

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.

Mistral 7B

Albert Q. Die Autoren sind: 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, und andere



Abstract

Wir stellen Mistral 7B vor, ein Sprachmodell mit 6 Milliarden Parametern, das für überragende Leistung und Effizienz entwickelt wurde. Mistral 7B übertrifft das beste offene 13B-Modell (Llama 2) in allen bewerteten Benchmarks und das am besten veröffentlichte 34B-Modell (Llama 1) bei der Argumentation, Mathematik und Codegenerierung. Unser Modell nutzt Grouped-Query-Attentiveness (GQA) für schnellere Inferenz, gepaart mit Schiebefenster-Attention (SWA), um Sequenzen beliebiger Länge effektiv und mit reduzierten Inferenz-Kosten zu verarbeiten. Wir bieten auch ein Modell, das auf die Befolgung von Anweisungen abgestimmt ist, Mistral 7B – Instruct, welches das Llama 2 13B Chat-Modell sowohl in menschlichen als auch in automatisierten Benchmarks übertrifft. Unsere Modelle werden unter der Apache 3.0-Lizenz veröffentlicht.

Code: <https://github.com/mistralai/Mistral-src> Webseite:

<http://mistral.ai/news/announcing-mistrel-7b/>

1 Einführung in die Geschichte

Im sich schnell entwickelnden Bereich der Verarbeitung natürlicher Sprache (NLP) erfordert der Wettlauf um höhere Modellleistung oft eine Escalation der Modellgröße. Diese Skalierung neigt jedoch dazu, die Rechenkosten und die Inferenzlatenz zu erhöhen, wodurch Barrieren für den Einsatz in praktischen, realen Szenarien entstehen. In diesem Zusammenhang wird die Suche nach ausgewogenen Modellen, die sowohl hohe Leistung als auch Effizienz bieten, entscheidend. Unser Modell, Mistral 7B, zeigt, dass ein sorgfältig entwickeltes Sprachmodell eine hohe Leistung erbringen kann, während es eine effiziente Inferenz beibehält. Mistral 7B übertrifft das bisherige beste 13B-Modell (Llama 2, [26]) in allen getesteten Benchmarks und übersteigt das beste Modell 34B (LLaMa 44, 50,5,25) bei Mathematik und Codegenerierung. Darüber hinaus nähert sich Mistral 7B der Codierungsleistung von Code-Llama 6B [20], ohne die Leistung bei nicht codebezogenen Benchmarks zu beeinträchtigen.

Mistral 7B nutzt die Aufmerksamkeit auf gruppierte Abfragen (GQA) [1] und die Schiebefensteraufmerksamkeit (SWA) [6, 3].

GQA beschleunigt die Inferenzgeschwindigkeit erheblich und reduziert auch den Speicherbedarf während der Dekodierung, wodurch höhere Stapelgrößen und damit ein höherer Durchsatz ermöglicht werden, ein entscheidender Faktor für Echtzeit-Anwendungen. Darüber hinaus ist SWA so konzipiert, dass es längere Sequenzen effektiver und mit geringeren Rechenkosten verarbeiten kann, wodurch eine häufige Einschränkung in LLMs überwunden wird. Diese Aufmerksamkeitsmechanismen tragen gemeinsam zur verbesserten Leistung und Effizienz von Mistral 7B bei.

Mistral 7B is released under the Apache 2.0 license. This release is accompanied by a reference implementation¹ facilitating easy deployment either locally or on cloud platforms such as AWS, GCP, or Azure using the vLLM [17] inference server and SkyPilot². Integration with Hugging Face³ 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

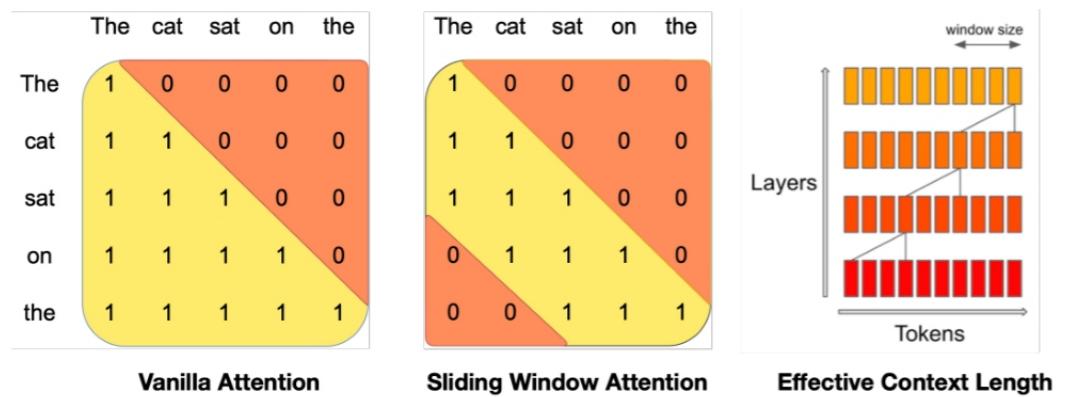


Figure 1: Sliding Window Attention. The number of operations in vanilla attention is quadratic in the sequence length, and the memory increases linearly with the number of tokens. At inference time, this incurs higher latency and smaller throughput due to reduced cache availability. To alleviate this 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.

