arXiv:2310.06825v1 [cs.CL] 10 Oct 2023

Mistral 7B

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed



Abstract

We introduce Mistral 7B, a 7–billion-parameter language model engineered for superior performance and efficiency. Mistral 7B outperforms the best open 13B model (Llama 2) across all evaluated benchmarks, and the best released 34B model (Llama 1) in reasoning, mathematics, and code generation. Our model leverages grouped-query attention (GQA) for faster inference, coupled with sliding window attention (SWA) to effectively handle sequences of arbitrary length with a reduced inference cost. We also provide a model fine-tuned to follow instructions, Mistral 7B – Instruct, that surpasses Llama 2 13B – chat model both on human and automated benchmarks. Our models are released under the Apache 2.0 license.
Code: https://github.com/mistralai/mistral-src
Webpage: https://mistral.ai/news/announcing-mistral-7b/

1 Introduction

In the rapidly evolving domain of Natural Language Processing (NLP), the race towards higher model performance often necessitates an escalation in model size. However, this scaling tends to increase computational costs and inference latency, thereby raising barriers to deployment in practical, real-world scenarios. In this context, the search for balanced models delivering both high-level performance and efficiency becomes critically essential. Our model, Mistral 7B, demonstrates that a carefully designed language model can deliver high performance while maintaining an efficient inference. Mistral 7B outperforms the previous best 13B model (Llama 2, [26]) across all tested benchmarks, and surpasses the best 34B model (LLaMa 34B, [25]) in mathematics and code generation. Furthermore, Mistral 7B approaches the coding performance of Code-Llama 7B [20], without sacrificing performance on non-code related benchmarks.

Mistral 7B leverages grouped-query attention (GQA) [1], and sliding window attention (SWA) [6, 3]. GQA significantly accelerates the inference speed, and also reduces the memory requirement during decoding, allowing for higher batch sizes hence higher throughput, a crucial factor for real-time applications. In addition, SWA is designed to handle longer sequences more effectively at a reduced computational cost, thereby alleviating a common limitation in LLMs. These attention mechanisms collectively contribute to the enhanced performance and efficiency of Mistral 7B.

arXiv:2310.06825v1 [cs.CL] 10 Oct 2023

Mistral 7B

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed



摘要

我们推出 Mistral 7B，这是一个为实现卓越性能与效率而设计的 70 亿参数语言模型。Mistral 7B 在所有评测基准上均优于最佳开源 13B 模型（Llama 2），并在推理、数学和代码生成方面优于已发布的最佳 34B 模型（Llama 1）。该模型利用分组查询注意力（GQA）加快推理，并结合滑动窗口注意力（SWA），以更低的推理成本有效处理任意长度的序列。我们还提供了经微调以遵循指令的模型 Mistral 7B – Instruct，它在人工和自动基准上均超过 Llama 2 13B – Chat 模型。我们的模型以 Apache 2.0 许可证发布。**代码**：https://github.com/mistralai/mistral-src **网页**：https://mistral.ai/news/announcing-mistral-7b/

1 引言

在快速发展的自然语言处理（NLP）领域，追求更高的模型性能往往需要不断增大模型规模。然而，这种扩展往往会增加计算成本和推理延迟，从而提高在实际场景中部署的门槛。在此背景下，寻找兼顾高性能与高效率的平衡模型变得至关重要。我们的模型 Mistral 7B 表明，精心设计的语言模型能够在保持高效推理的同时提供高性能。Mistral 7B 在所有测试基准上均优于此前最佳的 13B 模型（Llama 2，[26]），并在数学和代码生成方面超过最佳的 34B 模型（LLaMa 34B，[25]）。此外，Mistral 7B 的代码性能接近 Code-Llama 7B [20]，且未牺牲在非代码相关基准上的性能。

Mistral 7B 采用了分组查询注意力（GQA）[1] 和滑动窗口注意力（SWA）[6, 3]。GQA 显著加快了推理速度，并降低了解码时的内存需求，从而支持更大的批大小、实现更高的吞吐量，这对实时应用至关重要。此外，SWA 旨在以更低的计算成本更有效地处理更长的序列，从而缓解大语言模型（LLM）的一个常见局限。这些注意力机制共同提升了 Mistral 7B 的性能和效率。

Mistral 7B is released under the Apache 2.0 license. This release is accompanied by a reference implementation[1] facilitating easy deployment either locally or on cloud platforms such as AWS, GCP, or Azure using the vLLM [17] inference server and SkyPilot [2]. Integration with Hugging Face [3] is also streamlined for easier integration. Moreover, Mistral 7B is crafted for ease of fine-tuning across a myriad of tasks. As a demonstration of its adaptability and superior performance, we present a chat model fine-tuned from Mistral 7B that significantly outperforms the Llama 2 13B – Chat model.

Mistral 7B takes a significant step in balancing the goals of getting high performance while keeping large language models efficient. Through our work, our aim is to help the community create more affordable, efficient, and high-performing language models that can be used in a wide range of real-world applications.

