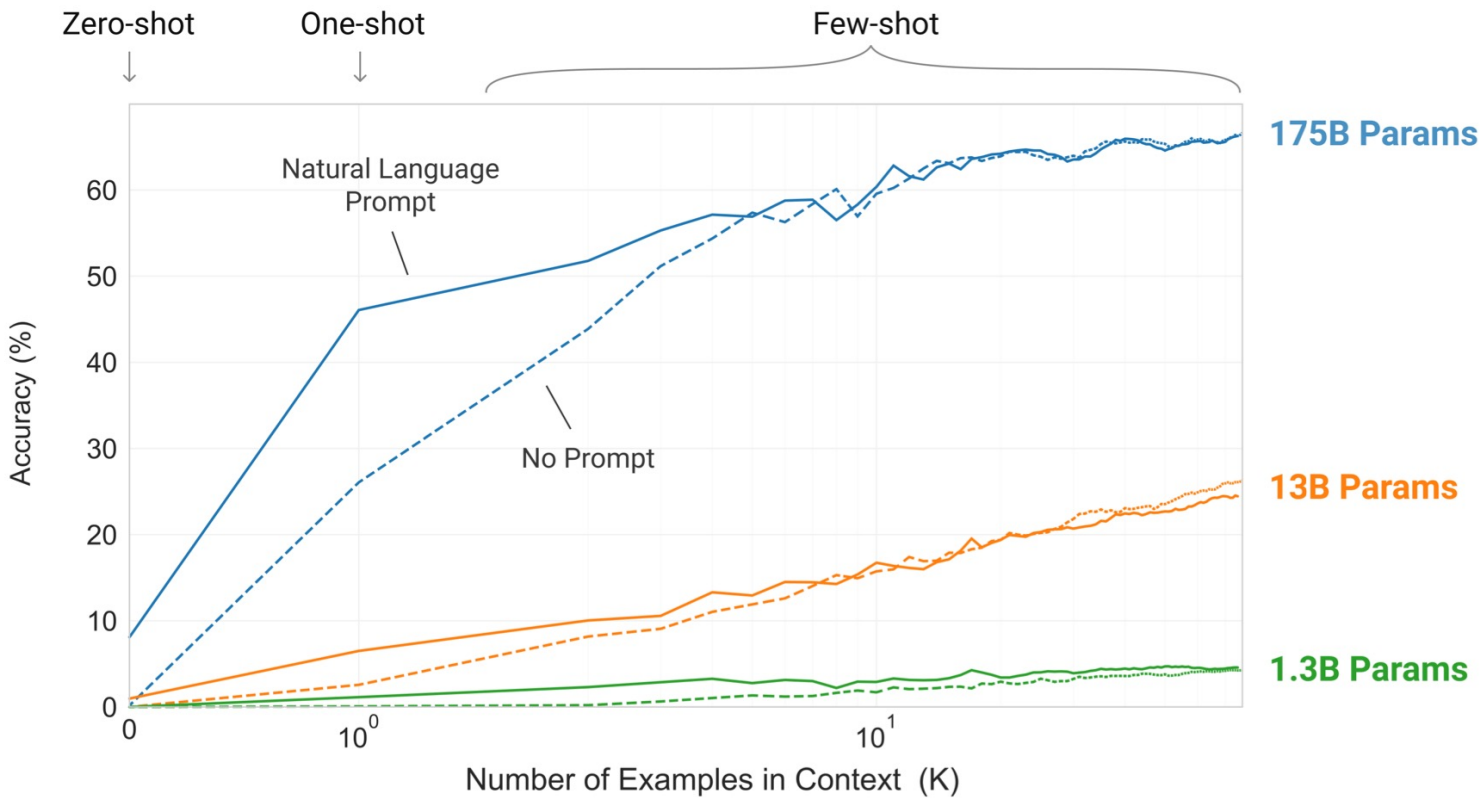


Figure 3.1: Smooth scaling of performance with compute. Performance (measured in terms of cross-entropy validation loss) follows a power-law trend with the amount of compute used for training. The power-law behavior observed in [KMH+20] continues for an additional two orders of magnitude with only small deviations from the predicted curve. For this figure, we exclude embedding parameters from compute and parameter counts.

GPT-3: Language Models are Few-Shot Learners

Decoder only model:

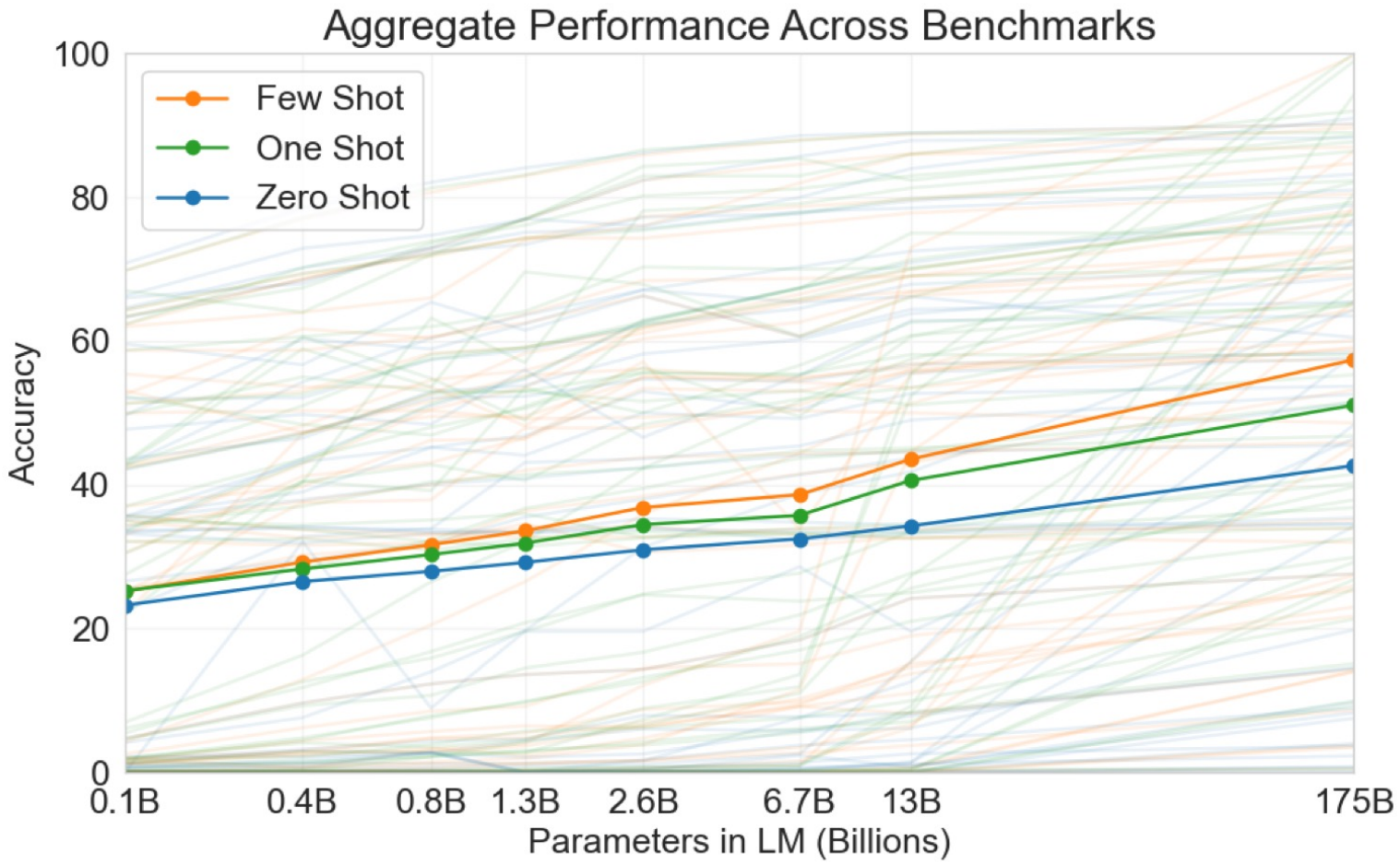
- 175B параметров
  L=96, H=12288, A=96
- 570ГБ текста
  300B токенов
  (Common Crawl, WebText, ..)
- Размер контекста
  2048 токенов



GPT-3: Language Models are Few-Shot Learners

Decoder only model:

- 175B параметров
  L=96, H=12288, A=96
- 570ГБ текста
  300B токенов
  (Common Crawl,
  WebText, ..)
- Размер контекста
  2048 токенов



Positional Embedding

Positional Embedding

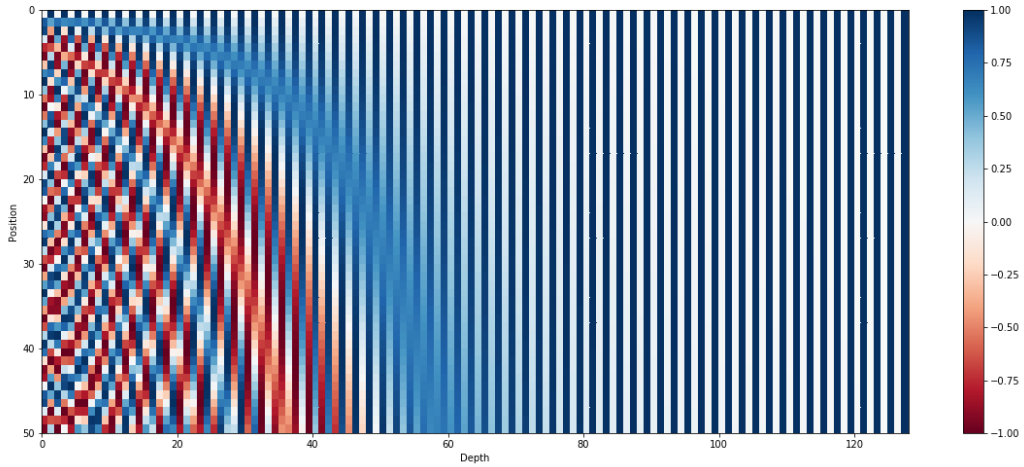
Absolute:

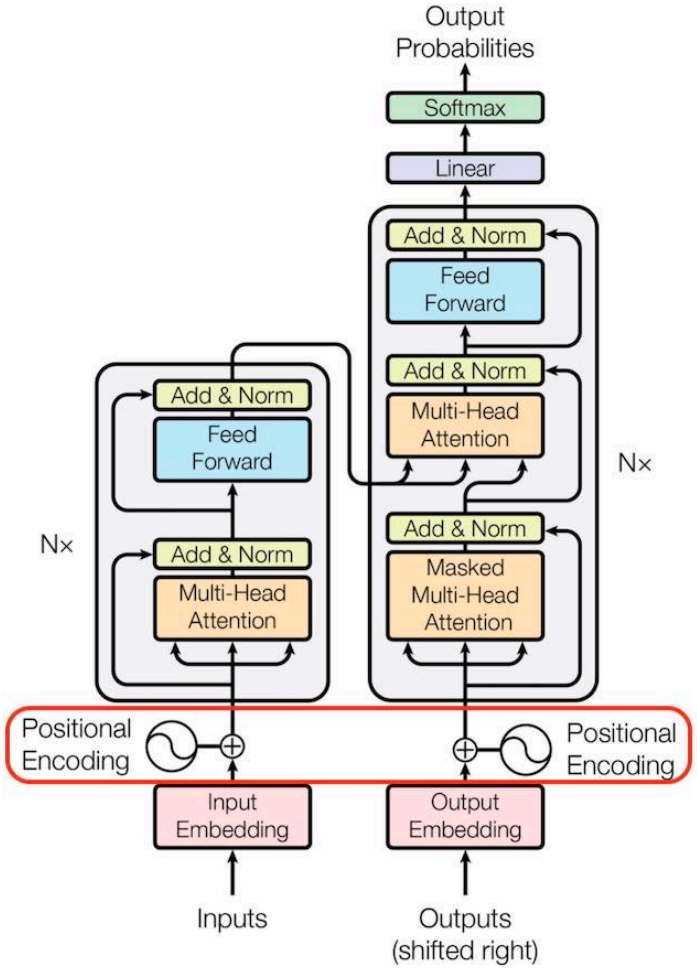
- **Sinusoidal (Attention Is All You Need)**