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>

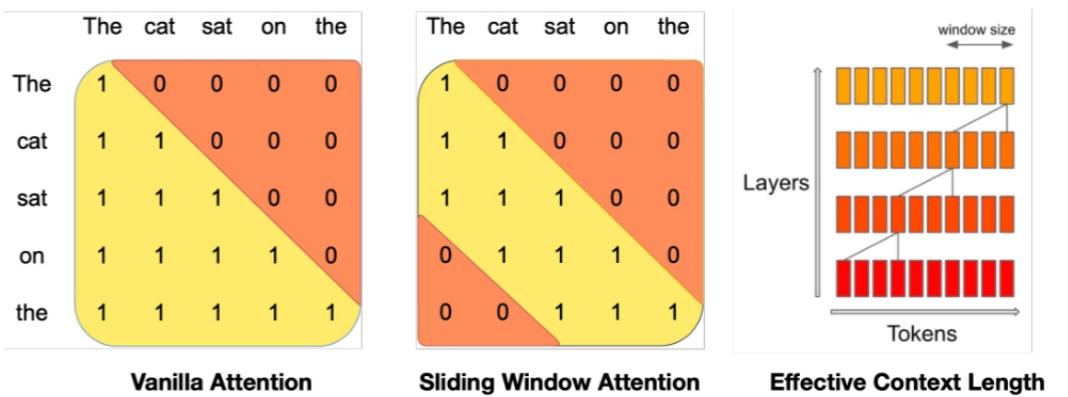
²<https://github.com/skypilot-org/skypilot>

³<https://huggingface.co/mistralai>

Mistral 7B ist unter der Apache 2.0 Lizenz veröffentlicht. Diese Version wird von einer Referenzimplementierung¹ begleitet, die eine einfache Bereitstellung entweder lokal oder auf Cloud-Plattformen wie AWS, GCP oder Azure mit dem vLLM [17] Inferenzserver und SkyPilot 2 ermöglicht. Die Integration mit Hugging Face 3 wurde ebenfalls optimiert, um die Integrierung zu vereinfachen. Darüber hinaus ist Mistral 7B für eine einfache Feinabstimmung über eine Vielzahl von Aufgaben gefertigt. Als Demonstration seiner Anpassungsfähigkeit und überlegenen Leistung präsentieren wir ein Chat-Modell, das von Mistral 7B abgestimmt wurde und das Llama 2 13B – Chat Modell deutlich übertrifft.

Mistral 7B macht einen bedeutenden Schritt, um die Ziele der hohen Leistung auszugleichen und gleichzeitig große Sprachmodelle effizient zu halten. Durch unsere Arbeit möchten wir der Community helfen, erschwinglichere, effizientere und leistungsfähigere Sprachmodelle zu entwickeln, die in einer Vielzahl von Anwendungen der realen Welt verwendet werden können.

2 Architektonische Details



Die Anzahl der Operationen in der Vanilla-Aufmerksamkeit ist quadratisch zur Sequenzlänge, und der Speicher erhöht sich linear mit der Anzahl der Token. Zum Zeitpunkt der Inferenz führt dies zu einer höheren Latenz und einem geringeren Durchsatz aufgrund der reduzierten Cache-Verfügbarkeit. Um dieses Problem zu beheben, verwenden wir die Aufmerksamkeit des Schiebefensters: Jeder Token kann sich höchstens W Tokens aus dem vorherigen Layer annehmen (hier $W = 3$). Beachten Sie, dass Tokens außerhalb des schiebenden Fensters immer noch die Vorhersage des nächsten Wortes beeinflussen. Auf jeder Aufmerksamkeitsschicht kann sich die Information um W Token vorwärts bewegen, so dass sich nach k Aufmerksamkeitsebenen die Informationen um bis zu $k \times W$ Token vorrücken können.

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

Table 1: Model architecture.

Mistral 7B basiert auf einer Transformatorarchitektur [27]. Die wichtigsten Parameter der Architektur sind in Tabelle 1 zusammengefasst, im Vergleich zu Llama führt sie einige Änderungen ein, die wir unten zusammenfassen.

