scientific reports

Check for updates

OPEN

APBench and benchmarking large language model performance in fundamental astrodynamics problems for space engineering

Di Wu✉, Raymond Zhang, Enrico M. Zucchelli, Yongchao Chen & Richard Linares

The problem-solving abilities of Large Language Models (LLMs) have become a major focus of research across various STEM fields, including mathematics and physics. Substantial progress has been made in both measuring and enhancing these abilities. Among the multitude of ways to advance space engineering research, one promising direction is the application of LLMs and foundational models in aiding in solving Ph.D. level research problems. To understand the full potential of LLMs in astrodynamics, we have developed the first Astrodynamics Problems Benchmark (APBench) to evaluate the capabilities of LLMs in this field. We crafted for the first time a collection of questions and answers that range a wide variety of subfields in astrodynamics, astronautics, and space engineering. On top of this first dataset built for space engineering, we evaluate the performance of foundational models, both open source models and closed ones, to validate their current capabilities in space engineering, and paving the road for further advancements towards a definition of intelligence for space.

Large language models (LLMs) have demonstrated the ability to perform complex tasks, such as recognizing intricate patterns, driving cars, operating humanoid robots, and even piloting spacecraft[1–6]. Powered by vast amounts of data - often terabytes collected from the internet - LLMs possess extensive general knowledge about the world. One notable advancement of LLMs over traditional artificial intelligence (AI) methods is their ability to perform zero-shot and few-shot learning, where they can complete tasks with little to no examples provided. With well designed prompting and fine-tuning, there have been a promising evaluation of the ability of the models to synthesize code from complex text[7]. Because of this, there has been an increasing interest in applying LLMs for space engineering problems. Rodriguez et al.[8] explored using LLM as an agent, with carefully crafted zero-shot prompting, directly controlling a spacecraft in a simulated environment for computationally intensive control policy determination; Zucchelli et al.[9] explored multiple space-oriented tasks for fine-tuned LLM and their findings indicate that LLMs are more sample efficient compared to traditional AI methods. Unlike traditional supervised or reinforcement learning, which often fails to handle rare events due to lack of exposure, LLMs are capable of handling such scenarios effectively. The interest in crafting foundational model application in space engineering has also been joined by Foutter et al.[10]. With fine-tuning techniques, a (state-of-the-art) SoTA vision-language foundational model has been adapted to plan a rover's navigation on Mars' surface.

Recent studies have demonstrated that LLMs can also exhibit a basic understanding of the physical world. LLMs like GPT-4 are celebrated not just for their sophisticated logic but also for their expansive knowledge base[11]. Using Chain-of-Thought (CoT) prompting—a method that guides the model step-by-step through reasoning processes—LLMs can perform a range of arithmetic and algebraic reasoning tasks by breaking them down into smaller, logical steps. With effective prompt engineering, LLMs have shown remarkable versatility. However, LLMs are still far away from an absolute satisfactory skill for solving mathematical problems due to the complex reasoning procedures that would be required[12].

Fedorenko et al.[13] have been questioning the primary use of LLMs as communication or thinking tools. On the quest of the big problem asked "Does LLM have intelligence", we are testing its thinking capabilities. Astrodynamics problems can involve high complexity, making this field a perfect area, but void domain, to be explored in terms of determining LLM's basic logical reasoning and problem solving skills. Astrodynamics problems cover domain-specific knowledge, causal explanation, advanced theories and operations, and spatio-temporal intelligence. LLMs have been implemented in physics and mathematics, with success coming from

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. ✉email: d_wu@mit.edu

the logical reasoning and basic arithmetic operations. Arguments have been raised on the essence of LLMs, but recent work from Zhu et al.[14] indicates that LLMs could "go beyond" the learning tasks and knowledge operation, suggesting a potential for a foundational model for space engineering. However, the question of defining intelligence for space foundational models has rarely been raised. Benchmarking has been the way to quantify the performance of LLMs in different fields such as mathematics[15,16], including problems that cannot be solved with a straightforward application of standard K-12 mathematics tools[17], astronomy[18], and finance[19].

In order to test the capabilities in an astrodynamics setting, we develop a specialized dataset and a detailed evaluation process for quantifying LLMs performance within this domain to establish a baseline and support future development, with the goal of bringing the astrodynamics community and AI community together in defining space foundational models. Figure 1 shows a basic framework of our astrodynamics dataset evaluation process.

Our APBench (Astrodynamics Problems Benchmark) dataset covers a wide variety of problems in astrodynamics, including 2-body and 3-body problems, concepts on general space terminologies, state variables conversions, orbit determination, maneuver optimization, rendezvous and docking, interplanetary mission planning, computation and analysis of perturbations, etc. We challenge a wide variety of LLMs (see Table 3), pinpointing individual strengths and weaknesses. To better suit to astrodynamics problems, we adapt an existing structure for the database that is efficient and informative for use with the general structure of Content and Questions as shown in Fig. 2. We source the benchmark questions to proven example problems rather than creating new questions[20] as those were already thought through with solutions and we can push to the number of questions so that the evaluation covers a wide range of questions. With our automatic evaluating pipeline handling open questions in both numeric and message styles, we primarily focus on the final answers given by LLMs as this is the most straightforward way of evaluating the performance on the open questions curated in the dataset. Our result is the first trial in space engineering to generate a universal baseline for both the definition and pursuit of a space foundational model.

The goal of generating the APBench is to identify LLMs that best employ logic and knowledge to solve astrodynamics problems in a clear and concise step-by-step process. Our work explores the performance of existing LLMs in the field of space engineering, with the goal of examining one aspect necessary for building practical space foundational model. The aspect is to provide consistently correct performance to fundamental space engineering questions. We predict a possible timeline of 2025 for this aspect to be satisfied by LLM's empirical performance improvement. Beyond the scope of astrodynamics, our data curation process and evaluation workflow aim at promoting the ongoing trend of customized benchmarking, providing a useful framework for evaluating artificial general intelligence (AGI) performance for other fields.

Results

The APBench dataset

To initiate this study, we focus on representative tasks of astrodynamics, leading to the creation of the first Astrodynamics Problems Benchmark (APBench). Similar to other STEM fields, textbook examples are a core component of engineering training used to qualify specialists in space engineering. Accordingly, we collected examples from available open resources to construct this dataset of college-level problem sets in astrodynamics.