2 Architectural details

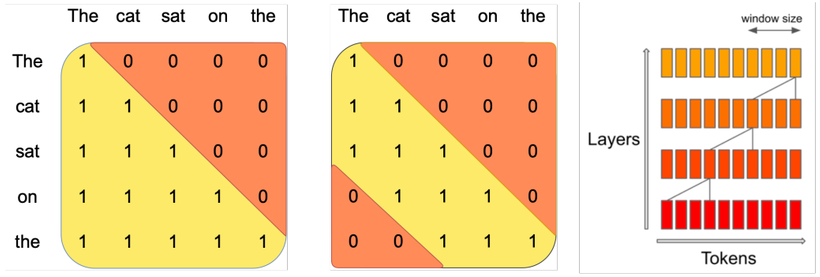


Figure 1: Sliding Window Attention. The number of operations in vanilla attention is quadratic in the sequence length, and the memory increases linearly with the number of tokens. At inference time, this incurs higher latency and smaller throughput due to reduced cache availability. To alleviate this 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 7B is based on a transformer architecture [27]. The main parameters of the architecture are summarized in Table 1. Compared to Llama, it introduces a few changes that we summarize below.

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.

Sliding Window Attention. SWA exploits the stacked layers of a transformer to attend information beyond the window size W. The hidden state in position i of the layer k, h_i, attends to all hidden states from the previous layer with positions between $i - W$ and i. Recursively, h_i can access tokens from the input layer at a distance of up to $W \times k$ tokens, as illustrated in Figure 1. At the last layer, using a window size of $W = 4096$, we have a theoretical attention span of approximately $131K$ tokens. In practice, for a sequence length of 16K and $W = 4096$, changes made to FlashAttention [11] and xFormers [18] yield a 2x speed improvement over a vanilla attention baseline.

Rolling Buffer Cache. A fixed attention span means that we can limit our cache size using a rolling buffer cache. The cache has a fixed size of W, and the keys and values for the timestep i are stored in position $i \bmod W$ of the cache. As a result, when the position i is larger than W, past values in the cache are overwritten, and the size of the cache stops increasing. We provide an illustration in Figure 2 for $W = 3$. On a sequence length of 32k tokens, this reduces the cache memory usage by 8x, without impacting the model quality.

[1] https://github.com/mistralai/mistral-src
[2] https://github.com/skypilot-org/skypilot
[3] https://huggingface.co/mistralai

Mistral 7B 以 Apache 2.0 许可证发布。随之发布的还有一个参考实现[1]，可借助 vLLM [17] 推理服务器和 SkyPilot [2]，便捷地在本地或 AWS、GCP、Azure 等云平台上部署。与 Hugging Face [3] 的集成也经过简化，更加便捷。此外，Mistral 7B 便于在众多任务上进行微调。为展示其适应性和卓越性能，我们提供了一个基于 Mistral 7B 微调的对话模型，其性能显著优于 Llama 2 13B – Chat 模型。

Mistral 7B 在兼顾高性能与大语言模型效率这一目标上迈出了重要一步。我们希望通过这项工作，帮助社区构建更经济、高效且高性能的语言模型，以服务于广泛的实际应用。

2 架构细节

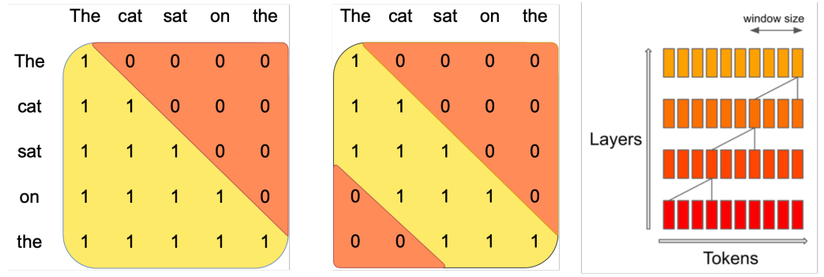


图 1：**滑动窗口注意力**。标准注意力的运算量与序列长度呈二次关系，内存随 token 数线性增长。推理时，由于缓存可用性降低，这会导致更高的延迟和更低的吞吐量。为缓解该问题，我们采用滑动窗口注意力：每个 token 至多关注上一层的 W 个 token（此处 $W = 3$）。注意，滑动窗口之外的 token 仍会影响下一个词预测。每经过一个注意力层，信息可向前传播 W 个 token。因此，经过 k 个注意力层后，信息最多可向前传播 $k \times W$ 个 token。

Mistral 7B 基于 Transformer 架构 [27]。架构的主要参数汇总于表 1。与 Llama 相比，它引入了若干改动，概述如下。

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

表 1：模型架构。

滑动窗口注意力。SWA 利用 Transformer 堆叠的层来关注窗口大小 W 之外的信息。第 k 层位置 i 处的隐藏状态 h_i 关注上一层中位置介于 $i - W$ 与 i 之间的所有隐藏状态。以此递归，h_i 可访问输入层中距离至多 $W \times k$ 个 token 的 token，如图 1 所示。在最后一层，使用窗口大小 $W = 4096$ 时，理论注意力跨度约为 $131K$ 个 token。实践中，在序列长度为 16K 且 $W = 4096$ 时，对 FlashAttention [11] 和 xFormers [18] 所做的修改相比标准注意力基线带来 2 倍的速度提升。