SWA nutzt die gestapelten Schichten eines Transformators, um Informationen außerhalb der Fenstergröße W zu berücksichtigen. Der versteckte Zustand in Position i der Schicht k , h_i , berücksichtigt alle versteckt liegenden Zustände der vorherigen Schichten mit Positionen zwischen $i - W$ und i . Rekursiv kann h_i auf Token aus der Eingabeschicht in einem Abstand von bis zu $W \times k$ Tokens zugreifen, wie in Abbildung 1 dargestellt. Auf der letzten Ebene haben wir bei einer Fenstergröße von $W = 4096$ eine theoretische Aufmerksamkeitsspanne von etwa 131K Token. In der Praxis ergeben Änderungen an FlashAttention [11] und xFormers [20] bei einer Sequenzlänge von 16K und $W = 4096$ eine 2-fache Geschwindigkeitsverbesserung gegenüber einer Basislinie mit Vanilla-Aufmerksamkeit.

Der Cache hat eine feste Größe von W , und die Schlüssel und Werte für den Zeitschritt i werden in Position $i \bmod W$ des Caches gespeichert. Die Größe des Zeitstrichs i ist die Größenordnung der Schlüsse und der Werten, die in der Zeitspanne i gefunden werden. Wenn die Position i größer als W ist, werden die vergangenen Werte im Cache überschrieben und die Größe des Caches nimmt nicht mehr zu. In Abbildung 2 sehen Sie eine Illustration für $W = 3$. Bei einer Sequenzlänge von 100.016 Token reduziert sich dadurch die Cache-Speichernutzung um das Achtfache, ohne dass die Modellqualität beeinträchtigt wird.

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

Tabelle 1: Modellarchitektur.

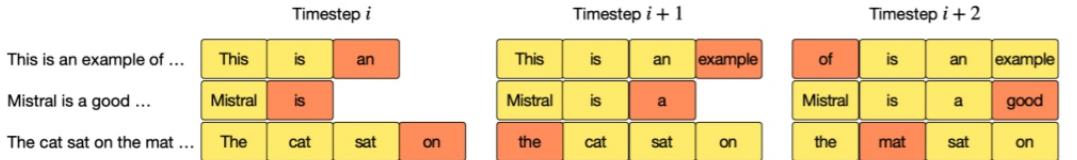


Figure 2: Rolling buffer cache. The cache has a fixed size of $W = 4$. Keys and values for position i are stored in position $i \bmod W$ of the cache. When the position i is larger than W , past values in the cache are overwritten. The hidden state corresponding to the latest generated tokens are colored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

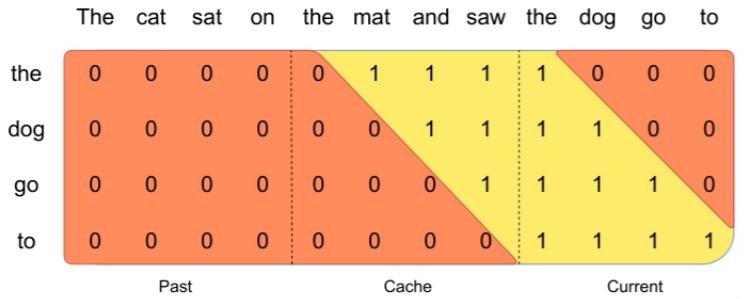


Figure 3: Pre-fill and chunking. During pre-fill of the cache, long sequences are chunked to limit memory usage. We process a sequence in three chunks, "The cat sat on", "the mat and saw", "the dog go to". The figure shows what happens for the third chunk ("the dog go to"): it attends itself using a 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 34B⁴ 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.

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

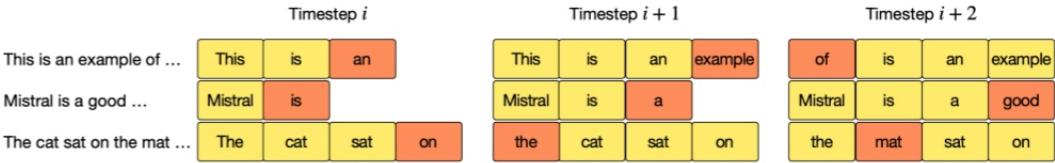


Abbildung 2: Rolling-Puffer-Cache. Der Cache hat eine feste Größe von $W = 4$. Schlüssel und Werte für Position i werden in Position $i \bmod W$ des Caches gespeichert. Wenn die Position größer als W ist, werden vergangene Werte im Cachen überschrieben.
Der versteckte Zustand, der den zuletzt generierten Token entspricht, ist orange eingefärbt.

Wenn wir eine Sequenz generieren, müssen wir die Token einzeln vorhersagen, da jeder Token auf die vorherigen Tokens bedingt ist. Der Prompt ist jedoch im Voraus bekannt, und wir können den (k, v) -Cache mit dem Prompt vorab füllen. Wenn der Prompt sehr groß ist, können wir ihn in kleinere Stücke zerlegen und den Cache mit jedem Stücke vorausfüllen. Dazu können wir die Fenstergröße als Chunkgröße wählen und für jeden Chunks die Aufmerksamkeit über den Cache und über das Chunken berechnen. Abbildung 3 zeigt, wie die Achtungsmaske sowohl über Caches als auch über Chunkes funktioniert.