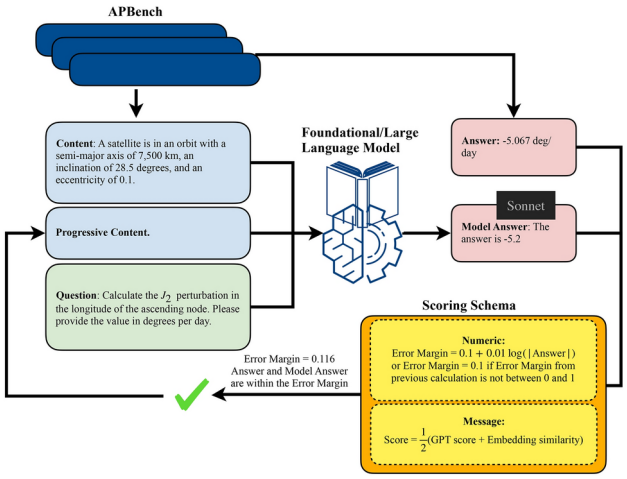

Fig. 1. APBench evaluation flow. A single evaluation starts with attaching Content information, Progressive Content (generated from previous evaluation), and Question into the general prompt to a Large Language Model. In this illustrated case, we use Claude 3.5 Sonnet. The true Answer is use in the Scoring Schema to determine a customized margin for a Numeric answer. As the Model Answer − 5.2 is between − 4.479 and − 5.655, upper and lower margin according to the Error Margin and Answer, the Model Answer is correct. Then, according to the definition of Progressive Content, this Question and Answer are included in the Progressive Content.

Content: Calculate the velocity change required to transfer a satellite from a circular 600 km orbit with an inclination of 28 degrees to an orbit of equal size with an inclination of 20 degrees.		
Questions:		
Question: What is the radius of the satellite's orbit given a circular 600 km orbit above Earth's surface? Please provide the answer in meters.	**Question:** What is the velocity of the satellite in a circular orbit at a radius of 6,978,140 meters? Please provide the velocity in meters per second (m/s).	**Question:** What is the change in velocity required to adjust the inclination of the satellite's orbit from 28 degrees to 20 degrees? Please provide the answer in meters per second (m/s) as a standard numeric expression.
Solution: To find the radius of the satellite's orbit, we add the Earth's radius to the altitude of the orbit. The Earth's average radius is approximately 6,378.14 km. Therefore, the radius r of the orbit is calculated as follows: $r = (6{,}378.14 + 600) \times 1{,}000 = 6{,}978{,}140$ m	**Solution:** The velocity V_i of a satellite in a circular orbit can be calculated using the formula: $V_i = \sqrt{\frac{GM}{r}}$ where $GM = 3.986005 \times 10^{14}\,\mathrm{m^3/s^2}$, is the gravitational parameter of Earth. Substituting the given values: $V_i = \sqrt{\frac{3.986005 \times 10^{14}}{6{,}978{,}140}}$ $V_i = 7{,}558$ m/s	**Solution:** The change in velocity ΔV required for an inclination change can be calculated using the formula: $\Delta V = 2 \times V_i \times \sin(\theta/2)$ where $\theta = 28 - 20 = 8$ degrees is the change in inclination. Substituting the known values: $\Delta V = 2 \times 7{,}558 \times \sin(8/2)$ $\Delta V = 1{,}054$ m/s
Answer: 6,978,140 m	**Answer:** 7558 m/s	**Answer:** 1054 m/s
Format: numeric **Source:** Braeunig	**Format:** numeric **Source:** Braeunig	**Format:** numeric **Source:** Braeunig

Fig. 2. Data structure of one example in the APBench. A single problem set structure composes of Content and *one or multiple* sets in Questions. Every set in Questions follows the same Question, Solution, Answer, Format, and Source structure.

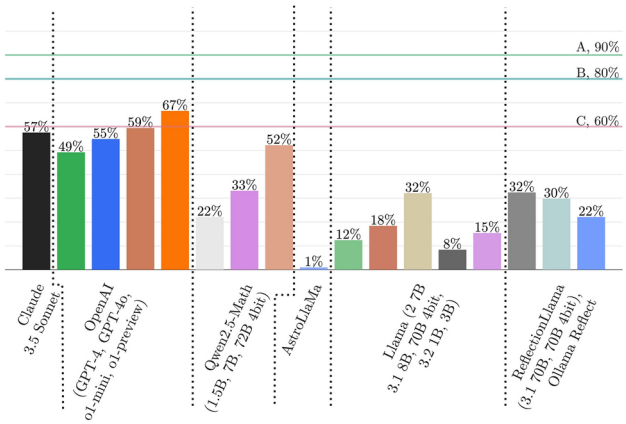

Fig. 3. The performance on APBench-γ from all the models in Table 3 is presented in comparison with thresholds of grade-A (90% accuracy), grade-B (80% accuracy) and grade-C (60% accuracy). The models are grouped according to their companies/collections. The left two groups are all closed source models; the rest are open source models. Closed source models perform well above open source models. However, with only one model (o1-preview) passing grade-C level, all of their performance are still far from reaching grade-B or grade-A.

The data curation flow is shown in Fig. 3. Specifically, the selected problem sets should fulfill the following **data selection criteria**:

- *Domain-specific.* The chosen problem sets require domain-specific knowledge, clear statement of the problem, and intermediate numerical computation ability.
- *Solution process availability.* To support the analysis of LLMs' ability to solve the problem sets, detailed solution processes would be required in the data to serve as a guideline for comparison purpose and guidelines.
- *Various levels of difficulty.* The problem sets cover various fundamental astrodynamics topics at different difficulty levels. This would further support analyzing the LLMs' current ability and potential improvements.

As shown in Fig. 2, the benchmark is provided in the following structure: a single problem set structure is composed of Content and *one or multiple* sets in Questions. Every set in Questions follows the same Question, Solution, Answer, Format, and Source structure.

APBench's open question formulation and space engineering focus make it a challenging benchmark. In space engineering practice, problems are not confined to a predefined subset; thus, it is crucial to avoid predetermined setups or multiple-choice formats. Constructing multiple-choice options inherently involves solving the problem beforehand. Either including the multiple choices in a prompt, which leaks the information necessary for large language model to find, or excluding them, which equivalents to direct evaluation on the

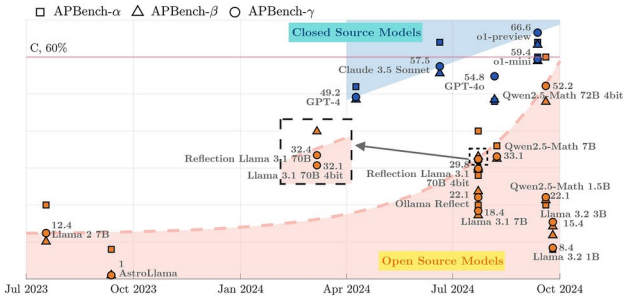

Fig. 4. We present the models performance evolution on APBench-α, APBench-β, APBench-γ, together with the models released date as listed in Table 3. The models are grouped as closed source models and open source models. Despite of the individual model's performance variation on the three Open APBench datasets, the difference between closed source models' performance and open source models' performance is clear. This temporal evolution also indicates a narrowing performance gap between the closed source model and the open source model. We further extrapolate the evolution bounds of the two groups by fitting on a subset of models' performance on APBench-γ in each group and find that year 2025 and 2026 are going to witness a grade-A LLM performance on APBench.