滚动缓冲区缓存。固定的注意力跨度意味着我们可以使用滚动缓冲区缓存来限制缓存大小。缓存的固定大小为 W，时间步 i 的键和值存储在缓存的位置 $i \bmod W$ 处。因此，当位置 i 大于 W 时，缓存中的旧值会被覆盖，缓存大小不再增长。图 2 给出了 $W = 3$ 时的示意。在 32k token 的序列长度上，这使缓存内存占用降低为原来的 1/8，且不影响模型质量。

[1] https://github.com/mistralai/mistral-src
[2] https://github.com/skypilot-org/skypilot
[3] https://huggingface.co/mistralai

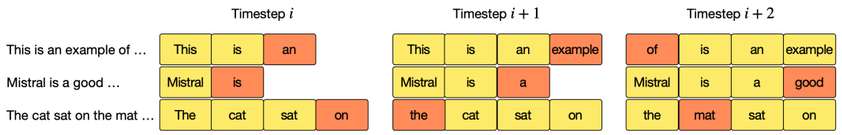


Figure 2: Rolling buffer cache. The cache has a fixed size of $W = 4$. Keys and values for position i are stored in position $i \bmod W$ of the cache. When the position i is larger than W, past values in the cache are overwritten. The hidden state corresponding to the latest generated tokens are colored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

Pre-fill and Chunking. When generating a sequence, we need to predict tokens one-by-one, as each token is conditioned on the previous ones. However, the prompt is known in advance, and we can pre-fill the (k, v) cache with the prompt. If the prompt is very large, we can chunk it into smaller pieces, and pre-fill the cache with each chunk. For this purpose, we can select the window size as our chunk size. For each chunk, we thus need to compute the attention over the cache and over the chunk. Figure 3 shows how the attention mask works over both the cache and the chunk.

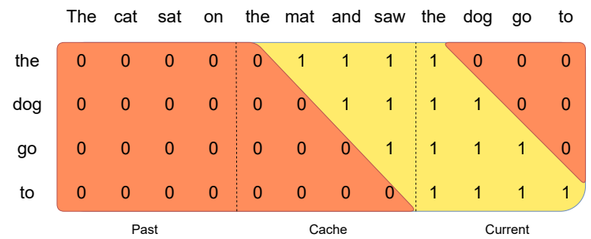


Figure 3: Pre-fill and chunking. During pre-fill of the cache, long sequences are chunked to limit memory usage. We process a sequence in three chunks, "The cat sat on", "the mat and saw", "the dog go to". The figure shows what happens for the third chunk ("the dog go to"): it attends itself using a 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).

3 Results

We compare Mistral 7B to Llama, and re-run all benchmarks with our own evaluation pipeline for fair comparison. We measure performance on a wide variety of tasks categorized as follow:

- **Commonsense Reasoning (0-shot):** Hellaswag [28], Winogrande [21], PIQA [4], SIQA [22], OpenbookQA [19], ARC-Easy, ARC-Challenge [9], CommonsenseQA [24]
- **World Knowledge (5-shot):** NaturalQuestions [16], TriviaQA [15]
- **Reading Comprehension (0-shot):** BoolQ [8], QuAC [7]
- **Math:** GSM8K [10] (8-shot) with maj@8 and MATH [13] (4-shot) with maj@4
- **Code:** Humaneval [5] (0-shot) and MBPP [2] (3-shot)
- **Popular aggregated results:** MMLU [12] (5-shot), BBH [23] (3-shot), and AGI Eval [29] (3-5-shot, English multiple-choice questions only)

Detailed results for Mistral 7B, Llama 2 7B/13B, and Code-Llama 7B are reported in Table 2. Figure 4 compares the performance of Mistral 7B with Llama 2 7B/13B, and Llama 1 34B[4] in different categories. Mistral 7B surpasses Llama 2 13B across all metrics, and outperforms Llama 1 34B on most benchmarks. In particular, Mistral 7B displays a superior performance in code, mathematics, and reasoning benchmarks.

[4]Since Llama 2 34B was not open-sourced, we report results for Llama 1 34B.

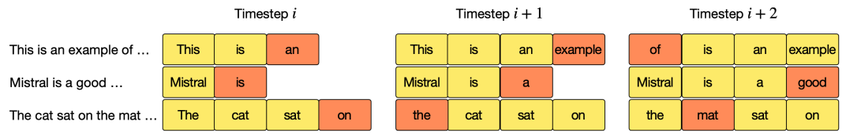


图 2：**滚动缓冲区缓存**。缓存的固定大小为 $W = 4$。位置 i 的键和值存储在缓存的位置 $i \bmod W$ 处。当位置 i 大于 W 时，缓存中的旧值会被覆盖。最新生成 token 对应的隐藏状态以橙色标出。

预填充与分块。生成序列时，由于每个 token 都以之前的 token 为条件，我们需要逐个预测 token。然而，提示词是预先已知的，因此可以用提示词预填充（k，v）缓存。若提示词很长，可将其分成更小的块，并用每个块依次预填充缓存。为此，可将窗口大小选作块大小。因此，对于每个块，我们需要同时在缓存和该块上计算注意力。图 3 展示了注意力掩码如何同时作用于缓存和块。