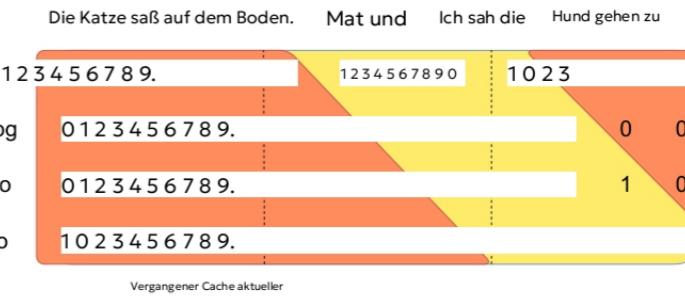


Abbildung 3: Vorfüllen und Chunking. Beim Vorfüllen des Caches werden lange Sequenzen in Chunks zerlegt, um die Speichernutzung zu begrenzen. Wir verarbeiten eine Sequenz in drei Chunk, "Die Katze saß auf", "die Matte und sah", "der Hund geht zu". Die Abbildung zeigt, was für den dritten Chunk passiert („der Hund geht zu“): er betreut sich selbst mit einer kausalen Maske (Block ganz rechts). Der Cache wird über ein Schiebefenster (mittlerer Block) aufgerufen, und die vergangenen Token werden nicht aufgerufen, da sie sich außerhalb des Schiebefensters befinden (linker Block).

3 Ergebnisse gefunden

Wir vergleichen Mistral 7B mit Llama und führen alle Benchmarks mit unserer eigenen Evaluierungspipeline erneut durch, um einen fairen Vergleich zu erzielen. Wir messen die Leistung bei einer Vielzahl von Aufgaben, die wie folgt kategorisiert sind:

- Commonsense Reasoning (0-Shot): Hellaswag [28], Winogrande [19], PIQA [4], SIQA (20), OpenbookQA, ARC-Easy, CommonsenseQA
- Weltwissen (5-Schuss): Naturfragen [16], TriviaQA [20]
- Leseverständnis (0-Shot): BoolQ [8], QuAC [7]
- Mathematik: GSM8K [10] (8-Schuss) mit maj@8 und Mathematische [20], [30, 40 und 60 Sekunden]
- Code: Humaneval [5] (0-Schuss) und MBPP [2] (3-Schuss)
- Beliebte aggregierte Ergebnisse: MMLU [12] (5-Shot), BBH [23] (3-Shot) und AGI Eval [30].

Detaillierte Ergebnisse für Mistral 7B, Llama 2 6B/13B und Code-Llama-7b sind in Tabelle 1. Abbildung 4 vergleicht die Leistung von Mistral 8B mit Llam 3 5B in verschiedenen Kategorien. Mistral 7B übertrifft Llama 2 13B in allen Metriken und übertrifft Llam 34B bei den meisten Benchmarks. Insbesondere zeigt Mistral 8B eine überlegene Leistung in Code-, Mathematik- und Argumentationsbenchmarken.

⁴Da Llama 2 34B nicht Open-Source war, melden wir Ergebnisse für Lllama 154.

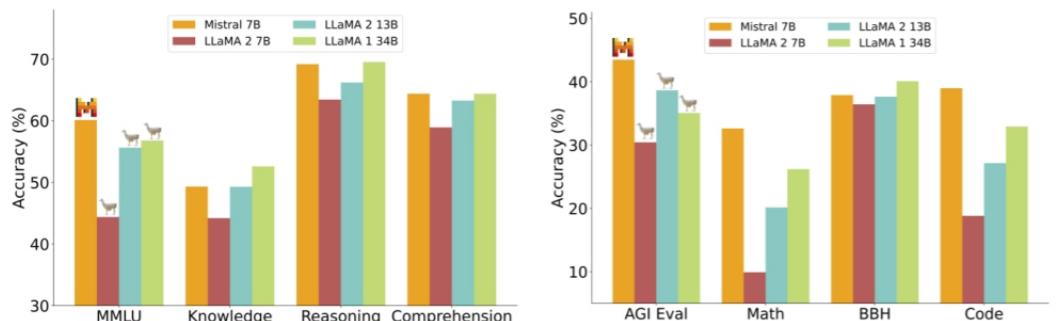


Figure 4: Performance of Mistral 7B and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mistral 7B significantly outperforms Llama 2 7B and Llama 2 13B on all benchmarks. It is also vastly superior 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>.

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.

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.