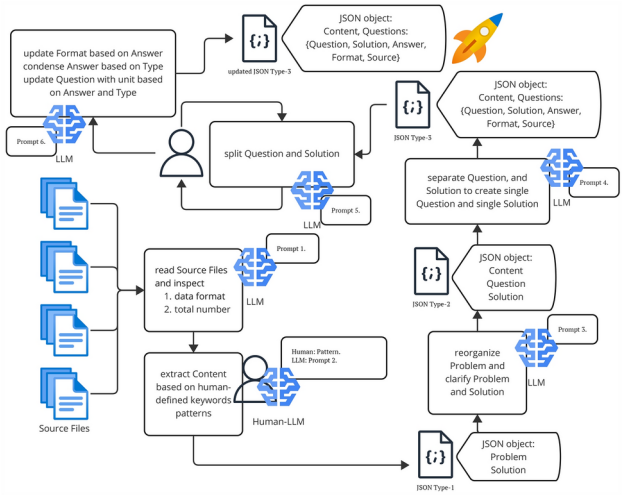

Fig. 5. Benchmark curation process workflow and the list of prompts should follow the section Benchmark Dataset Curation Process.

answer, provides little improvement to the setup. The complexity is not only revealed in the example case shown in Fig. 2 but also in the totally different procedures and steps taken by different language models in their attempts to answer the proposed question as illustrated by Figs. 4 and 5.

Answer format classification

Labeling questions is crucial for conducting a reasonable evaluation. We categorize the questions into two types: numeric and message. A numeric label indicates that the expected answer should focus on providing a realistic number, whereas a message label allows for a more relaxed structure of the answer. The Format is primarily determined by the nature of the answer. During the curation of the problem set, this labeling is also used in refining the Question section by specifying details such as units and whether the answer should be exclusively numeric. This additional information is incorporated into the curated question to enhance clarity and precision.

Open bench and private bench

We summarize the source information in Table 1 for the open APBench and Table 2 for the private APBench. We list the sources with acronyms for easy tracking in the tables; for open APBench, we also provide the corresponding websites. The open APBench sources its content from online available information. Open APBench is the primary benchmark for people of interest to directly test their models' abilities in solving space engineering problems. We provide this public benchmark also for validation and reproduction purposes. In the meantime, we also generate a private APBench. We choose to make this dataset private due to the copyright concern, which makes it unsuitable for public release. Instead, we provide only the number of problems we

[HTML]C0C0C0 Acronym	[HTML]C0C0C0Source	# Problems
Gordon	Introduction to Aerospace Flight Vehicles[a][21]	17
Qualify	PhD Qualifying Examination[b][22]	8
Braeunig	Rocket & Space Technology: Orbital Mechanics[c][23]	144
Lynnane	Introduction to Orbital Mechanics[d][24]	130
		Total: 299

Table 1. Summary of the source for open APBench and the number of problems from each source. [a]https://eaglepubs.erau.edu/introductiontoaerospaceflightvehicles. [b]https://phas.ubc.ca/sites/default/files/users/oser/astroq2016a.pdf. [c]http://www.braeunig.us/space/index.htm. [d]https://colorado.pressbooks.pub/introorbitalmechanics/.

[HTML]C0C0C0 Acronym	[HTML]C0C0C0Source	# Problems
Ashish	Atmospheric And Space Flight Dynamics[25]	78
BMW	Fundamentals of Astrodynamics[26]	167
Vallado	Fundamentals of Astrodynamics and Applications[27]	106
Mauri	The Three-Body Problem[28]	30
Curtis	Orbital Mechanics for Engineering Students[29]	586
Chegg	Orbital Mechanics for Engineering Students 3rd Edition Solutions[30]	522
Kluever	Space Flight Dynamics[31]	332
Schaub	Analytical Mechanics of Space Systems[32]	169
		Total: 1990

Table 2. Summary of the source for private APBench and the number of problems from each source. The source are published documents so we list the official names.

created from the database. The curation process described in Fig. 3 is well-suited for the reader interested in working with the listed sources or further expand to other sources.

Ethical statement

We conducted a manual examination of our dataset to ensure that it does not contain any sensitive background information or raise ethical concerns. The benchmark dataset is intended solely for academic use. Its collection complies with the Fair Use Law in the United States of America.

Minimization of data leakage risks

Canary string is employed in the field to reduce the risk of leakage into foundation model training corpora[20,33]. It intends to help researchers filter our APBench dataset out of web-scraped training corpora, so that models do not train on the benchmark set.

We request that you do not reveal examples from this dataset in plain text or images online, to reduce the risk of leakage into foundation model training corpora. Additionally, we incorporate this canary string

`psa:4c7a:89f2b3-12f4-8cbd-0dcb-a1ff0e4d`

into the dataset, to make it easier for training corpora to filter out our benchmark.

Model performance analysis

The LLMs' performances on APBench are shown in Table 3. Their evaluation of the APBench-γ, the largest open APBench, is shown in Fig. 6. The correctness on individual questions can be found in Supplementary Table S1. The majority of model performance is below the 60% threshold, which represents a passing C-grade for space engineering. OpenAI's o1-preview leads the performance, though its advantage is not significant. The model with superior math performance also performs well under the benchmark. Interestingly, the first fine-tuned model on arXiv's astronomy abstracts, AstroLlaMA, performs poorly despite claims of possessing knowledge