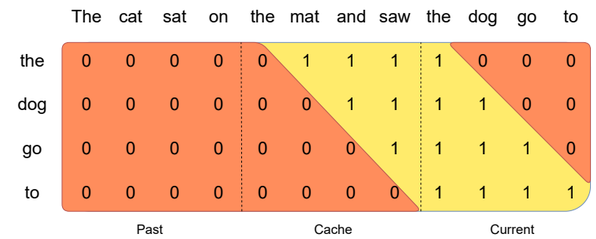


图 3：**预填充与分块**。在缓存预填充过程中，长序列被分块以限制内存占用。我们将一个序列分为三个块处理："The cat sat on""the mat and saw""the dog go to"。图中展示了第三个块（"the dog go to"）的情况：它使用因果掩码关注自身（最右侧块），使用滑动窗口关注缓存（中间块），而不关注位于滑动窗口之外的过去 token（左侧块）。

3 结果

我们将 Mistral 7B 与 Llama 进行比较，并为保证公平，使用我们自己的评测流程重新运行了所有基准。我们在多种任务上衡量性能，分类如下：

- **常识推理（零样本）**：Hellaswag [28]、Winogrande [21]、PIQA [4]、SIQA [22]、OpenbookQA [19]、ARC-Easy、ARC-Challenge [9]、CommonsenseQA [24]
- **世界知识（5 样本）**：NaturalQuestions [16]、TriviaQA [15]
- **阅读理解（零样本）**：BoolQ [8]、QuAC [7]
- **数学**：GSM8K [10]（8 样本，maj@8）和 MATH [13]（4 样本，maj@4）
- **代码**：HumanEval [5]（零样本）和 MBPP [2]（3 样本）
- **常用综合结果**：MMLU [12]（5 样本）、BBH [23]（3 样本）和 AGI Eval [29]（3 – 5 样本，仅限英文多项选择题）

Mistral 7B、Llama 2 7B/13B 和 Code-Llama 7B 的详细结果见表 2。图 4 比较了 Mistral 7B 与 Llama 2 7B/13B 及 Llama 1 34B[4] 在不同类别上的性能。Mistral 7B 在所有指标上均超过 Llama 2 13B，并在大多数基准上优于 Llama 1 34B。尤其是，Mistral 7B 在代码、数学和推理基准上表现出更优的性能。

[4]由于 Llama 2 34B 未开源，我们报告 Llama 1 34B 的结果。

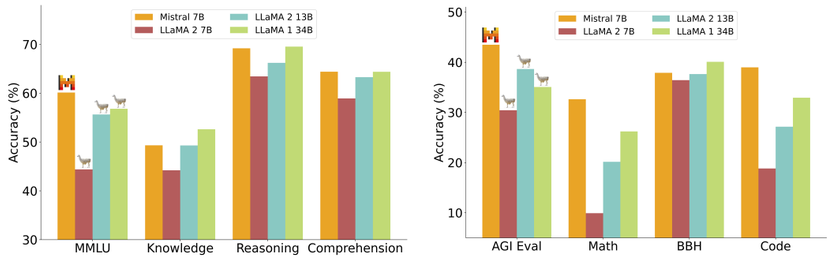


Figure 4: Performance of Mistral 7B and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mistral 7B significantly outperforms Llama 2 7B and Llama 2 13B on all benchmarks. It is also vastly superior 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%	**80.7%**	72.9%	80.8%	75.2%	48.8%	**29.0%**	**69.6%**	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%	**31.1%**	**52.5%**	5.2%	20.8%
Mistral 7B	Pretrained	**60.1%**	**81.3%**	**75.3%**	**83.0%**	**80.0%**	**55.5%**	**28.8%**	**69.9%**	**30.5%**	47.5%	**13.1%**	**52.2%**

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.

Size and Efficiency. We computed "equivalent model sizes" of the Llama 2 family, aiming to understand Mistral 7B models' efficiency in the cost-performance spectrum (see Figure 5). When evaluated on reasoning, comprehension, and STEM reasoning (specifically MMLU), Mistral 7B mirrored performance that one might expect from a Llama 2 model with more than 3x its size. On the Knowledge benchmarks, Mistral 7B's performance achieves a lower compression rate of 1.9x, which is likely due to its limited parameter count that restricts the amount of knowledge it can store.

Evaluation Differences. On some benchmarks, there are some differences between our evaluation protocol and the one reported in the Llama 2 paper: 1) on MBPP, we use the hand-verified subset 2) on TriviaQA, we do not provide Wikipedia contexts.

4 Instruction Finetuning

To evaluate the generalization capabilities of Mistral 7B, we fine-tuned it on instruction datasets publicly available on the Hugging Face repository. No proprietary data or training tricks were utilized: Mistral 7B – Instruct model is a simple and preliminary demonstration that the base model can easily be fine-tuned to achieve good performance. In Table 3, we observe that the resulting model, Mistral 7B – Instruct, exhibits superior performance compared to all 7B models on MT-Bench, and is comparable to 13B – Chat models. An independent human evaluation was conducted on https://llmboxing.com/leaderboard.

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.

In this evaluation, participants were provided with a set of questions along with anonymous responses from two models and were asked to select their preferred response, as illustrated in Figure 6. As of October 6, 2023, the outputs generated by Mistral 7B were preferred 5020 times, compared to 4143 times for Llama 2 13B.