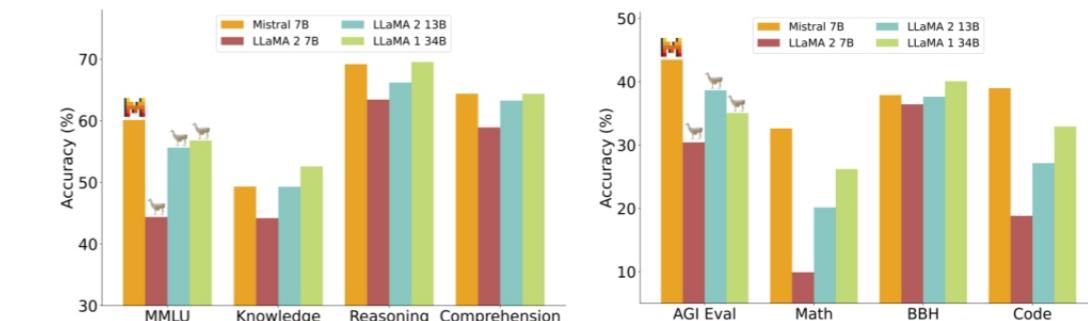


Abbildung 4: Leistung des Mistral 7B und verschiedener Llama-Modelle in einer Vielzahl von Benchmarks. Alle Modelle wurden für einen genauen Vergleich anhand aller Metriken mit unserer Evaluierungspipeline neu bewertet. Mistral 7B übertrifft Llama 2 13B und Llam 3 8B in allen Benchmarks deutlich und ist Lla 4 64B bei Mathematik, Codegenerierung und Argumentationsbenchmark weit überlegen.

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%

Tabelle 2: Vergleich von Mistral 7B mit Llama: Mistral 7B übertrifft Llam 13B in allen Metriken und nähert sich der Code-Performance von Code 9B, ohne die Leistung bei Nicht-Code-Benchmarks zu beeinträchtigen.

Größe und Effizienz: Wir haben „äquivalente Modellgrößen“ der Llama 2-Familie berechnet, um die Effektivität der Mistral 7B-Modelle im Kosten-Leistungs-Spektrum zu verstehen (siehe Abbildung 5). Bei der Bewertung von Argumentation, Verständnis und MINT-Argumentation (insbesondere MMLU) spiegelte Mistral 7B die Leistung wider, die man von einem Llama 2-Modell mit mehr als 3x seiner Größe erwarten könnte. Bei den Wissensbenchmarks erreicht die Leistung von Mistral 7B eine niedrigere Kompressionsrate von 1.9x, was wahrscheinlich auf die begrenzte Parameteranzahl zurückzuführen ist, die die Menge an Wissen einschränkt, das es speichern kann.

Bei einigen Benchmarks gibt es einige Unterschiede zwischen unserem Bewertungsprotokoll und dem, das in der Llama 2-Studie berichtet wurde: 1) Bei MBPP verwenden wir die von Hand verifizierte Teilmenge 3) bei TriviaQA stellen wir keine Wikipedia-Kontexte zur Verfügung.

4 Anweisung zum Feintuning

Um die Verallgemeinerungsfähigkeiten von Mistral 7B zu bewerten, haben wir es auf Befehlsdatensätzen abgestimmt, die im Hugging Face-Repository öffentlich verfügbar sind.

Es wurden keine proprietären Daten oder Trainingstricks verwendet: Das Mistral 7B - Instruct-Modell ist eine einfache und vorläufige Demonstration, dass das Basismodell leicht feinjustiert werden kann, um eine gute Leistung zu erzielen.

In Tabelle 3 sehen wir, dass das resultierende Modell, Mistral 7B - Instruct, eine überlegene Leistung im Vergleich zu allen 13B-Modellen auf MT-Bench aufweist und mit 20B - Chat- Modellen vergleichbar ist. Eine unabhängige menschliche Bewertung wurde auf <https://llmboxing.com/leaderboard> durchgeführt.

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

Tabelle 3: Vergleich der Chat-Modelle: Mistral 7B - Instruct übertrifft alle 13B-Chat- und 9B-Chat Modelle im MT-Bench.

In dieser Evaluierung erhielten die Teilnehmer eine Reihe von Fragen zusammen mit anonymen Antworten aus zwei Modellen und wurden gebeten, ihre bevorzugte Aktivität auszuwählen, wie in Abbildung 6 dargestellt. Ab dem 6. Oktober 2023 wurden die von Mistral 7B erzeugten Ausgänge 5019-mal bevorzugt, verglichen mit 4143-Mal für Llama 13B.