Model	Model Dates & Version	APBench-α (%)	APBench-β (%)	APBench-γ (%)	Comment
Sonnet	2024-06-14	64.0	55.6	57.5	Anthropic Claude 3.5
GPT-4	2024-04-09	52.0	48.5	49.2	OpenAI gpt-4-turbo-2024-04-09
GPT-4∗	2024-04-09	52.0	50.3	49.8	OpenAI gpt-4-turbo-2024-04-09 neutral progressive prompt
GPT4o	2024-08-06	48.0	48.5	54.8	OpenAI gpt-4o-2024-08-06
o1-mini	2024-09-12	60.0	58.9	59.4	OpenAI o1-mini-2024-09-12
o1-preview	2024-09-12	64.0	63.3	66.6	OpenAI o1-preview-2024-09-12
Qwen2.5-Math 1.5B	2024-09-19	20.0	21.3	22.1	Qwen2.5-Math-1.5B-Instruct
Qwen2.5-Math 7B	2024-08-08	36.0	32.5	33.1	Qwen2.5-Math-7B-Instruct
Qwen2.5-Math 72B	2024-09-19	60.0	47.9	52.2	Qwen2.5-Math-72B-Instruct, 4bit
AstroLlaMa	2023-09-12	8.0	1.0	1.0	UniverseTBD, astrollama
Llama 2 7B	2023-07-18	20.0	10.1	12.4	Llama-2-7b-chat-hf
Llama 3.1 8B	2024-07-23	20.0	17.2	18.4	Llama-3.1-8B-Instruct
Llama 3.1 70B	2024-07-23	28.0	29.6	32.1	Llama-3.1-70B-Instruct, 4bit
Llama 3.2 1B	2024-09-25	8.0	11.8	8.4	Llama-3.2-1B-Instruct
Llama 3.2 3B	2024-09-25	8.0	14.2	15.4	Llama-3.2-3B-Instruct
ReflectionLlama	2024-07-23	40.0	33.1	32.4	Reflection-Llama-3.1-70B
ReflectionLlama†	2024-07-23	28.6	30.2	29.8	Reflection-Llama-3.1-70B, 4bit
Ollama Reflection	2024-07-23	28.0	23.7	22.1	Ollama, 5084e77c1e10, 4bit

Table 3. Foundational / large language model performance on Open APBench. Three version of APBench, APBench-α, APBench-β, APBench-γ various in size, are provided. We evaluated both closed source models and open source models to have a full picture of SoTA LLM performance. ∗ GPT-4 model without progressive prompt. † Reflection-Llama-3.1-70B model loaded with 4-bit quantization.

and developing "scholarly" work in space and astronomy domains. Llama models, even the largest and most promising one, ReflectionLlama, fall behind the top performers in the benchmark.

Sonnet was expected to perform similarly to or even surpass GPT-4o, and it delivers performance in line with those expectations. OpenAI models follow a clear trend of improvement, progressing from GPT-4 to o1-preview, with o1-preview leading the APBench, albeit by a narrow margin. If we consider these models as four iterations of OpenAI's advancements, we might expect that two more iterations would be needed for OpenAI models to achieve a B-grade (80% accuracy), three to four iterations to reach an A-grade (90% accuracy), and five iterations to achieve the level of AI intelligence necessary for solving fundamental space engineering problems. It is important to note that we have employed relatively loose criteria for accuracy determination, and as AI models improve, a more stringent criterion will likely be required.

Llama's performances are also as expected. While Llama models generally lag behind in the APBench, a clear and intuitive growth trend is evident. Model size remains a significant factor in performance. The Llama 3.1 70B 4-bit quantized version delivers the best results, followed by Llama 3.1 8B. Interestingly, the newer, smaller models show notable improvement compared to older versions, as seen with the Llama 3.2 3B, which performs similarly to the 8B version. At the lower end of the performance spectrum, Llama 2 7B and Llama 3.2 1B show comparable results. The Llama 7B model, the earliest version, offers performance almost on par with the newer 1B model. This suggests that as the Llama models evolve, they increasingly concentrate their knowledge and capabilities.

The two Llama relatives—ReflectionLlama and AstroLlaMA – provide very interesting results. AstroLlaMA is fine-tuned on an extensive corpus of astronomy research abstracts, and it was expected to incorporate specialized knowledge from this scientific domain. However, its performance suffers when answering questions in APBench related to space engineering. In contrast, ReflectionLlama 3.1 performs reasonably well on APBench, maintaining a level of performance that was close to the top-performing Llama models. Notice that there is a difference in the target of conducting fine-tuning for AstroLlaMA and ReflectionLlama. AstroLlaMA's specialized knowledge traces back to textual content in the abstracts of research papers, while in APBench there is an emphasis on conducting calculations to solve engineering problems. Thus, AstroLlaMA's struggle with APBench hints at losing general reasoning and calculation skills while focusing on specialized knowledge. In the meantime, ReflectionLlama is fine-tuned to obtain a technique to conduct self-reflection in generating answers.

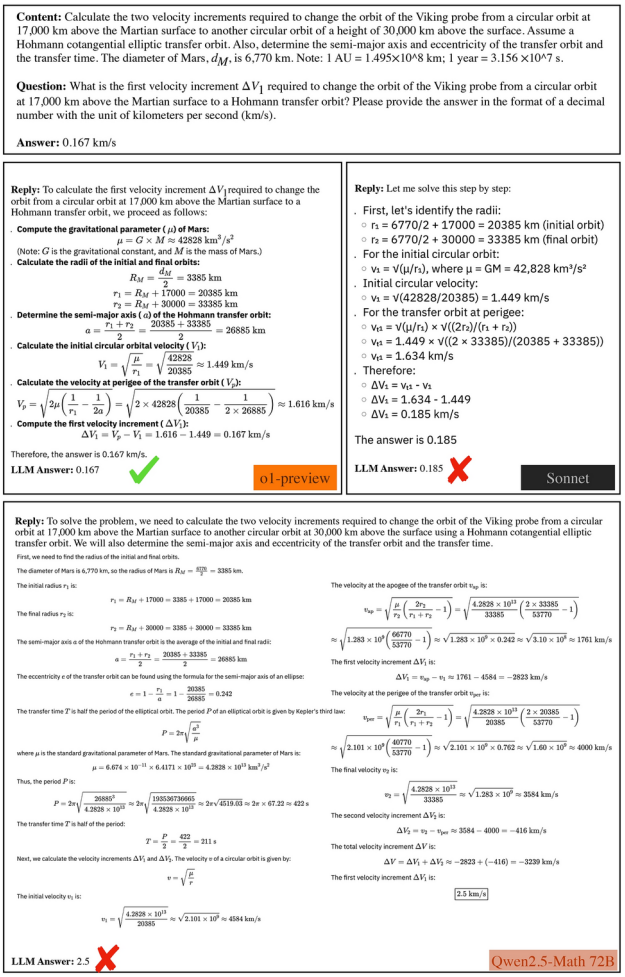

Fig. 6. o1-preview, Sonnect, Qwen2.5-Math-7B-Instruct performance on the example question.

This technique, similar to the effect of Chain-of-Thought[34], helps or at least does not hurt in solving engineering problems requiring calculations. Consequently, this leads to an interesting open question as to whether and how specialized knowledge could and should be incorporated with some abilities and techniques in providing solutions to advance the performance on solving problems requiring both – leading to a tailored foundational model that emphasizes both calculation-based and knowledge-based improvement.