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

PE_{pos+k} линейно зависит от PE_{pos}





- **Learnable (e.g. BERT, GPT)**

Relative Positional Embedding

Paper:

$$e_{ij} = \frac{x_i W^Q (x_j W^K + a_{ij}^K)^T}{\sqrt{d_z}}$$

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}}$$

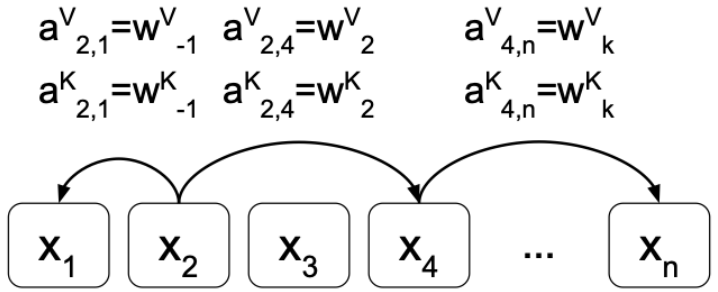
$$z_i = \sum_{j=1}^{n} \alpha_{ij} (x_j W^V + a_{ij}^V)$$

Learnable w_i:

$$a_{ij}^K = w_{\text{clip}(j-i,k)}^K$$
$$a_{ij}^V = w_{\text{clip}(j-i,k)}^V$$
$$\text{clip}(x,k) = \max(-k, \min(k,x))$$

Model	Position Information	EN-DE BLEU	EN-FR BLEU
Transformer (base)	Absolute Position Representations	26.5	38.2
Transformer (base)	Relative Position Representations	**26.8**	**38.7**
Transformer (big)	Absolute Position Representations	27.9	41.2
Transformer (big)	Relative Position Representations	**29.2**	**41.5**

Table 1: Experimental results for WMT 2014 English-to-German (EN-DE) and English-to-French (EN-FR) translation tasks, using newstest2014 test set.



$O(L^2 D)$ дополнительной памяти
L – длина поледовательности
D – размерность скрытого пространста

Relative Positional Embedding : Transformer-XL

Absolute encoding p_m, p_n:

$$q_m^\intercal k_n = x_m^\intercal W_q^\intercal W_k x_n + x_m^\intercal W_q^\intercal W_k p_n + p_m^\intercal W_q^\intercal W_k x_n + p_m^\intercal W_q^\intercal W_k p_n,$$

Reparameterization:

$$q_m^\intercal k_n = x_m^\intercal W_q^\intercal \underbrace{W_k} x_n + x_m^\intercal W_q^\intercal \underbrace{\widetilde{W}_k} \tilde{p}_{m-n} + \mathbf{u}^\intercal W_q^\intercal W_k x_n + \mathbf{v}^\intercal W_q^\intercal \widetilde{W}_k \tilde{p}_{m-n}$$

Context-based Location-based

p_{m-n} **– sinusoid-encoded**

u, v **– trainable vectors**

Method	PPL
Ours	**25.2**
With Shaw et al. (2018) encodings	25.7
Without recurrence	27.1

Table 7: Ablation study on One Billion Word, a dataset without long-term dependency.

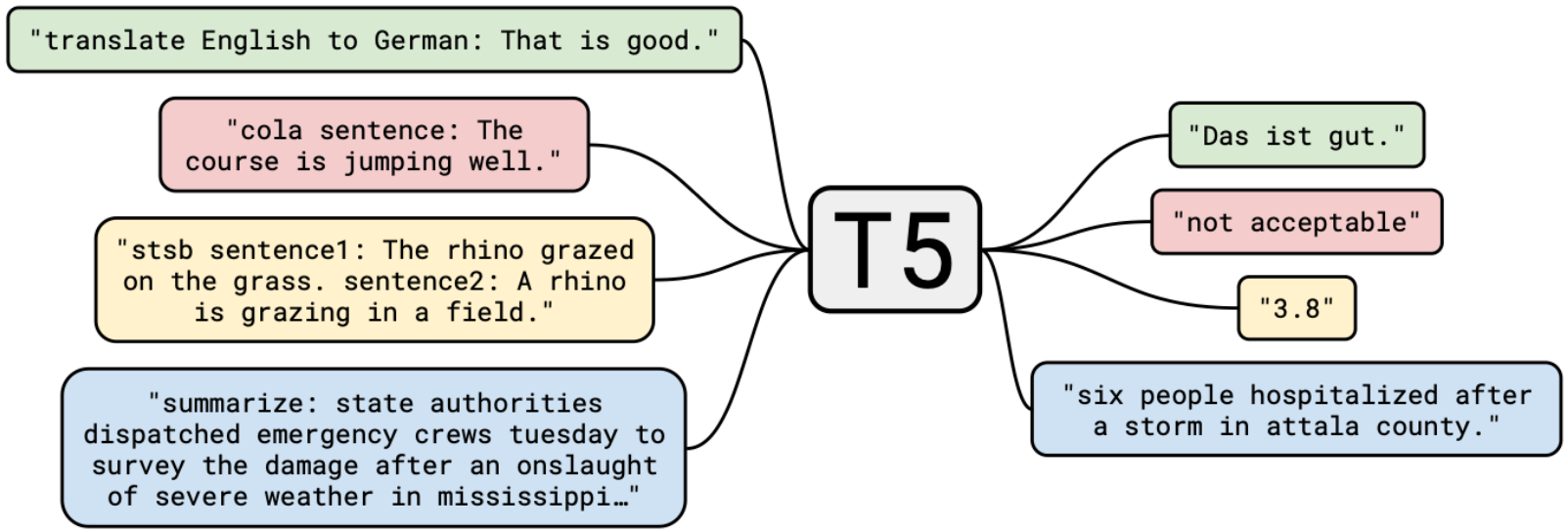
Relative Positional Embedding : T5

Simplify:

$$\boldsymbol{q}_m^\intercal \boldsymbol{k}_n = \boldsymbol{x}_m^\intercal \boldsymbol{W}_q^\intercal \boldsymbol{W}_k \boldsymbol{x}_n + \boldsymbol{x}_m^\intercal \boldsymbol{W}_q^\intercal \boldsymbol{W}_k \boldsymbol{p}_n + \boldsymbol{p}_m^\intercal \boldsymbol{W}_q^\intercal \boldsymbol{W}_k \boldsymbol{x}_n + \boldsymbol{p}_m^\intercal \boldsymbol{W}_q^\intercal \boldsymbol{W}_k \boldsymbol{p}_n,$$

To:

$$\boldsymbol{q}_m^\intercal \boldsymbol{k}_n = \boldsymbol{x}_m^\intercal \boldsymbol{W}_q^\intercal \boldsymbol{W}_k \boldsymbol{x}_n + b_{i,j}$$



Rotary Positional Embeddings

RoFormer

$$\langle f_q(\boldsymbol{x}_m, m), f_k(\boldsymbol{x}_n, n) \rangle = g(\boldsymbol{x}_m, \boldsymbol{x}_n, m-n).$$

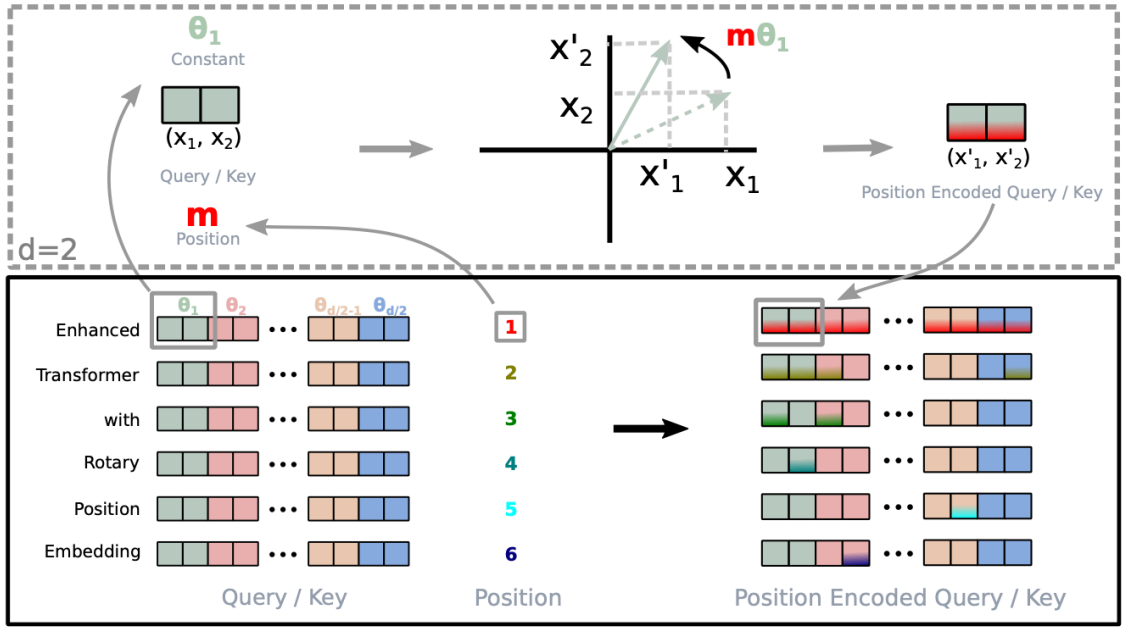


Figure 1: Implementation of Rotary Position Embedding(RoPE).

$$\begin{aligned} \boldsymbol{q}_m &= f_q(\boldsymbol{x}_q, m), & \boldsymbol{q} &= f_q(\boldsymbol{x}_q, 0), \\ \boldsymbol{k}_n &= f_k(\boldsymbol{x}_k, n), & \boldsymbol{k} &= f_k(\boldsymbol{x}_k, 0), \end{aligned}$$

$$\begin{aligned} \boldsymbol{q} &= f_q(\boldsymbol{x}_m, 0) = \boldsymbol{W}_q \boldsymbol{x}_n, \\ \boldsymbol{k} &= f_k(\boldsymbol{x}_n, 0) = \boldsymbol{W}_k \boldsymbol{x}_n. \end{aligned}$$

$$\begin{aligned} f_q(\boldsymbol{x}_m, m) &= (\boldsymbol{W}_q \boldsymbol{x}_m) e^{im\theta}, \\ f_k(\boldsymbol{x}_n, n) &= (\boldsymbol{W}_k \boldsymbol{x}_n) e^{in\theta}. \end{aligned}$$

Rotary Positional Embeddings

RoFormer

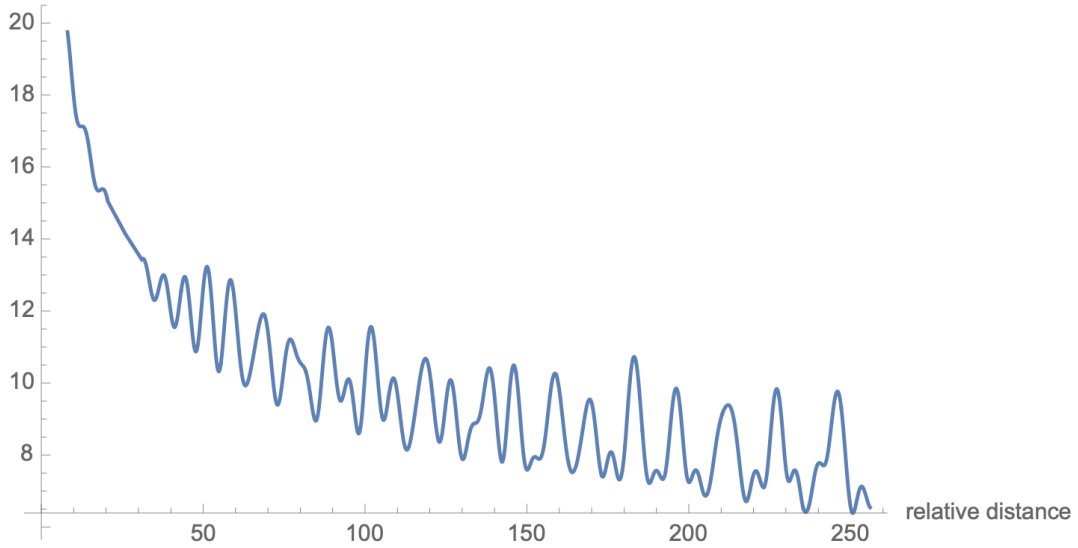


Figure 2: Long-term decay of RoPE.