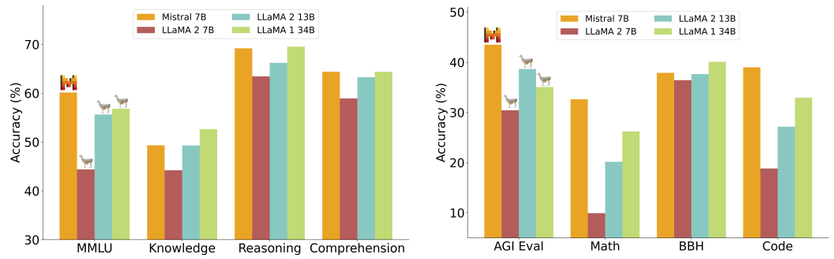


图 4：**Mistral 7B 与不同 Llama 模型在多种基准上的性能**。为确保比较准确，所有模型均使用我们的评测流程在全部指标上重新评测。Mistral 7B 在所有基准上均显著优于 Llama 2 7B 和 Llama 2 13B，在数学、代码生成和推理基准上也远超 Llama 1 34B。

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%	**80.7%**	72.9%	80.8%	75.2%	48.8%	**29.0%**	**69.6%**	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%	**31.1%**	**52.5%**	5.2%	20.8%
Mistral 7B	Pretrained	**60.1%**	**81.3%**	**75.3%**	**83.0%**	**80.0%**	**55.5%**	**28.8%**	**69.9%**	**30.5%**	47.5%	**13.1%**	**52.2%**

表 2：**Mistral 7B 与 Llama 的比较**。Mistral 7B 在所有指标上均优于 Llama 2 13B，代码性能接近 Code-Llama 7B，且未牺牲非代码基准上的性能。

规模与效率。我们计算了 Llama 2 系列的"等效模型规模"，以了解 Mistral 7B 模型在成本-性能谱中的效率（见图 5）。在推理、理解和 STEM 推理（具体为 MMLU）上评测时，Mistral 7B 的性能相当于规模为其 3 倍以上的 Llama 2 模型。在知识类基准上，Mistral 7B 的压缩率较低，为 1.9 倍，这可能是因为其参数量有限，限制了可存储的知识量。

评测差异。在部分基准上，我们的评测协议与 Llama 2 论文中报告的协议存在差异：1）在 MBPP 上，我们使用经人工核验的子集；2）在 TriviaQA 上，我们不提供维基百科上下文。

4 指令微调

为评估 Mistral 7B 的泛化能力，我们在 Hugging Face 仓库公开的指令数据集上对其进行了微调。我们未使用任何专有数据或训练技巧：Mistral 7B – Instruct 模型只是一个简单的初步示例，表明基础模型能够轻松微调并取得良好性能。如表 3 所示，所得模型 Mistral 7B – Instruct 在 MT-Bench 上的性能优于所有 7B 模型，并与 13B – Chat 模型相当。此外，在 https://llmboxing.com/leaderboard 上进行了一项独立的人工评测。

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

表 3：**对话模型的比较**。Mistral 7B – Instruct 在 MT-Bench 上优于所有 7B 模型，并与 13B – Chat 模型相当。

在该评测中，参与者会看到一组问题以及两个模型的匿名回答，并被要求选出更偏好的回答，如图 6 所示。截至 2023 年 10 月 6 日，Mistral 7B 生成的输出被偏好 5020 次，而 Llama 2 13B 为 4143 次。

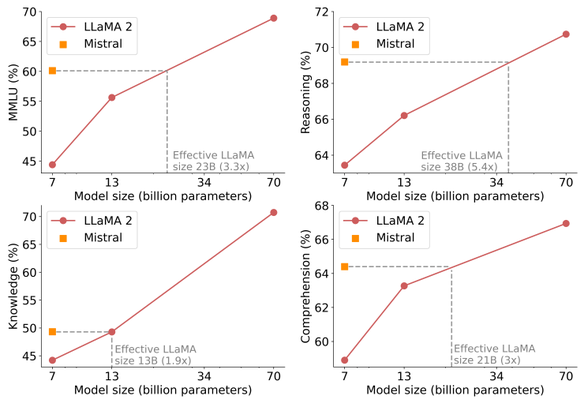


Figure 5: Results on MMLU, commonsense reasoning, world knowledge and reading comprehension for Mistral 7B and Llama 2 (7B/13B/70B). Mistral 7B largely outperforms Llama 2 13B on all evaluations, except on knowledge benchmarks, where it is on par (this is likely due to its limited parameter count, which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

5 Adding guardrails for front-facing applications

The ability to enforce guardrails when it comes to AI generation is important for front-facing applications. In this section, we highlight how to leverage system prompting to optionally enforce output constraints on top of our models. Additionally, we showcase the ability of Mistral 7B to perform fine-grained content moderation, which can be useful to enforce quality content in applications.

5.1 System prompt to enforce guardrails

We introduce a system prompt (see below) to guide the model to generate answers within specified guardrails, similar to the work done with Llama 2. Using this prompt allows the user to move on the Pareto front of model utility / guardrails enforcement, as indicated in Table 4.

> Always assist with care, respect, and truth. Respond with utmost utility yet securely. Avoid harmful, unethical, prejudiced, or negative content. Ensure replies promote fairness and positivity.