The performance differences within the various versions of ReflectionLlama also follow an intuitive pattern. The full-floating-point version of ReflectionLlama 3.1 70B slightly outperforms its 4-bit quantized counterpart, which suggests that quantization does not drastically affect its ability to solve space engineering problems. This result reinforces the confidence that the performance of other quantized models is still representative of the full model. Besides, the framework also plays a role in performance differences. Ollama Reflection, a 4-bit quantized version of ReflectionLlama, performs less favorably than the 4-bit quantized version of ReflectionLlama from Hugging Face's transformer framework. This indicates that the framework used for fine-tuning and inference may influence the overall performance of Llama relatives.

The best performing open-source model in the APBench benchmark is Qwen2.5-Math 72B. Note that this model went through the 4-bit quantization. As expected, performance within the Qwen2.5-Math family improves with increasing model size. Besides, the smallest model, Qwen2.5-Math 1.5B, outperforms larger models like Llama 2 7B, Llama 3.1 8B, and Llama 3.2 1B and 3B. The medium-sized Qwen2.5-Math 7B model matches the performance of the best model from the Llama family, ReflectionLlama, despite Qwen2.5-Math 7B being only a 7B model compared to the larger 70B ReflectionLlama. The best-performing Qwen2.5-Math 72B model reaches a performance level similar to that of advanced closed-source models like Sonnet, GPT-4, and GPT-4o. This places Qwen2.5-Math 72B in a comparable tier to these high-performing closed-source models, demonstrating its potential in the space engineering domain.

Model performance evolution

The temporal evolution of the APBench results in Fig. 7 offers valuable insights into the development of the large language models. We observe a notable difference in the growth patterns of closed-source and open-source models. Closed-source models show a more linear progression in performance, reflecting steady improvements

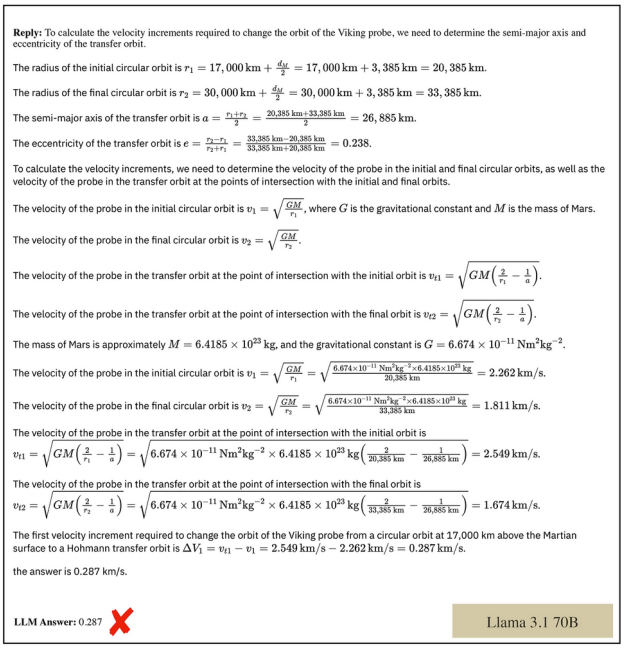

Fig. 7. Continuous Fig. 6, Llama 3.1 70B performance on the example question.

with each new iteration. In contrast, open-source models exhibit an exponential growth trend, indicating a faster pace of development and improvement. This disparity highlights the rapid advancements occurring within the open-source community, potentially driven by more open collaboration, frequent updates, and larger contributions from a diverse range of researchers and developers.

The open APBench is divided into three distinct versions: APBench-α, APBench-β, and APBench-γ, each defined by the different problem sets they include. APBench-α consists of the smallest benchmark, including problem sets from `Gordon` and `Qualify`. APBench-β builds upon APBench-α by adding problem sets from `Braeunig`, thus expanding the benchmark to a medium size. Finally, APBench-γ incorporates `Lynnane` into the problem sets, making it the full version of the APBench.

The performance on these three separate benchmarks is presented in Fig. 7, with markers □, △, and ◯ and their release dates or specific model dates. The closed-source models are marked in blue, and the open-source models in red. The models' abilities in space engineering are more effectively illustrated through the performance distribution, represented by their varying zero-shot accuracy on the different datasets. We also specify the accuracy in percentage for each model's performance on APBench-γ, rounded to the first decimal point. The performance across the three benchmarks for a specific model is close to each other. This consistency results from both the stable construction of the benchmark and the models' ability to solve engineering problems. They are not particularly sensitive to any of the source data. Similar to Fig. 6, we highlight the C-grade level 60% line. The closed-source model Sonnet passes the C-grade line on APBench-α, while o1-preview performs above this line on all benchmarks. O1-mini's performance hovers around the C-grade line. These results further validate the generally superior performance of closed-source models, as the only open-source model to achieve a similar performance is the Qwen2.5-Math 72B model on APBench-α.

By separating the models into two distinct groups, we can extrapolate the evolution trends of each group, approximated by the selected Pareto Front evolution. We use APBench-γ, the full dataset, to approximate the performance evolution. For closed-source models, we select GPT-4, Sonnet, and o1-preview for the Pareto Front approximation, and a linear approximation effectively captures the growing trend. For open-source models, we choose Llama 2 7B, Reflection Llama 3.1 70B, and Qwen2.5-Math 72B as the Pareto Front, finding that an exponential approximation curve best fits the performance trend. From this separation, we observe a well-defined gap between closed-source and open-source models on APBench. However, we also notice that this gap is narrowing over time. The gap, which was approximately 30% in April 2024, has reduced to around 10% by October 2024. Open-source models, in general, are undergoing significant improvement and are on track to catch up with closed-source models in terms of APBench performance.

Model performance prediction

We anticipate a breakthrough in model performance in the upcoming years, based on the development trends revealed by the evolution analysis. Extrapolating from the performance trends, we expect that closed-source models will achieve 80% (B-grade) performance by January 10, 2025, and 90% (A-grade) performance by April 10, 2025, under the current scoring schema setup as shown in Fig. 1. For open-source models, although we have selected a number of models for prediction, we observe that the performance improvement within a model family, such as the Llama family, does not always follow an clear trend. The introduction of Qwen2.5-Math has significantly strengthened the overall open-source model field. The Pareto Front approximation suggests that

open-source models could reach 80% accuracy by November 2, 2024, and 90% accuracy around the same time in 2024. However, this was not observed when this paper was written on December 1, 2024. However, as the open source model evolution follows less the heritages from the same model family and companies heritage. We should expect large variation from the prediction. In fact, if we exclude Qwen2.5-Math from the open-source models and focus solely on Llama-related models, we expect B-grade performance by August 9, 2026, and A-grade performance by December 11, 2026. This variance makes it difficult to precisely predict the future performance of open-source models, but we expect significant growth and a potential breakthrough in 2025.