Model	BLEU
Transformer-base Vaswani et al. [2017]	27.3
RoFormer	**27.5**

Table 2: Comparing RoFormer and BERT by fine tuning on downstream GLEU tasks.

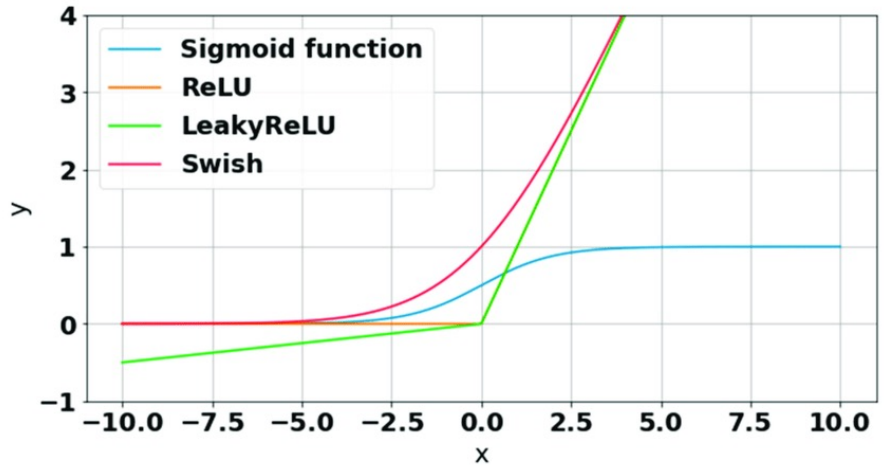
Model	MRPC	SST-2	QNLI	STS-B	QQP	MNLI(m/mm)
BERT Devlin et al. [2019]	88.9	93.5	90.5	85.8	71.2	84.6/83.4
RoFormer	**89.5**	90.7	88.0	**87.0**	**86.4**	80.2/79.8

LLaMA: Open and Efficient Foundation Language Models

Architecture:

- **Pre-normalization**
- **RMSNorm (Focus on rescaling only)**
- **SwiGLU**

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



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.**

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.

LLaMA: Open and Efficient Foundation Language Models

Architecture:

- **Pre-normalization**
- **RMSNorm (Focus on rescaling only)**
- **SwiGLU**
- **RoPE**

Languages: bg, ca, cs, da, de, en, es, fr, hr, hu, it, nl, pl, pt, ro, ru, sl, sr, sv, uk

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.**

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.

LLaMA: Open and Efficient Foundation Language Models

		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.**

		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.

		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.

LLaMA: Open and Efficient Foundation Language Models

	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.

		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.

LLaMA 2: Open Foundation and Fine-Tuned Chat Models

Pretraining data:
2T токенов, 4k контекст, 200 языков, включая русский

LLaMA 2:
- +40% данных для предобучения
- x2 размер контекста
- grouped-query attention

		Time (GPU hours)	Power Consumption (W)	Carbon Emitted (tCO_2eq)
Llama 2	7B	184320	400	31.22
	13B	368640	400	62.44
	34B	1038336	350	153.90
	70B	1720320	400	291.42
Total		3311616		539.00

	Training Data	Params	Context Length	GQA	Tokens	LR
Llama 1	*See Touvron et al. (2023)*	7B	2k	✗	1.0T	3.0×10^{-4}
		13B	2k	✗	1.0T	3.0×10^{-4}
		33B	2k	✗	1.4T	1.5×10^{-4}
		65B	2k	✗	1.4T	1.5×10^{-4}
Llama 2	*A new mix of publicly available online data*	7B	4k	✗	2.0T	3.0×10^{-4}
		13B	4k	✗	2.0T	3.0×10^{-4}
		34B	4k	✓	2.0T	1.5×10^{-4}
		70B	4k	✓	2.0T	1.5×10^{-4}

Table 1: Llama 2 family of models. Token counts refer to pretraining data only. All models are trained with a global batch-size of 4M tokens. Bigger models — 34B and 70B — use Grouped-Query Attention (GQA) for improved inference scalability.

LLaMA 2: Open Foundation and Fine-Tuned Chat Models

Pretraining:

Model	Size	Code	Commonsense Reasoning	World Knowledge	Reading Comprehension	Math	MMLU	BBH	AGI Eval
MPT	7B	20.5	57.4	41.0	57.5	4.9	26.8	31.0	23.5
	30B	28.9	64.9	50.0	64.7	9.1	46.9	38.0	33.8
Falcon	7B	5.6	56.1	42.8	36.0	4.6	26.2	28.0	21.2
	40B	15.2	69.2	56.7	65.7	12.6	55.4	37.1	37.0
LLAMA 1	7B	14.1	60.8	46.2	58.5	6.95	35.1	30.3	23.9
	13B	18.9	66.1	52.6	62.3	10.9	46.9	37.0	33.9
	33B	26.0	70.0	58.4	67.6	21.4	57.8	39.8	41.7
	65B	30.7	70.7	60.5	68.6	30.8	63.4	43.5	47.6
LLAMA 2	7B	16.8	63.9	48.9	61.3	14.6	45.3	32.6	29.3
	13B	24.5	66.9	55.4	65.8	28.7	54.8	39.4	39.1
	34B	27.8	69.9	58.7	68.0	24.2	62.6	44.1	43.4
	70B	**37.5**	**71.9**	**63.6**	**69.4**	**35.2**	**68.9**	**51.2**	**54.2**

Table 3: Overall performance on grouped academic benchmarks compared to open-source base models.

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	**86.4**	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	**86.1**	85.0
Natural Questions (1-shot)	–	–	29.3	**37.5**	33.0
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8
HumanEval (0-shot)	48.1	**67.0**	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	**65.7**	51.2

Table 4: Comparison to closed-source models on academic benchmarks. Results for GPT-3.5 and GPT-4 are from OpenAI (2023). Results for the PaLM model are from Chowdhery et al. (2022). Results for the PaLM-2-L are from Anil et al. (2023).

LLaMA 2: Open Foundation and Fine-Tuned Chat Models

Reward model:

$$\mathcal{L}_{\text{ranking}} = -\log(\sigma(r_\theta(x, y_c) - r_\theta(x, y_r) - m(r)))$$

Dataset	Num. of Comparisons	Avg. # Turns per Dialogue	Avg. # Tokens per Example	Avg. # Tokens in Prompt	Avg. # Tokens in Response
Anthropic Helpful	122,387	3.0	251.5	17.7	88.4
Anthropic Harmless	43,966	3.0	152.5	15.7	46.4
OpenAI Summarize	176,625	1.0	371.1	336.0	35.1
OpenAI WebGPT	13,333	1.0	237.2	48.3	188.9
StackExchange	1,038,480	1.0	440.2	200.1	240.2
Stanford SHP	74,882	1.0	338.3	199.5	138.8
Synthetic GPT-J	33,139	1.0	123.3	13.0	110.3
Meta (Safety & Helpfulness)	1,418,091	3.9	798.5	31.4	234.1
Total	2,919,326	1.6	595.7	108.2	216.9

Table 6: Statistics of human preference data for reward modeling. We list both the open-source and internally collected human preference data used for reward modeling. Note that a binary human preference comparison contains 2 responses (chosen and rejected) sharing the same prompt (and previous dialogue). Each example consists of a prompt (including previous dialogue if available) and a response, which is the input of the reward model. We report the number of comparisons, the average number of turns per dialogue, the average number of tokens per example, per prompt and per response. More details on Meta helpfulness and safety data per batch can be found in Appendix A.3.1.

LLaMA 2: Open Foundation and Fine-Tuned Chat Models

Reward model:

	Meta Helpful.	Meta Safety	Anthropic Helpful	Anthropic Harmless	OpenAI Summ.	Stanford SHP	Avg
SteamSHP-XL	52.8	43.8	66.8	34.2	54.7	75.7	55.3
Open Assistant	53.8	53.4	67.7	68.4	71.7	55.0	63.0
GPT4	58.6	58.1	-	-	-	-	-
Safety RM	56.2	64.5	55.4	74.7	71.7	65.2	64.3
Helpfulness RM	63.2	62.8	72.0	71.0	75.5	80.0	70.6

Table 7: Reward model results. Performance of our final helpfulness and safety reward models on a diverse set of human preference benchmarks. Note that our model is fine-tuned on our collected data, as opposed to the other baselines that we report.

LLaMA 2: Open Foundation and Fine-Tuned Chat Models

RLHF:

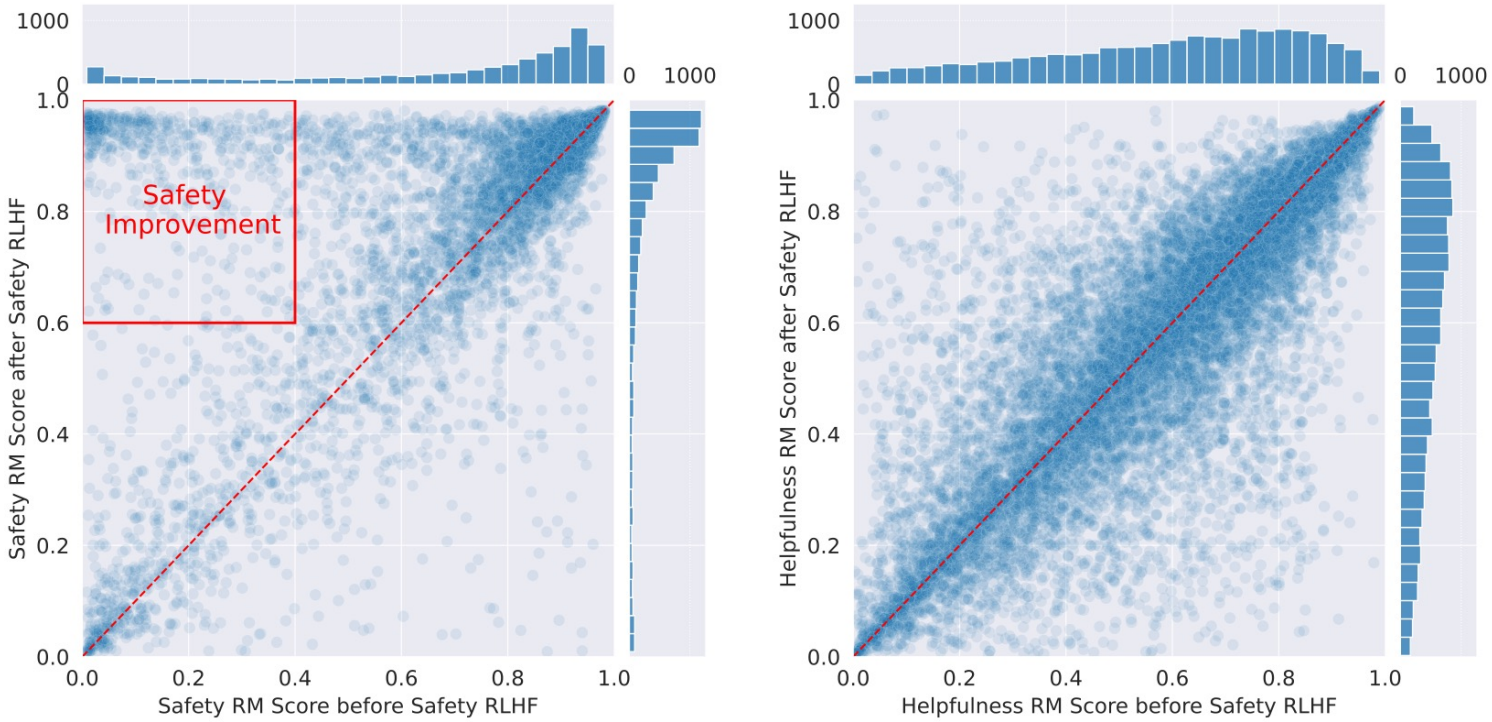
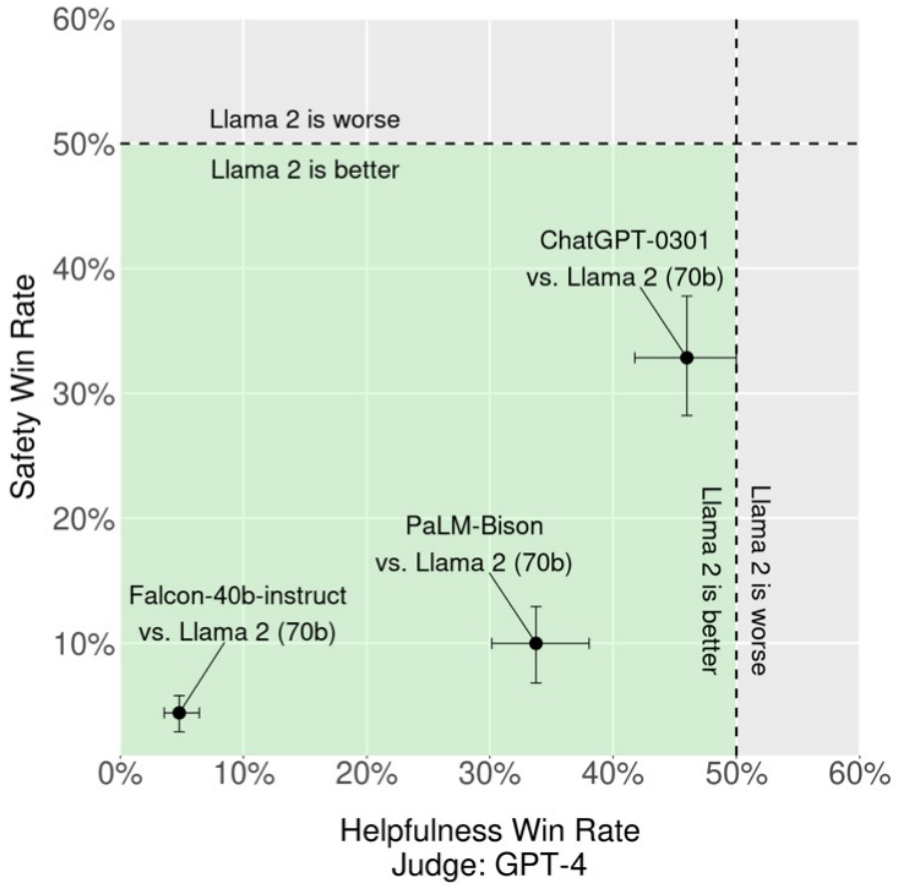
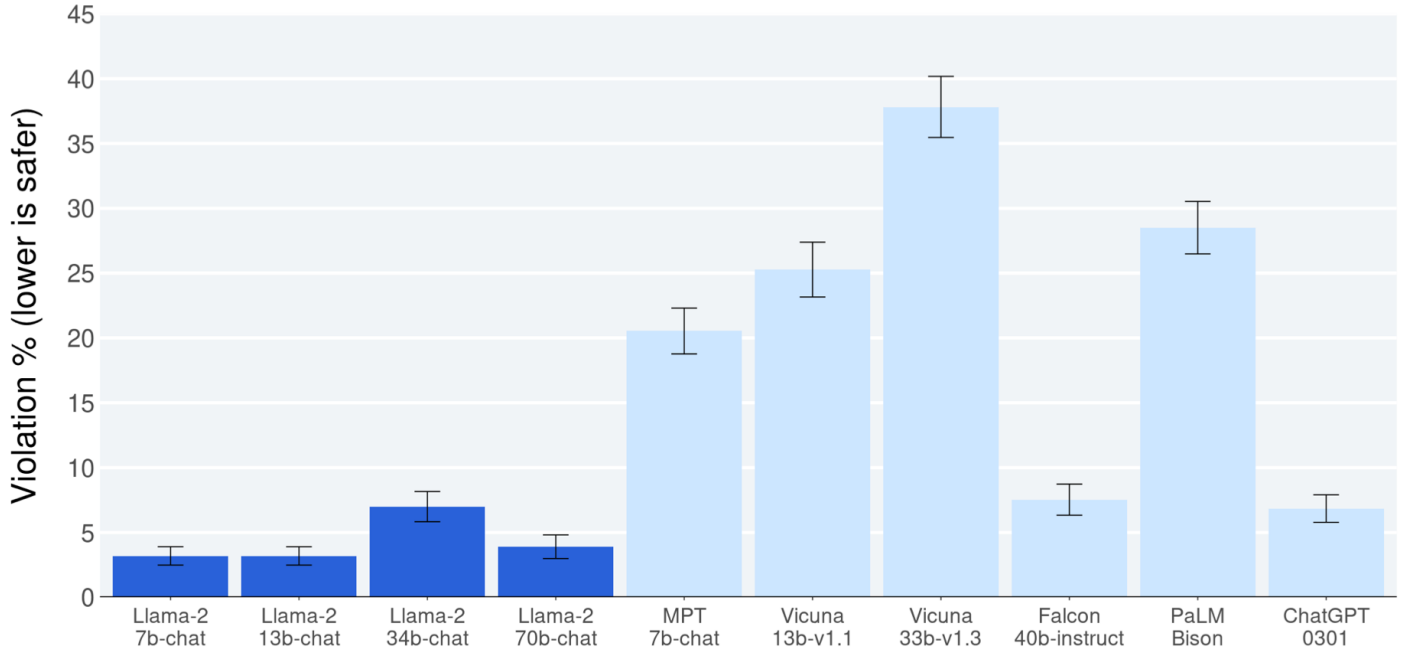


Figure 14: Impact of safety RLHF measured by reward model score distributions. *Left*: safety reward model scores of generations on the Meta Safety test set. The clustering of samples in the top left corner suggests the improvements of model safety. *Right*: helpfulness reward model scores of generations on the Meta Helpfulness test set.

LLaMA 2: Open Foundation and Fine-Tuned Chat Models

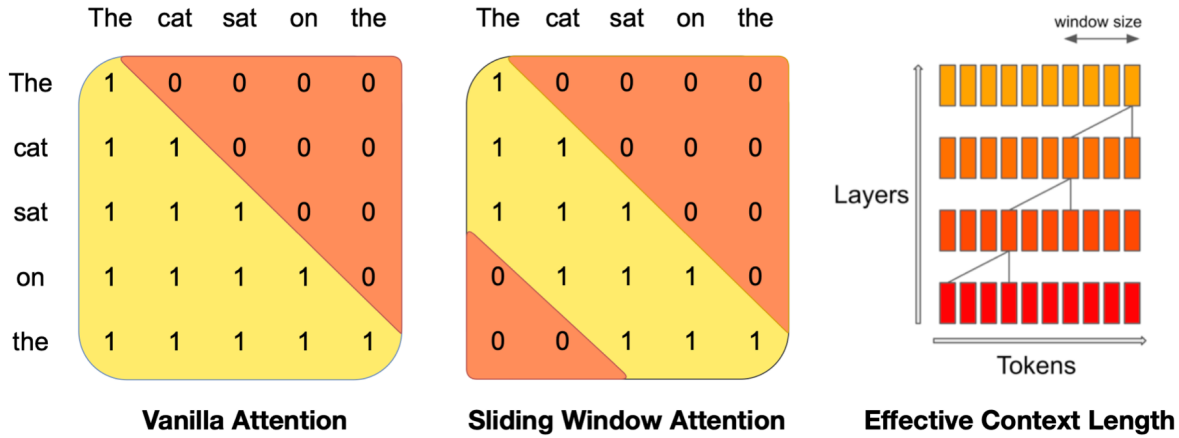
Chat:





Mistral 7B

Architecture:



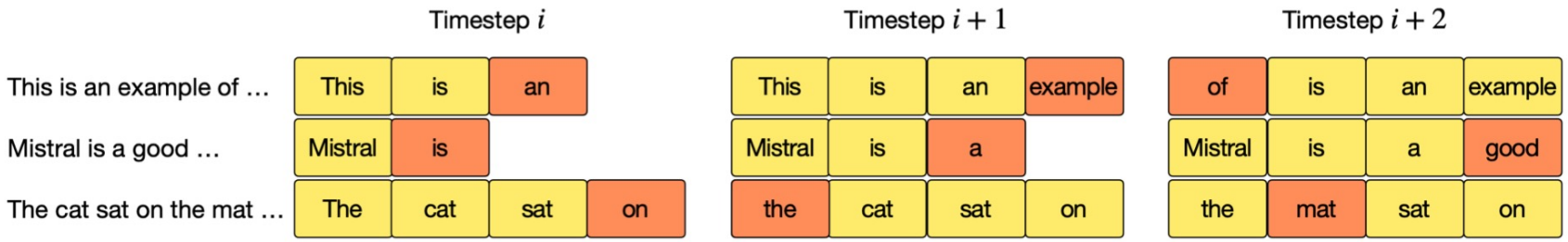


Figure 2: Rolling buffer cache. The cache has a fixed size of $W = 4$. Keys and values for position i are stored in position $i \bmod W$ of the cache. When the position i is larger than W, past values in the cache are overwritten. The hidden state corresponding to the latest generated tokens are colored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

Mistral 7B

Architecture:

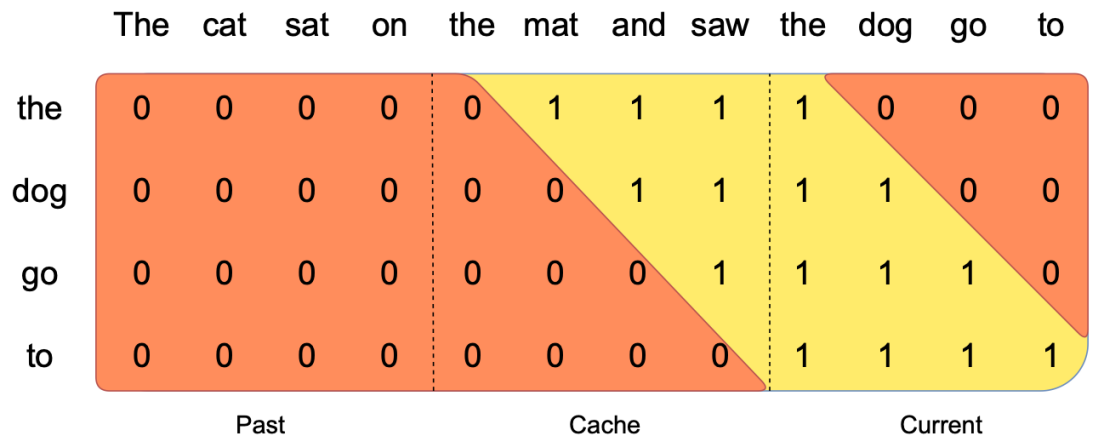
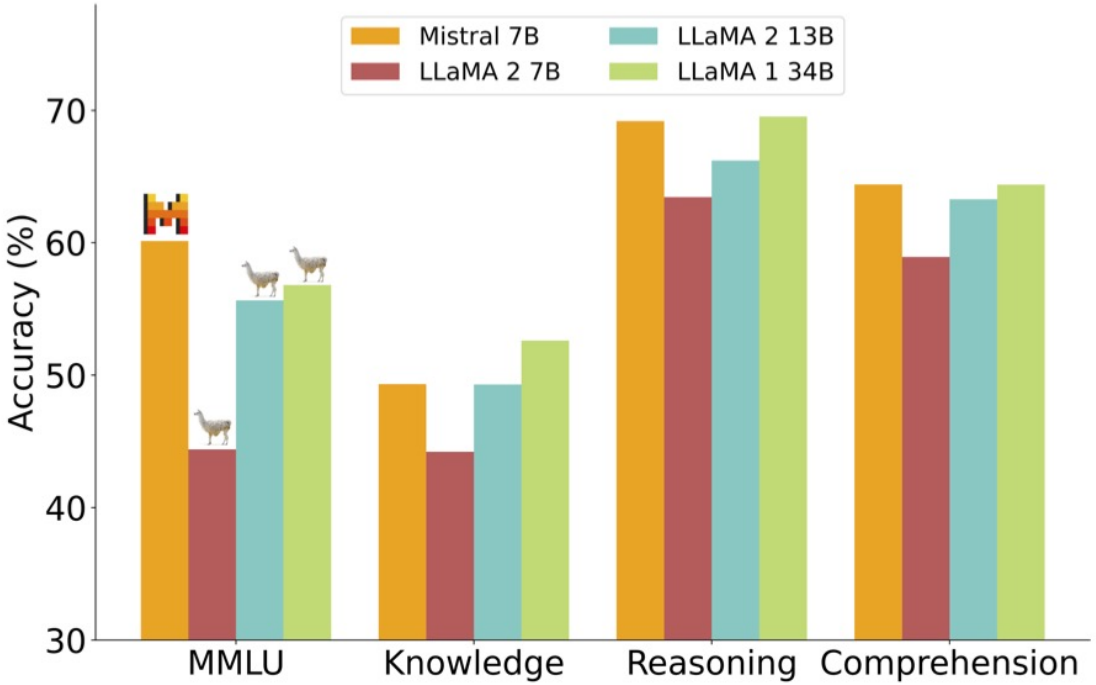
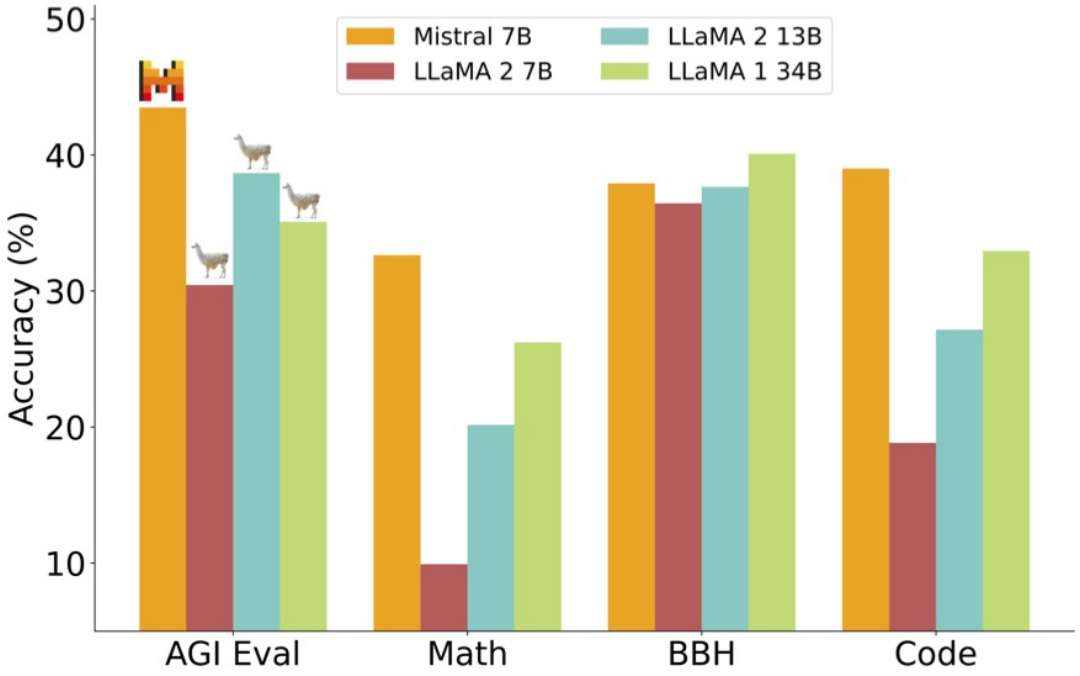


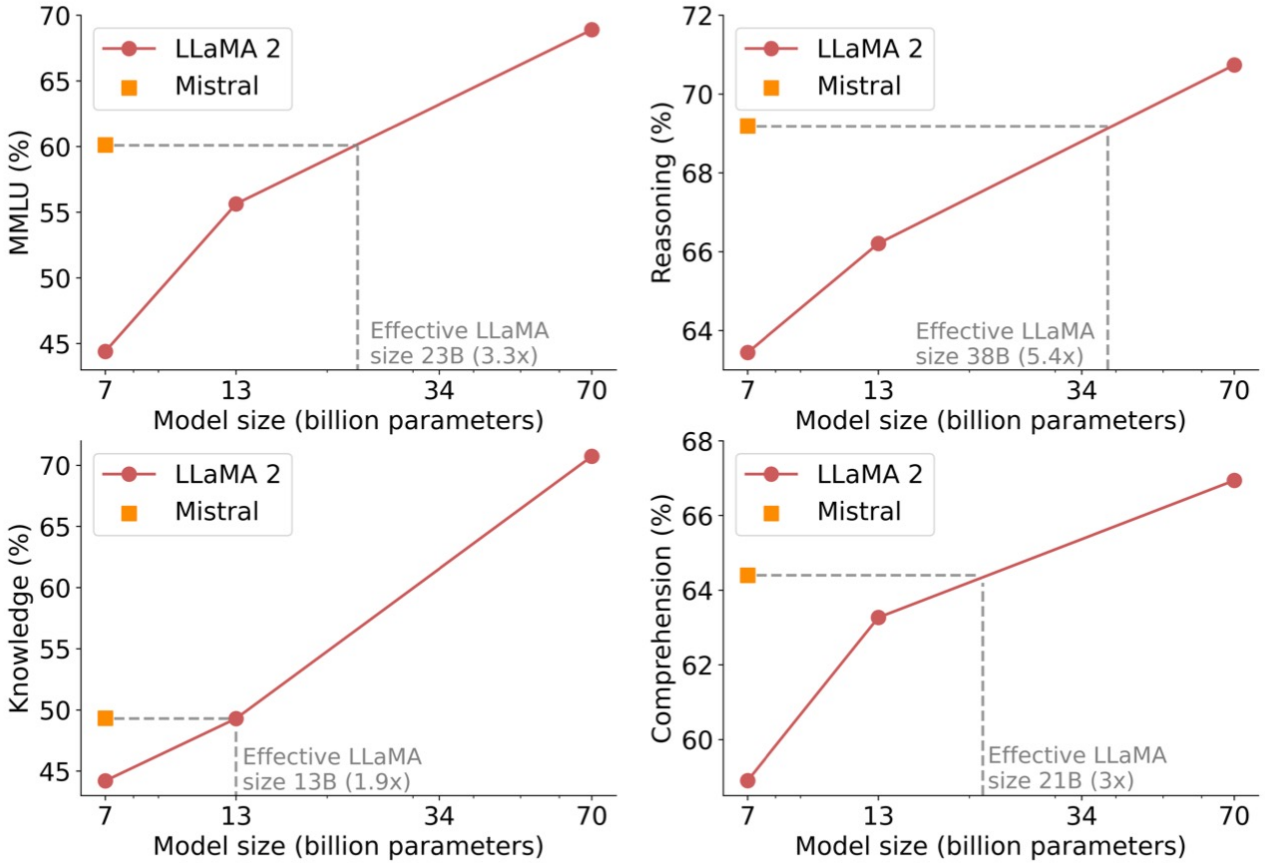
Figure 3: Pre-fill and chunking. During pre-fill of the cache, long sequences are chunked to limit memory usage. We process a sequence in three chunks, "The cat sat on", "the mat and saw", "the dog go to". The figure shows what happens for the third chunk ("the dog go to"): it attends itself using a causal mask (rightmost block), attends the cache using a sliding window (center block), and does not attend to past tokens as they are outside of the sliding window (left block).

Mistral 7B


Mistral 7B
LLaMA 2 7B
LLaMA 2 13B
LLaMA 1 34B
Accuracy (%)
70
60
50
40
30
MMLU
Knowledge
Reasoning
Comprehension


Mistral 7B
LLaMA 2 7B
LLaMA 2 13B
LLaMA 1 34B
Accuracy (%)
50
40
30
20
10
AGI Eval
Math
BBH
Code

Mistral 7B



Model	Chatbot Arena ELO Rating	MT Bench
WizardLM 13B v1.2	1047	7.2
Mistral 7B Instruct	**1031**	**6.84 +/- 0.07**
Llama 2 13B Chat	1012	6.65
Vicuna 13B	1041	6.57
Llama 2 7B Chat	985	6.27
Vicuna 7B	997	6.17
Alpaca 13B	914	4.53

Table 3: Comparison of Chat models. Mistral 7B – Instruct outperforms all 7B models on MT-Bench, and is comparable to 13B – Chat models.

Prompting

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

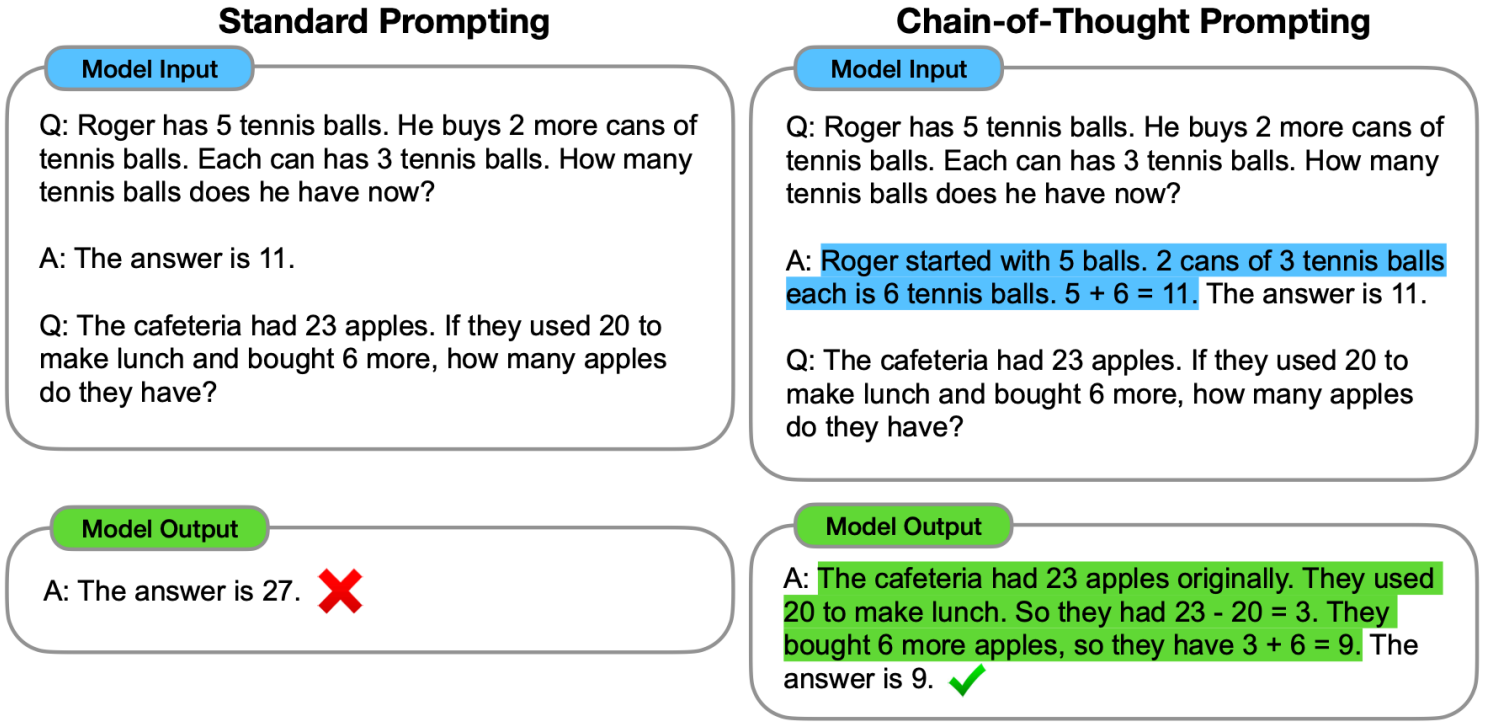
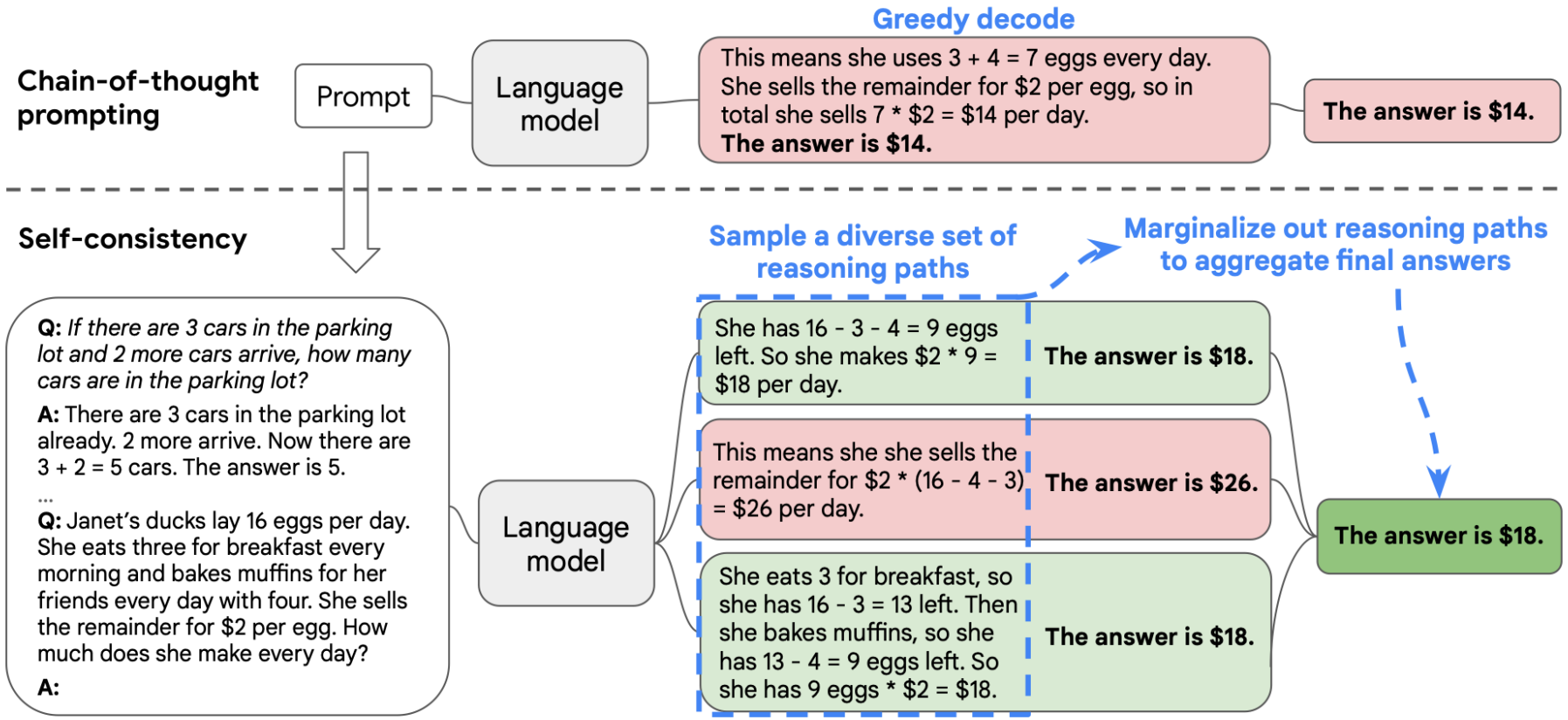