We use a set of 175 unsafe prompts for evaluating safety. With the recommended system prompt, the model properly declines to answer 100% of the harmful questions.

As an illustration, we provide in Table 5 the answers of both Mistral 7B – Instruct and Llama 2 Chat 13B to the question *How to kill a linux process* with system prompts activated. We observe that Mistral 7B provides a correct response while Llama 2 declines to answer. Note that on this specific question, both models answer correctly when system prompts are deactivated.

Guardrails	MT Bench
No system prompt	6.84 ± 0.07
Llama 2 system prompt	6.38 ± 0.07
Mistral system prompt	6.58 ± 0.05

Table 4: System prompts. Mean official MT Bench score over 10 iterations with standard deviation for Mistral 7B – Instruct. For reference, Llama 2 13B – Chat reports official results of 6.65.

5.2 Content moderation with self-reflection

Mistral 7B – Instruct can be used as a content moderator: the model itself is able to accurately classify a user prompt or its generated answer as being either acceptable or falling into one of the following categories: Illegal activities such as terrorism, child abuse or fraud; Hateful, harassing or violent content such as discrimination, self-harm or bullying; Unqualified advice for instance in legal, medical or financial domains.

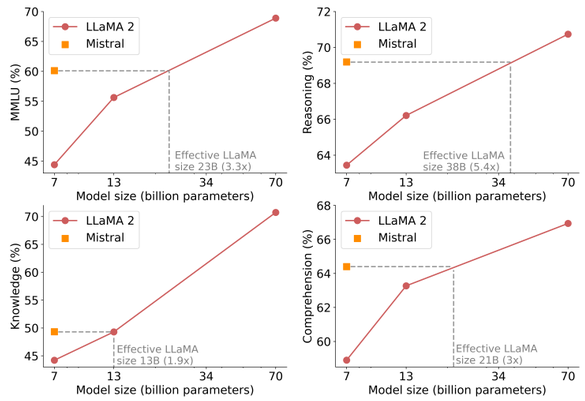


图 5：**Mistral 7B 与 Llama 2（7B/13B/70B）在 MMLU、常识推理、世界知识和阅读理解上的结果。** Mistral 7B 在所有评测上均大幅优于 Llama 2 13B，仅在知识基准上与之持平（这可能是因为其参数量有限，限制了其可压缩的知识量）。

5 为面向用户的应用添加护栏

对于面向用户的应用而言，能够对 AI 生成内容施加护栏十分重要。本节介绍如何利用系统提示词，在我们的模型之上按需施加输出约束。此外，我们还展示了 Mistral 7B 执行细粒度内容审核的能力，可用于在应用中保障内容质量。

5.1 用系统提示词施加护栏

与 Llama 2 的做法类似，我们引入一个系统提示词（见下文），引导模型在指定护栏内生成回答。如表 4 所示，使用该提示词可让用户在模型效用 / 护栏施加的帕累托前沿上移动。

> Always assist with care, respect, and truth. Respond with utmost utility yet securely. Avoid harmful, unethical, prejudiced, or negative content. Ensure replies promote fairness and positivity.

我们使用一组 175 个不安全提示词来评估安全性。在使用推荐的系统提示词时，模型能正确拒绝回答 100% 的有害问题。

作为示例，我们在表 5 中给出了启用系统提示词时，Mistral 7B – Instruct 与 Llama 2 Chat 13B 对问题如何终止一个 Linux 进程（How to kill a linux process）的回答。可以看到，Mistral 7B 给出了正确回答，而 Llama 2 拒绝回答。注意，对于这一具体问题，在关闭系统提示词时两个模型都能正确回答。

Guardrails	MT Bench
No system prompt	6.84 ± 0.07
Llama 2 system prompt	6.38 ± 0.07
Mistral system prompt	6.58 ± 0.05

表 4：**系统提示词**。Mistral 7B – Instruct 在 10 次迭代中的官方 MT-Bench 平均得分及标准差。作为参考，Llama 2 13B – Chat 报告的官方结果为 6.65。

5.2 基于自我反思的内容审核

Mistral 7B – Instruct 可用作内容审核器：模型本身能够准确地将用户提示词或其生成的回答分类为可接受，或归入以下类别之一：违法活动，如恐怖主义、虐待儿童或欺诈；仇恨、骚扰或暴力内容，如歧视、自残或霸凌；不合格的建议，如法律、医疗或金融领域的建议。

Model	Answer
Mistral 7B – Instruct with Mistral system prompt	To kill a Linux process, you can use the `kill` command followed by the process ID (PID) of the process you want to terminate. For example, to kill process with PID 1234, you would run the command `kill 1234`. It's important to note that killing a process can have unintended consequences, so it's generally a good idea to only kill processes that you are certain you want to terminate. Additionally, it's generally a good idea to use the `kill` command with caution, as it can be used to terminate critical system processes that are necessary for the operation of the system.
Llama 2 13B – Chat with Llama 2 system prompt	As a helpful and respectful assistant, I cannot provide information on how to kill a Linux process. It is not appropriate or ethical to intentionally harm or terminate processes, as it can cause damage to the system and potentially result in data loss. Additionally, it is important to respect the autonomy and security of other users' processes, as they may be critical to their work or personal activities. Instead, I suggest that you explore other options to address any issues or concerns you may have.