Lastly, it is important to note that we have set a consistent buffer area for the evaluation criteria in scoring schema. This buffer will need to be tightened in the future as the precision requirements for high-stakes space engineering applications increase.

Discussion

The development of APBench (Astrodynamics Problems Bench) represents a groundbreaking advancement in benchmarking Large Language Models (LLMs) for space engineering applications. Unlike existing benchmarks, which predominantly focus on general knowledge or high-school-level problems, APBench introduces domain-specific challenges that revolve around pivotal topics such as orbital mechanics, spacecraft dynamics, and mission design. By addressing these foundational aspects of space engineering, APBench bridges a crucial gap, enabling a more targeted evaluation of LLMs' capabilities in this highly specialized field.

Another key contribution of this work is the comprehensive evaluation of state-of-the-art (SoTA) models, encompassing both leading closed-source models and the rapidly maturing open-source alternatives. Our discussion explores the systematic construction of the evaluation framework, detailing the careful selection of problem sets and the design of adaptable scoring criteria. The evaluation reveals the strengths and limitations of these models in solving complex space engineering problems but also a compelling trend: while closed-source models currently dominate in terms of performance, open-source models are making rapid advancements and steadily narrowing the gap.

Moreover, we project the timeline for achieving critical performance milestones. By analyzing trends in model evolution, we forecast when both closed-source and open-source LLMs might reach accuracy levels suitable for practical deployment in space engineering tasks. These projections underscore the accelerating pace of improvement, offering a glimpse into a future where LLMs could play a transformative role in addressing real-world engineering challenges.

We make the open APBench readily accessible so that we could introduce the space engineering problems to the broader community, encouraging not only space engineers but also the general AI research community to explore those problems. On the other hand, going beyond the fundamental space engineering questions presented in APBench, there are more advanced topics worth incorporating into the benchmark in the future, such as frozen orbit, mean motion resonance, periodic and quasi-periodic orbits, rotational dynamics, to name a few. With a continuing effort to enlarge and improve the benchmark itself, we expect the growth of APBench being similar to MATH[16] where continuous growth and further exploration would help pushing the boundary of foundational models in space engineering. The integration between LLM and astronomy, a closely related field, has shown a promising direction in terms of benchmark building and methodology innovation[18,35], as well as multi-model application[36]. In addition, the incorporation of other tool using ability to deal with the high precision operation and other advanced techniques such as RAG could also enable a better performance in the future. Aiming for further assisting the growth of AGI explainability, we believe this foundational benchmark would support pinpointing the explainability for AI in space engineering, pushing AI's general acceptance in "the final frontier".

Methods

Related work

While LLMs have transformed the field of NLP (Natural Language Processing) and have shown vast potential across various fields, rapid advancements in LLM capabilities present the possibility that in the future, superhuman AI systems could help advance "the final frontier"—space engineering. To measure our ability to align LLMs for this purpose, we need evaluation datasets that characterize the trustworthiness of these models on typical classroom-level astrodynamics problems. These tests should serve as the preliminary threshold before any trust is to be put in the answers from an LLM for practical applications to space system. As LLMs are used to answer increasingly difficult questions and are challenged on harder and harder questions where the scalable oversight, a phenomenon describing the incompetency in human's generating solutions on their own to verify and test the improving difficulties LLM can solve, starts to appear[20,37], we would expect that, despite of the existing generally well-known issues in LLMs like hallucination[38], sycophancy[39], and textual reasoning limitation[40], among others, it is necessary to start evaluating the performance of LLMs on the specialized task of space engineering to reveal improvement potentials so as to both support the adoption of LLMs into space engineering and also to return valuable directions for the general AI development.

STEM benchmarks

Evaluating LLMs has made significant strides, and the evolvement of benchmarks has both proven the advancement of LLMs' improving abilities and led to the call for new benchmarks[41]. Under the realm of STEM, we have witnessed a great number of benchmark datasets.

Benchmarks such as GSM8K[15], designed for middle-school mathematics, and MATH[16], focused on high-school math competitions, have become standard in the field. For university-level mathematics, datasets like ProofNet[42] and OCWCourses[43] are widely recognized. Furthermore, Olympiad-level problems are covered by

datasets like MiniF2F[44] and AlphaGeometry[45], while the SAT dataset[46] includes problems from the College Board SAT examination. As LLM performance continues to improve, there is a demand for even more challenging benchmarks. This has led to the creation of benchmarks such as MathOdyssey[47], OlympiadBench[48], and the collective dataset MathInstruct[49]. Beyond math, the evaluation of coding capabilities is also crucial[40], as argued by CS-Bench[50] that Computer Science (CS) represents the intricacies of human intelligence, profoundly advancing artificial intelligence and modern society.

Building upon the progress in math-related benchmarking, other fields have begun to develop specialized benchmarks. In general physics and chemistry, benchmarks targeting college-level problems have been introduced. The latest dataset SciBench[51] formulates open questions incorporating both textual and image information. ScienceQA[52] provide datasets with multimodal multiple-choice science questions, answers, tasks, lecture notes, and detailed annotations. BIG-Bench[53] offers a general-purpose test suite with 204 tasks, ranging from multiple-choice to exact-match questions, while its extension, BIG-Bench Hard[54], introduces more complex Chain of Thought (CoT) prompts. SciEval[55] evaluates LLM's understanding and application through a mix of objective and subjective questions across various scientific disciplines. JEEBench[56] draws on pre-engineering-level scientific problems from college entrance exams, and AGIEval[57] assesses LLM performance on standardized exams such as college entrance and lawyer qualification tests.

More fields within STEM are recognizing the importance and promise of understanding LLMs' capabilities in their specific domains. In finance, the introduction of the FinBench benchmark[19] addresses the need to evaluate LLMs in more complex and specialized tasks. Their dataset serves as the first benchmark designed specifically for inherently complex financial problems, facilitating the integration of finance with the rapidly advancing landscape of LLM development. Similarly, in the field of requirements engineering, Norheim et al.[58] highlights the limitations posed by the absence of benchmarks in advancing the understanding of LLMs in requirement engineering. The authors propose the development of benchmark datasets to enable systematic exploration and evaluation of LLM and NLP methods across various RE tasks in the future. In addition, as they cite, "the lack of benchmarks also makes it hard to benchmark the novel LLMs against past methods that have been applied (e.g., rule-based, feature-based ML)[59]". This underscores the critical role of benchmarks in evaluating the transformative capabilities LLMs pose across STEM disciplines, as there is a significant difference between LLMs and traditional machine learning.