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Task + Let's think step by step

Self-Consistency Improves Chain of Thought Reasoning in Language Models



Let's think step by step + aggregation

Self-Consistency Improves Chain of Thought Reasoning in Language Models

	ANLI R1 / R2 / R3	e-SNLI	RTE	BoolQ	HotpotQA (EM/F1)
Standard-prompting (no-rationale)	69.1 / 55.8 / 55.8	85.8	84.8	71.3	27.1 / 36.8
CoT-prompting (Wei et al., 2022)	68.8 / 58.9 / 60.6	81.0	79.1	74.2	28.9 / 39.8
Self-consistency	**78.5 / 64.5 / 63.4**	**88.4**	**86.3**	**78.4**	**33.8 / 44.6**

Table 5: Compare Standard/CoT prompting with self-consistency on common NLP tasks.

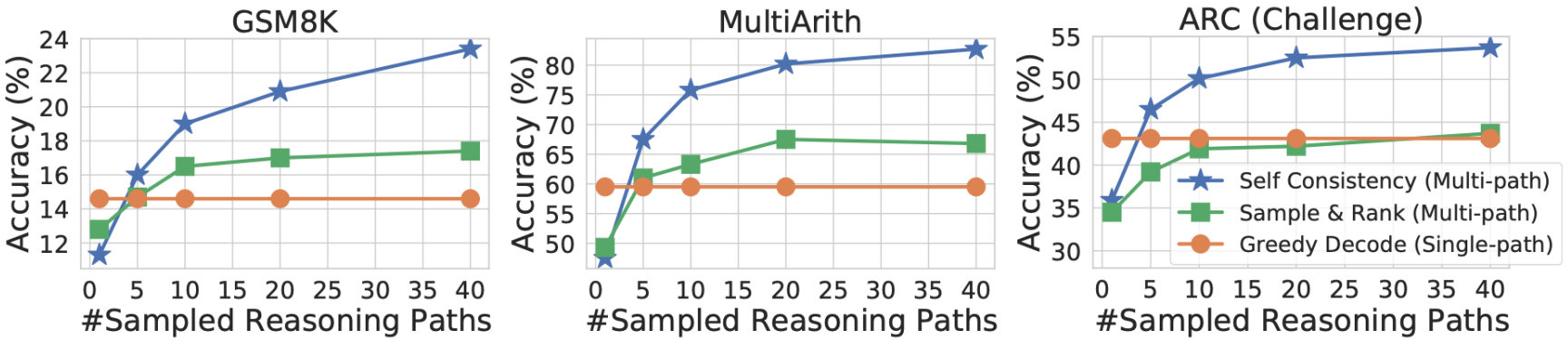


Figure 3: Self-consistency significantly outperforms sample-and-rank with the same # of samples.

Self-Consistency Improves Chain of Thought Reasoning in Language Models

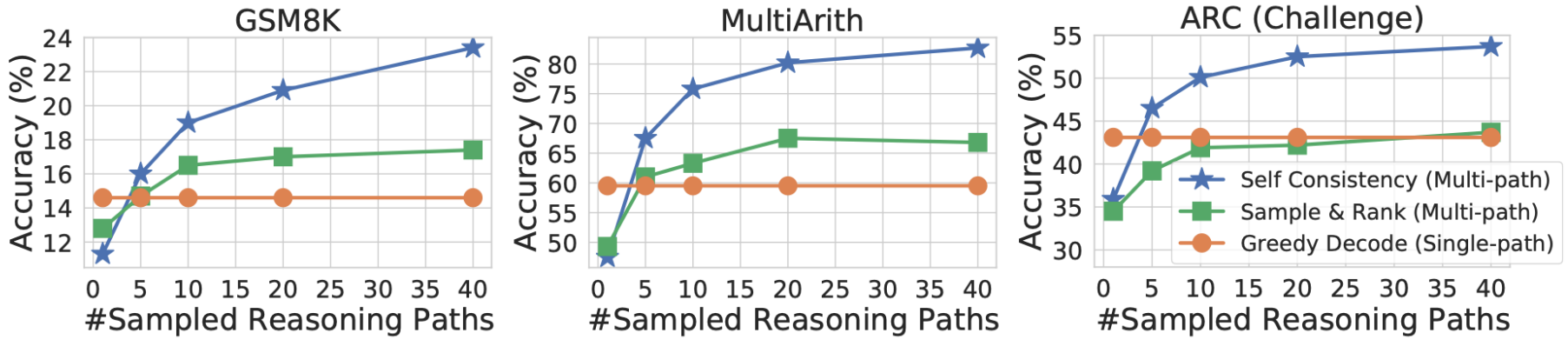
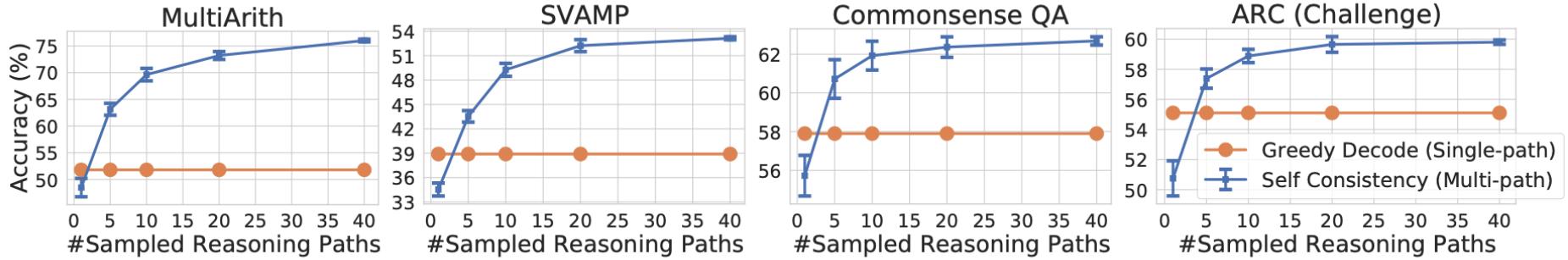
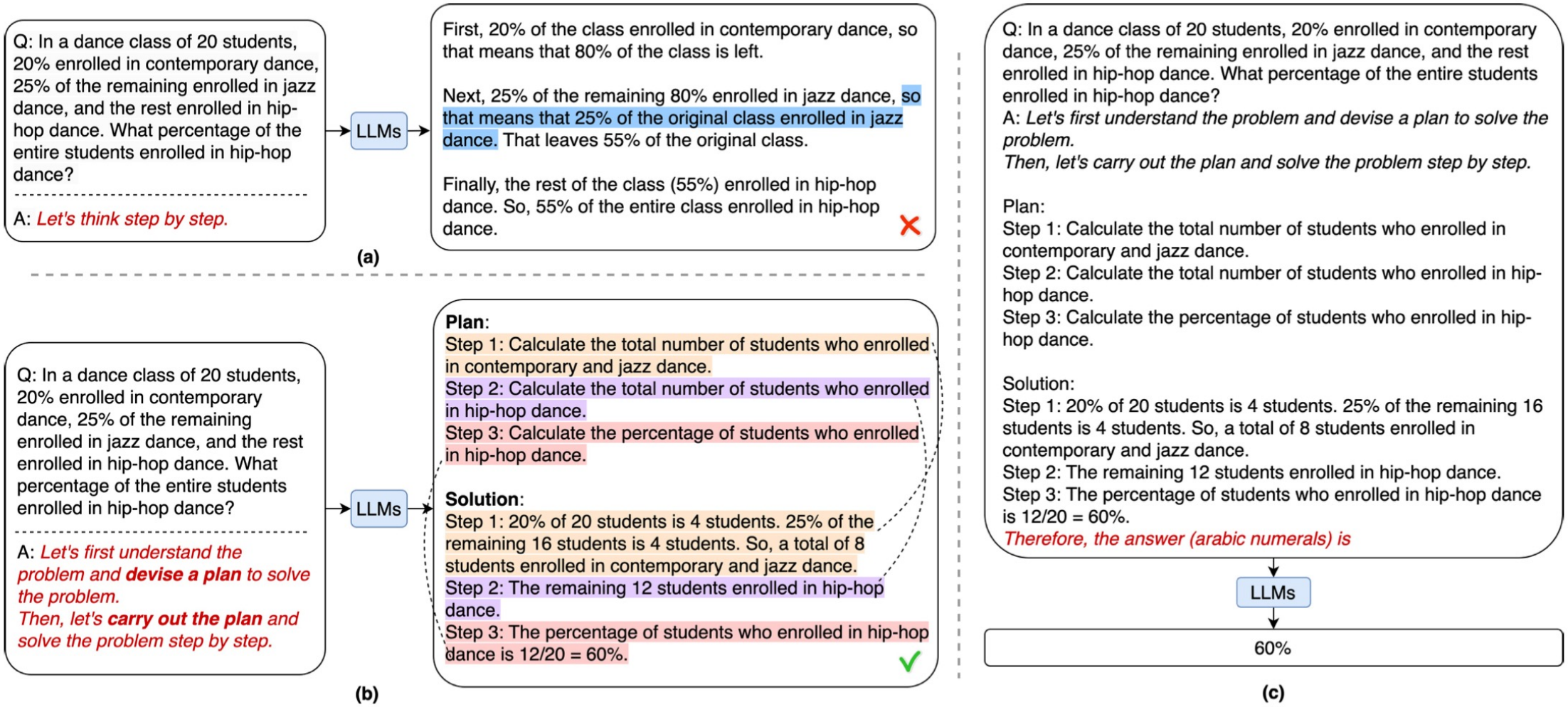


Figure 3: Self-consistency significantly outperforms sample-and-rank with the same # of samples.



Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models



Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models

Table 2: Accuracy comparison on six math reasoning datasets. The best and second best results are boldfaced and underlined respectively.

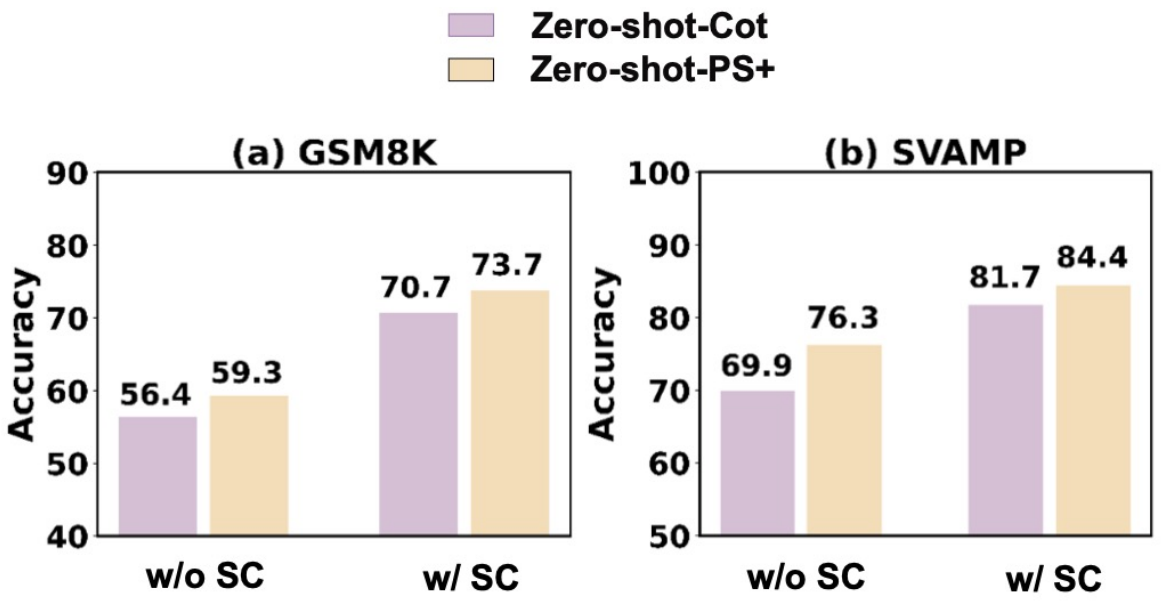
Setting	Method (text-davinci-003)	MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	Average
Zero-Shot	CoT	83.8	56.4	85.3	38.9	88.1	69.9	70.4
	PoT	**92.2**	57.0	85.1	<u>43.9</u>	<u>91.7</u>	70.8	<u>73.5</u>
	PS (ours)	87.2	<u>58.2</u>	<u>88.1</u>	42.5	89.2	<u>72.0</u>	72.9
	PS+ (ours)	<u>91.8</u>	**59.3**	**92.2**	**46.0**	**94.7**	**75.7**	**76.7**
Few-Shot	Manual-CoT	93.6	58.4	91.6	48.4	93.5	80.3	77.6
	Auto-CoT	95.5	57.1	90.8	41.7	92.1	78.1	75.9

Table 3: Accuracy on commonsense reasoning datasets.

Method	CSQA	StrategyQA
Few-Shot-CoT (Manual)	78.3	71.2
Zero-shot-CoT	65.2	63.8
Zero-shot-PS+ (ours)	**71.9**	**65.4**

Table 4: Accuracy on symbolic reasoning datasets.

Method	Last Letter	Coin Flip
Few-Shot-CoT (Manual)	70.6	100.0
Zero-shot-CoT	64.8	96.8
Zero-shot-PS+ (ours)	**75.2**	**99.6**



LoRA & QLoRA adapters

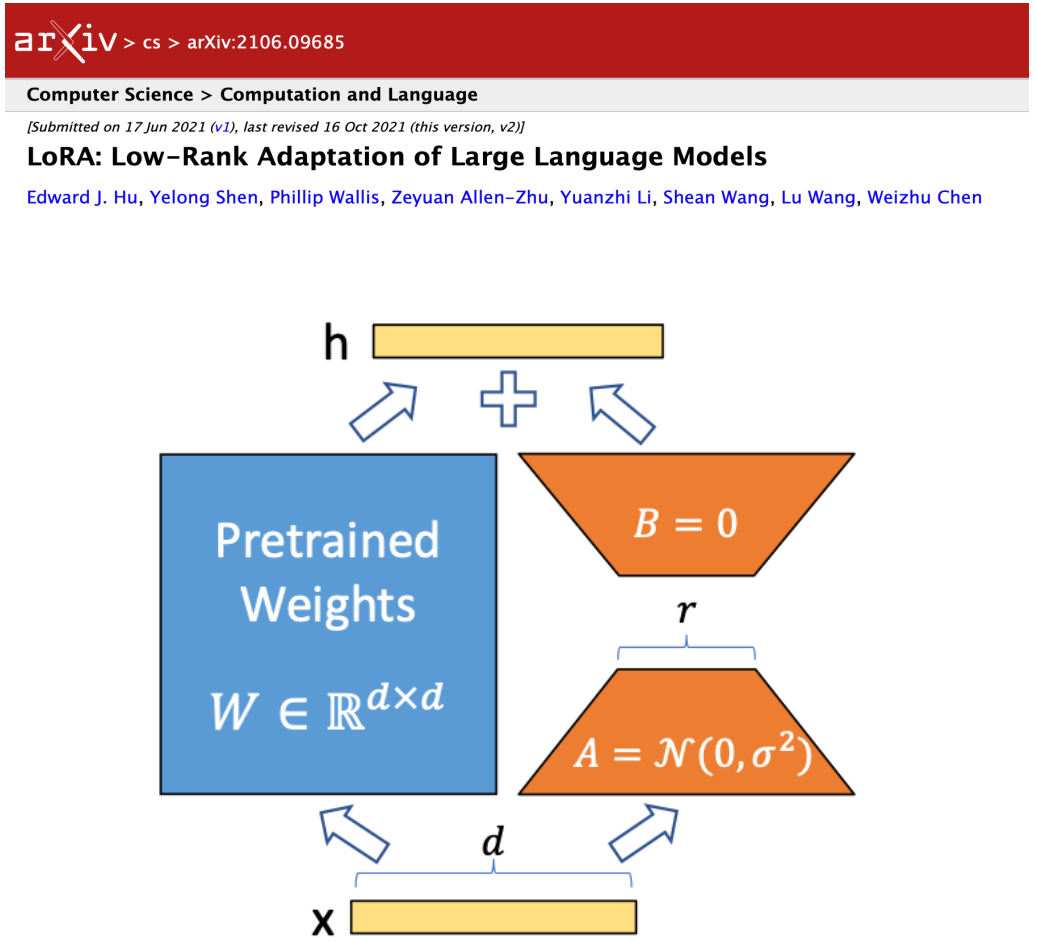
Адаптеры: LoRA

- К основным весам модели добавляется матрица
- Данная матрица составляется из произведения двух других матриц малых размерностей
- Основные веса модели заморожены и не обучаются
- Адаптер уменьшает количество обучаемых параметров до 1-3 процентов

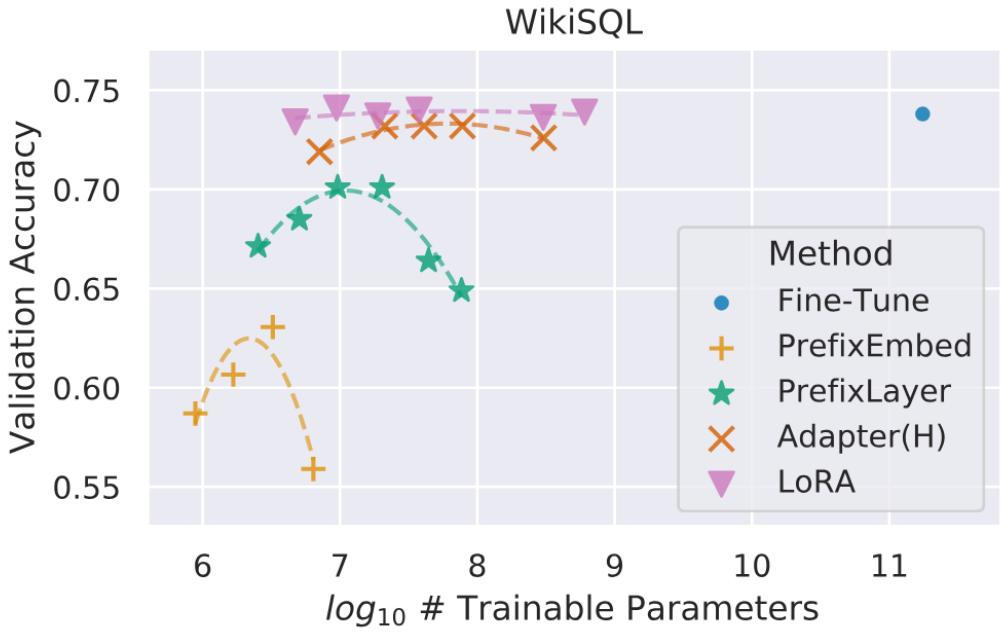
$$h = W_0 x + \Delta W x = W_0 x + BAx$$

Инициализация:

- A – нормальное распределение
- B – нулевая



Адаптеры: LoRA



Model & Method	# Trainable Parameters	E2E NLG Challenge BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)*	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M (Adapter^{L})*	0.37M	66.3	8.41	45.0	69.8	2.40
GPT-2 M (Adapter^{L})*	11.09M	68.9	8.71	46.1	71.3	2.47
GPT-2 M (Adapter^{H})	11.09M	$67.3_{\pm.6}$	$8.50_{\pm.07}$	$46.0_{\pm.2}$	$70.7_{\pm.2}$	$2.44_{\pm.01}$
GPT-2 M (FT^{Top2})*	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (PreLayer)*	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.85}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{71.8}_{\pm.1}$	$\mathbf{2.53}_{\pm.02}$
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L (Adapter^{L})	0.88M	$69.1_{\pm.1}$	$8.68_{\pm.03}$	$46.3_{\pm.0}$	$71.4_{\pm.2}$	$\mathbf{2.49}_{\pm.0}$
GPT-2 L (Adapter^{L})	23.00M	$68.9_{\pm.3}$	$8.70_{\pm.04}$	$46.1_{\pm.1}$	$71.3_{\pm.2}$	$2.45_{\pm.02}$
GPT-2 L (PreLayer)*	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (LoRA)	0.77M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.89}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{72.0}_{\pm.2}$	$2.47_{\pm.02}$

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 (Adapter^{H})	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 (Adapter^{H})	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

Адаптеры: QLoRA

arXiv > cs > arXiv:2106.09685

Computer Science > Computation and Language

[Submitted on 17 Jun 2021 (v1), last revised 16 Oct 2021 (this version, v2)]