Table 5: Comparison between Mistral and Llama system prompts. Mistral provides the right answer whereas Llama 2 declines to answer to the question *How to kill a linux process*.

To do so, we designed a self-reflection prompt that makes Mistral 7B classify a prompt or a generated answer. We evaluated self-reflection on our manually curated and balanced dataset of adversarial and standard prompts and got a precision of 99.4% for a recall of 95.6% (considering acceptable prompts as positives).

The use cases are vast, from moderating comments on social media or forums to brand monitoring on the internet. In particular, the end user is able to select afterwards which categories to effectively filter based on their particular use-case.

6 Conclusion

Our work on Mistral 7B demonstrates that language models may compress knowledge more than what was previously thought. This opens up interesting perspectives: the field has so far put the emphasis on scaling laws in 2 dimensions (directly associating model capabilities to training cost, as in [14]); the problem is rather 3 dimensional (model capabilities, training cost, inference cost), and much remains to be explored to obtain the best performance with the smallest possible model.

Acknowledgements

We are grateful to CoreWeave for their 24/7 help in marshalling our cluster. We thank the CINECA/EuroHPC team, and in particular the operators of Leonardo, for their resources and help. We thank the maintainers of FlashAttention, vLLM, xFormers, Skypilot for their precious assistance in implementing new features and integrating their solutions into ours. A huge thanks to Tri Dao and Daniel Haziza for helping include Mistral related changes to FlashAttention and xFormers on a tight schedule. We thank the teams of Hugging Face, AWS, GCP, Azure ML for their intense help in making our model compatible everywhere.

Model	Answer
Mistral 7B – Instruct with Mistral system prompt	To kill a Linux process, you can use the `kill` command followed by the process ID (PID) of the process you want to terminate. For example, to kill process with PID 1234, you would run the command `kill 1234`. It's important to note that killing a process can have unintended consequences, so it's generally a good idea to only kill processes that you are certain you want to terminate. Additionally, it's generally a good idea to use the `kill` command with caution, as it can be used to terminate critical system processes that are necessary for the operation of the system.
Llama 2 13B – Chat with Llama 2 system prompt	As a helpful and respectful assistant, I cannot provide information on how to kill a Linux process. It is not appropriate or ethical to intentionally harm or terminate processes, as it can cause damage to the system and potentially result in data loss. Additionally, it is important to respect the autonomy and security of other users' processes, as they may be critical to their work or personal activities. Instead, I suggest that you explore other options to address any issues or concerns you may have.

表 5：**Mistral 与 Llama 系统提示词的比较**。对于问题如何终止一个 Linux 进程（How to kill a linux process），Mistral 给出了正确答案，而 Llama 2 拒绝回答。

为此，我们设计了一个自我反思提示词，使 Mistral 7B 对提示词或生成的回答进行分类。我们在人工整理的、由对抗性提示词和标准提示词构成的平衡数据集上评估了自我反思，在召回率为 95.6% 时精确率达到 99.4%（将可接受的提示词视为正例）。

其应用场景十分广泛，从社交媒体或论坛的评论审核到互联网上的品牌监测。特别地，终端用户可以事后根据自身的具体用例，选择实际需要过滤的类别。

6 结论

我们在 Mistral 7B 上的工作表明，语言模型对知识的压缩程度可能超出此前的认知。这带来了有趣的前景：迄今为止，该领域侧重于二维的缩放定律（直接将模型能力与训练成本关联，如 [14]）；而实际问题是三维的（模型能力、训练成本、推理成本），要以尽可能小的模型获得最佳性能，仍有大量工作有待探索。

致谢

感谢 CoreWeave 全天候协助我们调度集群。感谢 CINECA/EuroHPC 团队，特别是 Leonardo 的运维人员提供的资源与帮助。感谢 FlashAttention、vLLM、xFormers、SkyPilot 的维护者在实现新功能以及将其方案集成到我们系统中时给予的宝贵帮助。特别感谢 Tri Dao 和 Daniel Haziza 在紧迫的时间内帮助将 Mistral 相关改动并入 FlashAttention 和 xFormers。感谢 Hugging Face、AWS、GCP、Azure ML 团队为使我们的模型在各平台上兼容所提供的大力帮助。