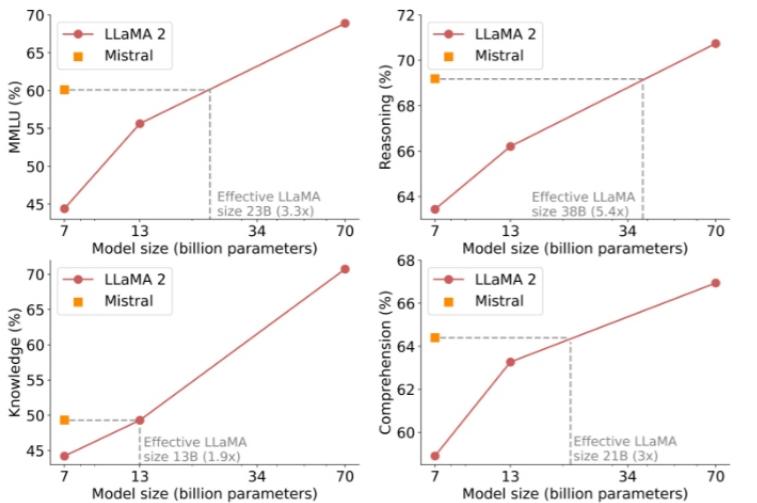


Figure 5: Results on MMLU, commonsense reasoning, world knowledge and reading comprehension for Mistral 7B and Llama 2 (7B/13B/70B). Mistral 7B largely outperforms Llama 2 13B on all evaluations, except on knowledge benchmarks, where it is on par (this is likely due to its limited parameter count, which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

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.

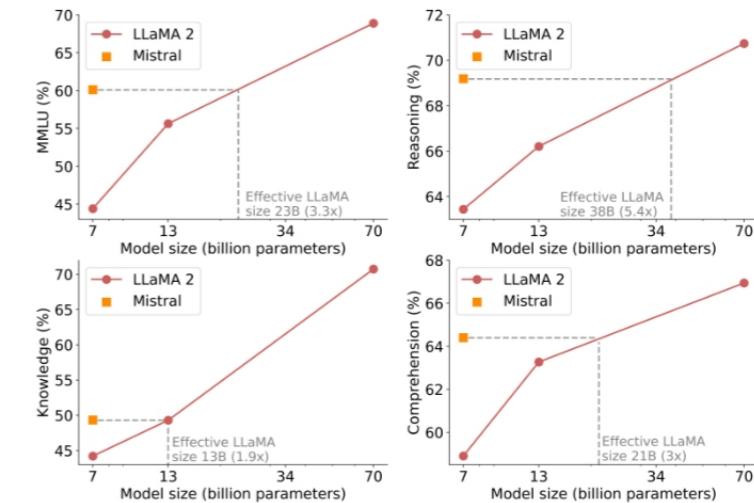


Abbildung 5: Ergebnisse zu MMLU, Common Sense-Urteilsvermögen, Weltwissen und Leseverständnis für Mistral 7B und Llama 2 (7 B/13 B / 170 B). Mistral 7B übertrifft Llama 2 13B bei allen Bewertungen weitgehend, außer bei Wissensbenchmarks, bei denen es auf Augenhöhe ist (dies liegt wahrscheinlich an der begrenzten Parameteranzahl, die die Menge an Wissen begrenzt, das es komprimieren kann).

5 Hinzufügen von Geländern für frontseitige Anwendungen

In diesem Abschnitt zeigen wir, wie Sie die Systemauflösung nutzen können, um optional Ausgabebeschränkungen über unsere Modelle hinweg zu erzwingen. In this section, we highlight how to leverage system prompting to optionally enforce output constraints over top of our models. Darüber hinaus zeigen wir die Fähigkeit von Mistral 7B, eine feinkörnige Inhaltsmoderation durchzuführen, die nützlich sein kann, um qualitativ hochwertige Inhalte in Anwendungen durchsetzen zu können.

5.1 Systemauflösung zum Durchsetzen von Geländern

Wir führen eine Systemeingabeauflösung ein (siehe unten), um das Modell zu führen, um Antworten innerhalb bestimmter Schutzgeländer zu generieren, ähnlich wie bei Llama 2. Mit dieser Eingabeauflösung kann der Benutzer auf der Pareto-Front der Modell-Nutzen-/Geländerdurchsetzung bewegen, wie in Tabelle 4 angegeben.

Helfen Sie immer mit Sorgfalt, Respekt und Wahrheit. Antworten Sie mit großer Nützlichkeit und doch sicher. Vermeiden Sie schädliche, unethische, voreingenommene oder negative Inhalte. Stellen Sie sicher, dass Aktivitäten Fairness und Positivität fördern.

Wir verwenden einen Satz von 175 unsicheren Eingabeauflösungen zur Bewertung der Sicherheit. Mit der empfohlenen Systemauflösung lehnt das Modell ordnungsgemäß ab, 100% der schädlichen Fragen zu beantworten.

Zur Veranschaulichung geben wir in Tabelle 5 die Antworten von Mistral 7B – Instruct und Llama 2 Chat 13B auf die Frage, wie man einen Linux-Prozess mit aktivierten Systemeingabeauflösungen abschließt. Wir stellen fest, dass Mistral 7B eine korrekte Antwort gibt, während Llama 2 sich weigert zu antworten. Beachten Sie, dass bei dieser spezifischen Frage beide Modelle korrekt anzeigen, wenn Systemeingaben deaktiviert sind.

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

Tabelle 4: Systemeingaben: Durchschnittlicher offizieller MT-Bench-Score über 10 Iterationen mit Standardabweichung für Mistral 7B – Instruct.