LoRA: Low-Rank Adaptation of Large Language Models

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen

arXiv > cs > arXiv:2305.14314

Computer Science > Machine Learning

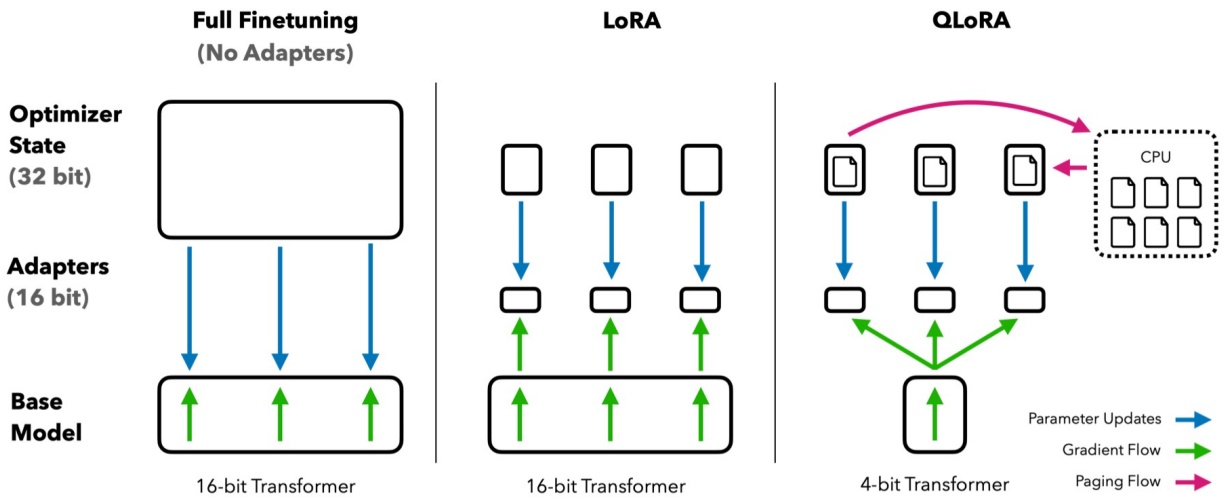
[Submitted on 23 May 2023]

QLoRA: Efficient Finetuning of Quantized LLMs

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer

LLaMA 2	FP16	4bit	8bit
7B	13.9	5.1	8.1
13B	26	8.6	14.5
70B	130	38.3	66.9

Model size, GB



Dataset	7B	13B	33B	65B
LLaMA no tuning	35.1	46.9	57.8	63.4
Self-Instruct	36.4	33.3	53.0	56.7
Longform	32.1	43.2	56.6	59.7
Chip2	34.5	41.6	53.6	59.8
HH-RLHF	34.9	44.6	55.8	60.1
Unnatural Instruct	41.9	48.1	57.3	61.3
Guanaco (OASST1)	36.6	46.4	57.0	62.2
Alpaca	38.8	47.8	57.3	62.5
FLAN v2	44.5	51.4	59.2	63.9

MMLU 5-shot test results

Адаптеры: QLoRA

Из FP32 в Int8 переводит число в целочисленный интервал [-127, 127]

$$\mathbf{X}^{\text{Int8}} = \text{round}\left(\frac{127}{\text{absmax}(\mathbf{X}^{\text{FP32}})}\mathbf{X}^{\text{FP32}}\right) = \text{round}(c^{\text{FP32}} \cdot \mathbf{X}^{\text{FP32}})$$

$$\text{dequant}(c^{\text{FP32}}, \mathbf{X}^{\text{Int8}}) = \frac{\mathbf{X}^{\text{Int8}}}{c^{\text{FP32}}} = \mathbf{X}^{\text{FP32}}$$

NormalFloat:

Вместо разбиения интервала на равные части можем предположить, что данные имеют нормальное распределение и разбить данные на интервалы по квантилям нормального распределения

Двойная квантизация – квантизуем еще и константу квантизации

Адаптеры: QLoRA

Из FP32 в Int8 переводит число в целочисленный интервал [-127, 127]

$$\mathbf{Y}^{\text{BF16}} = \mathbf{X}^{\text{BF16}} \text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{NF4}}) + \mathbf{X}^{\text{BF16}} \mathbf{L}_1^{\text{BF16}} \mathbf{L}_2^{\text{BF16}},$$

$$\text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{k-bit}}) = \text{dequant}(\text{dequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}), \mathbf{W}^{\text{4bit}}) = \mathbf{W}^{\text{BF16}},$$

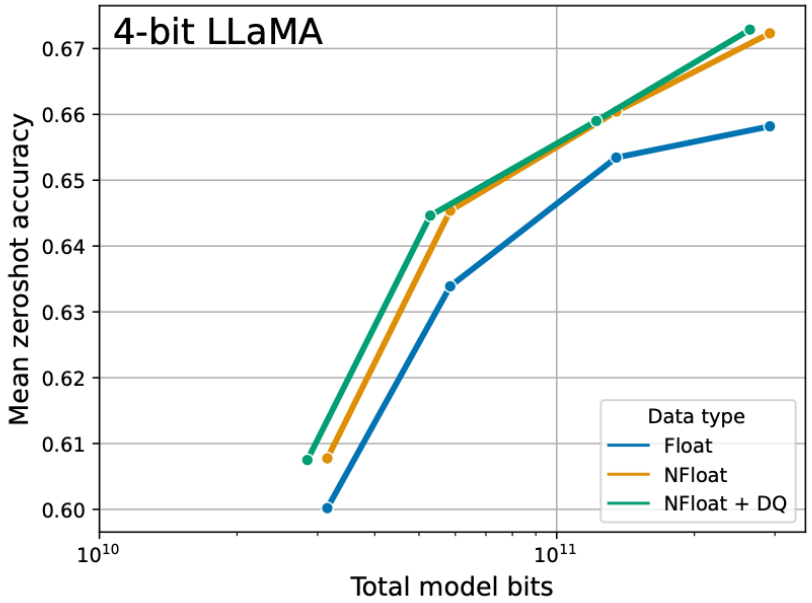


Table 3: Experiments comparing 16-bit BrainFloat (BF16), 8-bit Integer (Int8), 4-bit Float (FP4), and 4-bit NormalFloat (NF4) on GLUE and Super-NaturalInstructions. QLoRA replicates 16-bit LoRA and full-finetuning.

Dataset	GLUE (Acc.)	Super-NaturalInstructions (RougeL)				
Model	RoBERTa-large	T5-80M	T5-250M	T5-780M	T5-3B	T5-11B
BF16	88.6	40.1	42.1	48.0	54.3	62.0
BF16 replication	88.6	40.0	42.2	47.3	54.9	-
LoRA BF16	88.8	40.5	42.6	47.1	55.4	60.7
QLoRA Int8	88.8	40.4	42.9	45.4	56.5	60.7
QLoRA FP4	88.6	40.3	42.4	47.5	55.6	60.9
QLoRA NF4 + DQ	-	40.4	42.7	47.7	55.3	60.9

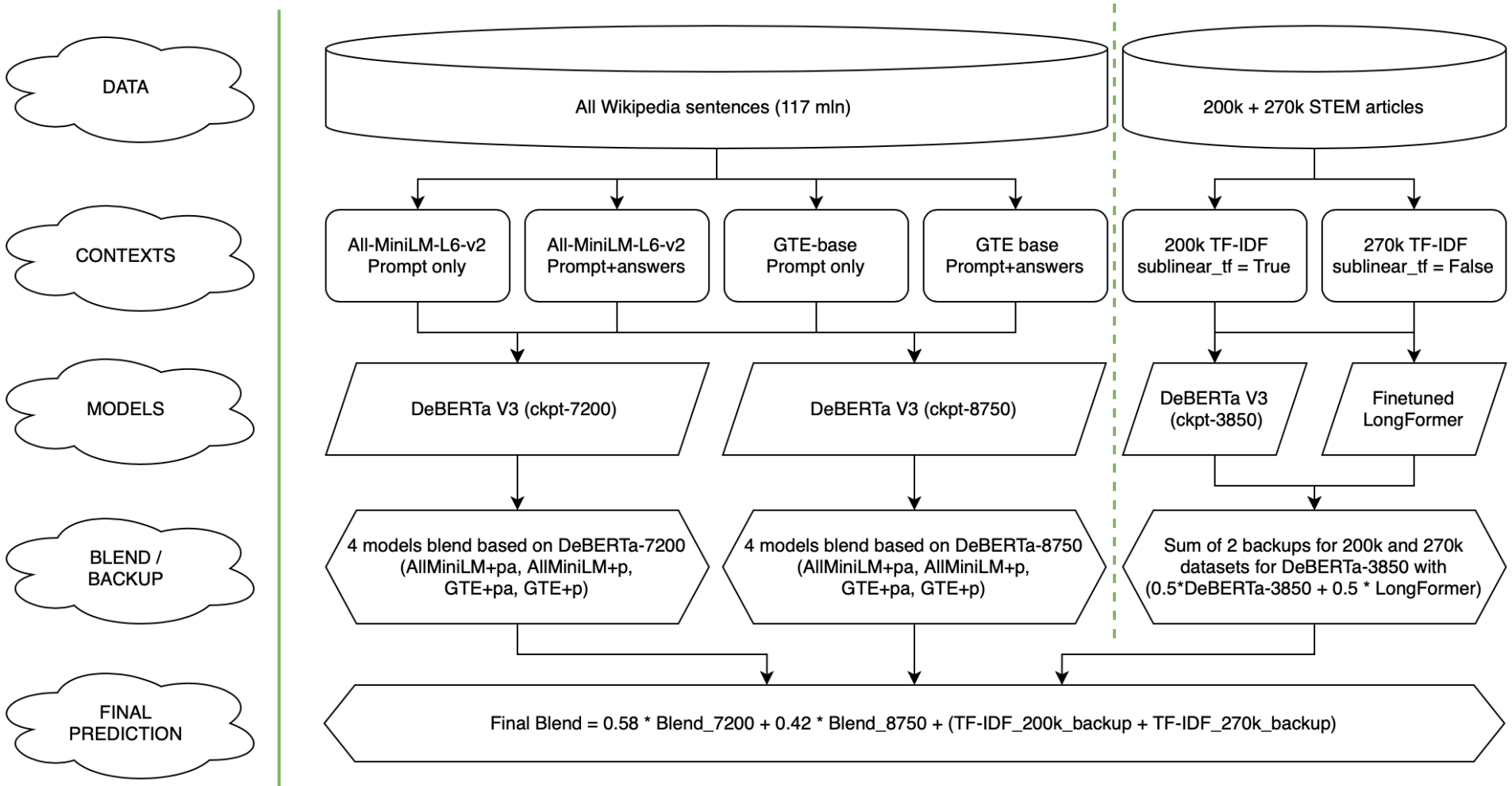
Kaggle LLM Competitions

- [Kaggle - LLM Science Exam]
- [AI Mathematical Olympiad - Progress Prize 1]
- [Learning Agency Lab - Automated Essay Scoring 2.0]
- [LLM Prompt Recovery]

#	Team	Members	Score	Entries	Last	Solution
1	Team H2O LLM Studio		0.933208	435	1mo	
2	lytic		0.931853	156	1mo	
3	Podpall		0.928415	140	1mo	
4	Preferred Scantron		0.927633	99	1mo	
5	Preferred おしゃべりんぼう(Chatty Kids)		0.926279	196	1mo	
6	RAG to riches		0.925914	274	1mo	
7	days		0.925184	366	1mo	
8	TT		0.924507	128	1mo	
9	SUSTech BDI		0.923205	391	1mo	
10	kami		0.922215	117	1mo	
11	ScienceGPT		0.921329	360	1mo	
12	kaggle is few shot learner		0.921329	302	1mo	
13	kurupical		0.921121	102	1mo	
14	Maxim Tsygankov		0.920495	89	1mo	
15	Life is full of surprise		0.920183	222	1mo	
16	CSP		0.918672	126	1mo	
17	Sadge		0.918620	110	1mo	
18	Chan Kha Vu		0.918203	53	1mo	
19	steubk		0.917838	158	1mo	
20	SkyFish Up Up		0.916901	129	1mo	
21	**Contextual kagglers**		0.916588	294	1mo	

https://www.kaggle.com/competitions/kaggle-llm-science-exam

Kaggle LLM Science Exam



https://www.kaggle.com/competitions/kaggle-llm-science-exam/discussion/447589

Q & A

Sber AI Lab

Апрель 2024