The curated benchmark, APBench, is strongly inspired by the need to improve benchmark difficulties in recent years. This includes scaling up benchmarks, as seen in math-related ones, and increasing the difficulty levels, as observed in physics-related benchmarks. Additionally, APBench addresses the demand for field-specific benchmarks, with the field of space engineering raising unique concerns about LLMs' specialized capabilities.

Data curation process and principles

We collected each problem from source PDF documents and processed them locally using a combination of scripts and manual operations to divide the source contents into two sections: Content and Questions. Within Questions, each individual set has five items: Question, Solution, Answer, Format, and Source, while sharing the same Content. To streamline this process, we wrote scripts employing OpenAI's API to conduct batch process. The script identifies chunks of information, separates them into the the above mentioned sections, and extracts the problems and solution processes while organizing the content and multiple problems accordingly. The final answers are classified as either *numeric* or *message*, and the source is indicated by an *acronym* as shown in Tables 1 and 2. Numeric answers are converted to floating-point numbers while message answers retain their original expressions during model performance evaluation. Additionally, step-by-step solutions are included for each problem. The section Content is reused across multiple problems without duplication.

Progressive problems settings

The source file may have multiple steps for a single problem set with a shared background information. We formulate the problem set structure as a shared content with multiple problems. The problems are organized as progressive problems. During evaluation process as illustrated in Fig. 1, we evaluate the correctness of the solution to a single problem and decide if the answered problem should be returned as part of the prompt based on the following condition: in the case of correctly solving a problem, the correctly answered problem along with its solution is attached to the content information. This provides additional context to the model, as many problems in the dataset under the same content are progressive. Figure 2 illustrates a case in which the answer to the second question could be directly used to solve the third question as the middle question is a necessary step for the right question.

We have also tested the performance of the model without information from past problems; this means that the model would treat each question independently. Our experiments indicate a similar performance; whether correctly answered questions are included in the prompt or questions are answered independently. The two cases of GPT-4 and GPT $-$ 4* in Table 3 indicate that for GPT-4 model (gpt-4-turbo-2024-04-09), employing progressive prompt and employing independent prompt result in similar performance. At first glance, this seems counterintuitive, but according to Park et al.[60], generative models possess "hidden capabilities" related to prompting. LLMs could answer a question correctly with a right way of prompting the question. A similar point was made in Zhu et al.[11,14,61] regarding the concept of knowledge storage and extraction, suggesting that a question which can be correctly answered and which forms an important step for the next question can always be answered correctly. This implies that the correct answering of a question is determined not by its formulation but by its content. In essence, LLMs know what they know, and explicitly including what it could know in prompting is not necessarily the best way to extend their capability.

Despite the similar testing performance observed for both employing progressive prompt and not, we chose to structure the benchmark with progressive problems. This approach provides a more concise way of formulating

the dataset and avoids unnecessary duplication of content information. Moreover, this structure reflects the inherent setup of the problems, offering additional information that could be further explored in future studies.

Data source and ethical statement

To construct the dataset, we select a combination of online resources. We focus on including questions with more challenging, open-ended answers rather than the multiple-choice format commonly used in previous works[17,51,52,62]. The problem sets include questions requiring both numeric and message answers. This aligns with our goal of providing the benchmarking on the topics of astrodynamics in a general way, since astrodynamics problems requiring either answers are common in the field. Moreover, since a message answer could include analytical solutions and other intuitive insights, those solutions have a broader impact than numerical ones tailored to a specific scenario.

Models and experiment setup

We evaluate the benchmarking dataset on both open-source and closed-source LLMs on our local machine, employing a zero-shot prompting strategy. The evaluation workflow is depicted by an example case in Fig. 7. For each set of questions within a single problem set, the Content and the Question are combined in the prompt to request the LLM's final solution and answer. The answer from the LLM is then evaluated by comparing it to the correct answer in the problem set. In the zero-shot prompt setting, no prior examples or specific training data are provided. The model is expected to generate accurate responses based solely on its pre-existing knowledge and its ability to generalize from the prompt. This setup tests the LLM's capacity to solve problems without relying on previously seen examples, thereby evaluating its inherent astrodynamics knowledge and problem-solving capabilities in the context of the task at hand.

Recent development in fine-tuning has made the process of creating specialized LLM much more possible through not only programming techniques, but also the process of the knowledge and methods. The most recent update from OpenAI's new model, OpenAI o1-preview, is the newest area of improvement. It demonstrates remarkable advancements in critical areas such as mathematics, physics, chemistry, and coding. These improvements highlight the ongoing refinement of LLM capabilities in solving intricate problems across STEM disciplines. Simultaneously, the rapid growth of open-source models has led to an explosion of foundational models, accompanied by diverse and innovative methodologies for creating and optimizing LLMs.

We selected a diverse set of leading LLMs to evaluate their performance on APBench. On the closed-source side, we chose several models from OpenAI, including GPT-4 (gpt-4-turbo-2024-04-09), GPT-4o (gpt-4o-2024-08-06), o1-mini (o1-mini-2024-09-12), and o1-preview (o1-preview-2024-09-12). These models represent a growing trend of confidence in their ability to solve a variety of problems efficiently. Additionally, we included Claude 3.5 Sonnet from Anthropic, released on 2024-06-14. Sonnet is reported to achieve similar or superior performance compared to GPT-4o on several math and coding-related benchmarks[63]. On the open-source side, we selected multiple versions of Llama from Meta, one of the most popular open-source models. Specifically, we included Llama 2 7B (Llama-2-7b-chat-hf), Llama 3.1 8B (Llama-3.1-8B-Instruct), Llama 3.1 70B (Llama-3.1-70B-Instruct), Llama 3.2 1B (Llama-3.2-1B-Instruct), and Llama 3.2 3B (Llama-3.2-3B-Instruct). Due to hardware limitations, the large Llama 3.1 70B model was loaded using a 4-bit quantization method. These models cover a wide range of sizes and evolutionary steps within the Llama family. We also included AstroLLaMA[64] available on 2023-09-12, a fine-tuned model trained on arXiv astronomy abstracts, aiming to be able to conduct "scholarly astronomy". Given the foundational connections between astronomy and space engineering, we selected this model to explore whether its training on astronomy content would aid in solving space engineering problems. Another significant inclusion was the *Reflection Llama* series. For the Reflection Llama 3.1 70B model (Reflection-Llama-3.1-70B), we evaluated both the full floating-point version and a 4-bit quantized version to test the impact of quantization. Additionally, we tested the Ollama Reflection model, another 4-bit quantized version of Reflection Llama under the Ollama framework. According to its developers, "trained using Reflection-Tuning, a technique developed to enable LLMs to fix their own mistakes", Reflection Llama claimed to achieve SOTA performance on multiple benchmark on twitter. Last but the not the least, we also include Qwen2.5-Math of various size 1.5B (Qwen2.5-Math-1.5B-Instruct), 7B (Qwen2.5-Math-7B-Instruct), and 72B (Qwen2.5-Math-72B-Instruct). For Qwen2.5-Math 72B model, we loaded it using 4-bit quantization. Qwen2.5-Math is reported to outperform most contemporary models in the mathematics domain.