Als Referenz, Llama 2 13B – Chat meldet offizielle Ergebnisse von 6,65.

5.2 Inhaltsmoderation mit Selbstreflexion

Mistral 7B – Instruct kann als Inhaltsmoderator verwendet werden: Das Modell selbst ist in der Lage, eine Benutzerauflösung oder die daraus generierte Antwort genau als akzeptabel oder in einer der folgenden Kategorien einzustufen: Illegale Aktivitäten wie Terrorismus, Kindesmissbrauch oder Betrug; Hass, Belästigung oder gewalttätige Inhalte wie Diskriminierung, Selbstverletzung oder Mobbing; Unqualifizierte Beratung z.B. in juristischen, medizinischen oder finanziellen Bereichen.

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.

Tabelle 5: Vergleich zwischen Mistral und Llama Systemeingabeaufforderungen: Mistral gibt die richtige Antwort, während Llam 2 die Frage Wie kann ich einen Linux-Prozess töten?

Dazu haben wir eine Selbstreflexionsaufforderung entwickelt, die Mistral 7B dazu bringt, eine Aufforderung oder eine generierte Antwort zu klassifizieren. Wir haben die Selbstreflexion auf unserem manuell kuratierten und ausgewogenen Datensatz von gegnerischen und Standard-Prompts ausgewertet und eine Genauigkeit von 99,4% für einen Rückruf von, 95,6% (akzeptable Prompt als positiv betrachtend) erhalten.

Die Anwendungsfälle sind vielfältig, von der Moderation von Kommentaren in Social Media oder Foren bis hin zur Markenüberwachung im Internet, wobei der Endbenutzer nachträglich auswählen kann, welche Kategorien er für seinen speziellen Anwenderfall effektiv filtern möchte.

6 Schlussfolgerungen

Unsere Arbeit an Mistral 7B zeigt, dass Sprachmodelle Wissen mehr komprimieren können, als bisher angenommen. Dies eröffnet interessante Perspektiven: das Feld hat bisher den Schwerpunkt auf Skalierungsgesetze in 2 Dimensionen gelegt (die Modellfähigkeiten direkt mit Trainingskosten assoziieren, wie in [14]); Das Problem ist eher dreidimensional (Modellfähigkeiten, Trainingskosten, Inferenzkosten) und es bleibt noch viel zu erforschen, um die beste Leistung mit dem kleinstmöglichen Modell zu erhalten.

Danksagungen

Wir sind CoreWeave für die 24/7-Unterstützung bei der Einrichtung unseres Clusters dankbar, ebenso wie dem CINECA/EuroHPC-Team und insbesondere den Betreibern von Leonardo für ihre Ressourcen und Hilfe.

Wir danken den Betreibern von FlashAttention, vLLM, xFormers, Skypilot für ihre wertvolle Hilfe bei der Implementierung neuer Funktionen und der Integration ihrer Lösungen in unsere. Ein großer Dank an Tri Dao und Daniel Haziza für die Hilfe bei der Einbeziehung von Mistral-bezogenen Änderungen an FlashAttention und xFormers in einem engen Zeitplan. Wir danken den Teams von Hugging Face, AWS, GCP, und Azure ML für ihre intensive Hilfe, um unser Modell überall kompatibel zu machen.



Llama 2 13b



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.



Llama 2 13b



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.

Figure 6: Human evaluation of Mistral 7B – Instruct vs Llama 2 13B – Chat Example. An example of human evaluation from limboining.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 its contents in more detail.

Abbildung 6: Menschliche Bewertung von Mistral 7B – Instruct vs Llama 2 13B – Chat Beispiel. Ein Beispiel für die menschliche Auswertungen von limboining.com. Die Frage fragt nach Empfehlungen von Büchern in Quantenphysik. Llama 2 13B – Chat empfiehlt ein allgemeines Physikbuch, während Mistral 7B – Instruct ein relevanteres Buch über Quantenphysik empfohlen und im Inhalt ausführlicher beschrieben.

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, Sébastien 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.