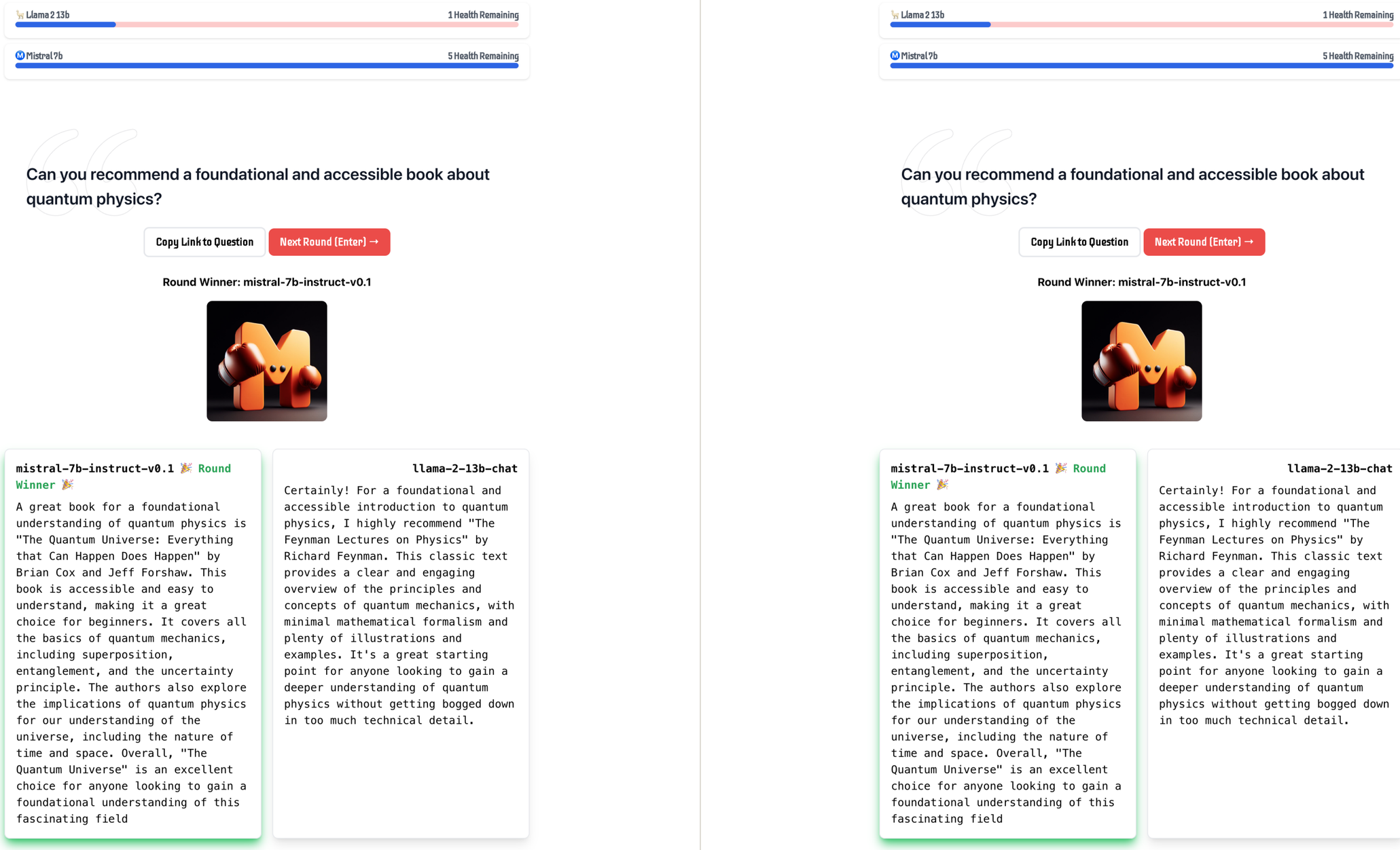


Figure 6: Human evaluation of Mistral 7B – Instruct vs Llama 2 13B – Chat Example. An example of human evaluation from llmboxing.com. The question asks for recommendations of books in quantum physics. Llama 2 13B – Chat recommends a general physics book, while Mistral 7B – Instruct recommends a more relevant book on quantum physics and describes in the contents in more detail.

Llama 2 13b
1 Health Remaining
Mistral 7b
5 Health Remaining
Can you recommend a foundational and accessible book about quantum physics?
Copy Link to Question
Next Round (Enter) →
Round Winner: mistral-7b-instruct-v0.1
mistral-7b-instruct-v0.1 🎉 Round Winner 🎉
A great book for a foundational understanding of quantum physics is "The Quantum Universe: Everything that Can Happen Does Happen" by Brian Cox and Jeff Forshaw. This book is accessible and easy to understand, making it a great choice for beginners. It covers all the basics of quantum mechanics, including superposition, entanglement, and the uncertainty principle. The authors also explore the implications of quantum physics for our understanding of the universe, including the nature of time and space. Overall, "The Quantum Universe" is an excellent choice for anyone looking to gain a foundational understanding of this fascinating field
llama-2-13b-chat
Certainly! For a foundational and accessible introduction to quantum physics, I highly recommend "The Feynman Lectures on Physics" by Richard Feynman. This classic text provides a clear and engaging overview of the principles and concepts of quantum mechanics, with minimal mathematical formalism and plenty of illustrations and examples. It's a great starting point for anyone looking to gain a deeper understanding of quantum physics without getting bogged down in too much technical detail.

图 6：Mistral 7B – Instruct 与 Llama 2 13B – Chat 的人工评测示例。来自 llmboxing.com 的一个人工评测示例。问题要求推荐量子物理方面的书籍。Llama 2 13B – Chat 推荐了一本普通物理书籍，而 Mistral 7B – Instruct 推荐了一本与量子物理更相关的书，并更详细地介绍了其内容。

References

[1] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

[2] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[4] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[7] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

[8] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

[9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[11] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

[12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[13] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[15] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[16] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

References

[1] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

[2] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[4] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[7] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

[8] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

[9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[11] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

[12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[13] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[15] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[16] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[17] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[18] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022.

[19] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[20] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[21] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[22] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[23] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[24] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[25] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[29] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

[17] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[18] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022.

[19] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[20] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[21] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[22] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[23] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[24] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[25] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[29] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.