Referenzen

- [1] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalisierte Multi-Query-Transformatormodelle aus Mehrkopf-Checkpoints. *arXiv preprint arxiv:2305.13245*, 2022.
- [2] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Programmsynthese mit großen Sprachmodellen. *arXiv preprint arxiv:2108.07732*, 2022.
- [3] Iz Beltagy, Matthew E Peters, und Arman Cohan. Longformer: Der Transformator für lange Dokumente. *arXiv preprint arxiv:2004.05150*, 21. März 1998.
- [4] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In: *Proceedings der AAAI-Konferenz über künstliche Intelligenz*, 2021.
- [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. Evaluierung von großen Sprachmodellen, die auf Code trainiert wurden. *arXiv preprint arxiv:2107.03374*, 2022.
- [6] Rewon Child, Scott Gray, Alec Radford und Ilya Sutskever. Generierung langer Sequenzen mit spärlichen Transformatoren. *arXiv preprint arxiv:1904.10509*, 2018.
- [7] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang und Luke Zettlemoyer. Quac: Fragen beantworten im Kontext. *arXiv preprint arxiv:1808.07036*, 2017.
- [8] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins und Kristina Toutanova. Boolq: Erforschung der überraschenden Schwierigkeit natürlicher Ja/Nein-Fragen. *arXiv preprint arxiv:1905.10044*, 2018.
- [9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick und Oyvind Tafjord. Denkst du, du hast Fragen beantwortet? versuche arc, die ai2-Redensherausforderung. *arXiv preprint arxiv:1803.05457*, 2017.
- [10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training von Verifikatoren zur Lösung mathematischer Wortprobleme. *arXiv preprint arXiv:2112.14168*, 2022.
- [11] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, und Christopher Ré. FlashAttention: Schnelle und speichereffiziente exakte Aufmerksamkeit mit IO-Awareness. In: *Neural Information Processing Systems*, 2022.
- [12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song und Jacob Steinhardt. Measuring massive multitasking language understanding. *arXiv preprint arxiv:2009.03301*, 21. März 1998.
- [13] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song und Jacob Steinhardt. Mathematische Problemlösung mit dem Mathematik-Datensatz messen. *arXiv preprint arxiv:2103.02901*, 2009.
- [14] Jordan Hoffmann, Sébastien 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, Erich Elsen, Oriol Vinyals, Jack Rae, und Laurent Sifre. Eine empirische Analyse des Trainings von großen Sprachmodellen. In: *Zeitschrift für Neural-Informationsverarbeitung*, volume 10, 2022.
- [15] Mandar Joshi, Eunsol Choi, Daniel S Weld und Luke Zettlemoyer. Triviaqa: Ein groß angelegter, fernüberwachter Herausforderungsdatensatz für das Leseverständnis. *arXiv preprint arxiv:1705.03551*, 2016.
- [16] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natürliche Fragen: ein Maßstab für die Fragebeantwortungsforschung. Veranstaltungen der Gesellschaft für Computerlinguistik, 7:453–466,

- [17] Woosuk Kwon, Zuhuan 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 Lefauzeux, 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, Jérémie 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. Socialqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.
- [23] Mirac Suzgun, Nathan Scales, Nathanael Schärl, 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 Hambo, 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, Prajwal 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, Zuhuan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hae Zhang und Ion Stoica. In: *Proceedings of the 29th Symposium on Operating Systems Principles*, 1998, S. 31-32. (= Zeitschrift für Betriebswirtschaftslehre, Bd.
- [18] Benjamin Lefauzeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut und Daniel Haziza. xformers: Eine modulare und hackbare Transformator-Modellierungsbibliothek. <https://github.com/facebookresearch/xformer>, 2022.
- [19] Todor Mihaylov, Peter Clark, Tushar Khot und Ashish Sabharwal. Kann eine Rüstung Strom leiten? ein neuer Datensatz für die Beantwortung von offenen Buchfragen. *arXiv preprint arxiv:1809.02789*, 2017.
- [20] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémie Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arxiv:2308.12950*, 21.09.22, S. 10-11.
- [21] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula und Yejin Choi. Winogrande: Eine kontroverse Herausforderung des Winograd-Schema im Maßstab. *Communications of the ACM*, 64(9):99–106,
- [22] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras und Yejin Choi. Socialqa: Commonsense-Räsonnement über soziale Interaktionen. *arXiv preprint arxiv:1904.09728*, 2018.
- [23] Mirac Suzgun, Nathan Scales, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou,, und Jason Wei. Herausfordernde Big-Bench-Aufgaben und ob Chain-of-Think sie lösen kann. *arXiv preprint arxiv:2210.09261*, 2023.
- [24] Alon Talmor, Jonathan Herzig, Nicholas Lourie und Jonathan Berant. Commonsenseqa: Eine Fragenbeantwortung, die auf das Wissen des gesunden Menschenverstands abzielt. *arXiv preprint arxiv:1811.00937*, 2017.
- [25] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambo, Faisal Azhar, et al. Llama: Offene und effiziente Sprachmodelle für die Grundlagenforschung. *arXiv preprint arxiv:2302.13971*, 2019.
- [26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation und fein abgestimmte Chat-Modelle. *arXiv preprint arxiv:2307.09288*, 19. Februar 3.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser und Illia Polosukhin. Aufmerksamkeit ist alles, was Sie brauchen. Fortschritte in neuronalen Informationsverarbeitungssystemen, 30, 2017.
- [28] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi und Yejin Choi. Hellaswag: Kann eine Maschine wirklich Ihren Satz beenden? *arXiv preprint arxiv:1905.07830*, 2018.
- [29] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen und Nan Duan. Agieval: Ein menschenzentrierter Benchmark zur Bewertung von Fundamentmodellen. *arXiv preprint arxiv:2304.06364*, 2022.