Automatic scoring pipeline and evaluation criteria

APBench is designed with the target of providing adequate evaluation for deploying foundational models in real-world engineering scenarios. APBench requires a prior Format of being either a numeric or message answer; an automatic scoring pipeline is developed accordingly.

The answer from LLMs is evaluated using predefined rules based on the Format of the reference answer in a problem set. The scoring schema is shown in Fig. 1. For numeric Format, the answer is converted from text strings to floating numbers, and numeric values are extracted for evaluation. An error margin, which adapts to the true answer of the problem, is used to determine if the answer is approximately correct. Similar error margin criteria has been adopted in Wang et al.[51]; and this criteria can be adjusted, either tightened or loosened, in future evaluations. We employ an error margin expression of $(0.1 + 0.01 \log(|\text{Answer}|))$ to evaluate if the difference between an LLM answer and the reference answer is acceptable. In the case of this calculated error margin exceeds the range between 0 and 1, a fixed error margin of 0.1 is used. The known challenges associated with arithmetic operations were the primary consideration in determining the error boundary.

On the other hand, if the Format of the expected Answer is message, a combination of LLM-as-a-judge evaluation and an embedding-level evaluation is employed. The criteria for problem in Message format is a weighted score of $0.5(\text{LLM-Judge} + \text{Embedding-Similarity})$, and a score higher than 0.6 is considered

accurate; else wrong. A similar weighted way of evaluating the performance is also employed in Wang et al.[65] while the single employment of LLM-as-a-judge was employed in Foutter et al.[66]. The LLM-as-a-judge employs GPT4o (gpt-4o-2024-08-06) to evaluate if the LLM Answer and the reference Answer are the same between a scale of 0 to 10 and then normalize that number to 1. LLM's ability to incorporate different sequence, formulation, and expression of the same expression gives it a unique role as a judge. The evaluation is also augmented by employing embedding-level evaluation. We select the pretrained model all-MiniLM-L6-v2[67] to map Answer and Model Answer to two 384 dimensional dense vectors. As the expressions may use different symbols and orders, employing embedding-level similarity check could also support the decision from LLM-as-a-judge process. Embedding similarity is checked with the cosine similarity, naturally scaled between 0 and 1, adding vagueness into the criteria.

Benchmark dataset curation process

The benchmark dataset curation process is depicted in Fig. 3. We take the source files in various types (PDF, PNG, JPEG, etc.) and feed them into the block that inspect the source files. Prompt 1 is used to get the number of examples in the file. It is more efficient to include human in the next step of editing the pattern found by LLM and use that pattern with Prompt 2 to create the first JSON file JSON Type-1. JSON Type-1 only has two attributes: Problem and Solution. Then we use Prompt 3 to separate Problem into Content and Question, creating JSON Type-2 JSON file. However, the Question identified could be a combination of multiple questions, so we separate the Question and Solution simultaneously into multiple Questions and corresponding Solutions, generating JSON Type-3 JSON file.

Furthermore, we employ a two-step operation with an if logic condition, to split Question and Solution further into more detailed Question and Solution. The corresponding prompts are Prompt 5 step 1 and Prompt 5 step 2. A human interference is introduced to evaluate the completeness of Question and Solution splitting. If it needs further splitting, we repeat the process. But if the Question and Solution cannot be further separated, we move to the step of updating Format, condensing Answer, and updating Question part. The corresponding prompt is also a two-step prompt: Prompt 6 step 1 to update the Format attribute, and Prompt 6 step 2 to use the updated Format attribute to condense Answer and update Question when the Format is numeric while do nothing for message Format.

Throughout the process, we are able to conduct large amount of automatic operation over the creation of the benchmark datasets. Human interference is still needed in multiple places such as determining whether to further split Question and Solution; determining the quality of the Format determination; also as early as pattern determination in Source files. Finally, we achieve the updated JSON Type-3 JSON file with the attributes shown in Fig. 2.

The list of prompts are:

Prompt 1:	How many examples are in this file?
Prompt 2:	Extract the example {Pattern} in the pdf and separate the content into problem and solution.
Prompt 3:	Reorganize Problem: {prob_str}, and Solution: {solu_str} to the format of Content:, Question:, and Solution: 1. Separate the problem part into content: and question:. Keep everything in the problem. 2. Clarify the question by formulating as a question and mark with a question mark. If the question is not apparent, summarize a question. 3. Keep everything (codes, procedures, comments, derivations, and drawings) in Solution and add some clarification by clarifying the final answer using "the answer is:".
Prompt 4:	Create a single JSON object with the field of Questions where the number of Question matches the number of Solution. Give me the result in the format of: {{"Content":..., "Questions":[{{"Question":..., "Solution":..., "Answer":..., "Format":..., "Source":... }}, {{"Question":..., "Solution":..., "Answer":..., "Format":..., "Source":... }}]}}. The original JSON object is {data}.
Prompt 5 Step 1	Here is a question and its solution from a technical problem set. Determine if this question should be split into multiple questions so that each part has a single, specific answer. Question: "{question}" Solution: "{solution}" Analyze and respond with "Yes, split into multiple questions" or "No, keep as a single question." If splitting, suggest how to divide the question clearly.
Prompt 5 Step 2	*For problem set with response "Yes, split into multiple questions"* The following question has been identified as needing to be split into multiple parts. Please divide it into clear, separate questions, each with a single answer. Format the output for each part as follows: - "Question": <text> - "Solution": <detailed solution> - "Answer": <final answer summary> - "Format": "Text" - "Source": "{source}" Original Question: "{question}" Original Solution: "{solution}" Provide each part in this structured format. *For problem set with response "No, keep as a single question."* Keep the original format.
Prompt 6 Step 1	You are a helpful assistant. Classify the following answer into one of two categories: - "numeric" if the goal of the answer is primarily to provide a numeric value or purely a calculation result. - "message" if the answer contains symbolic expressions, formulas, or explanations in text form. Answer: "{answer}" Respond with one of the two words only: "numeric" or "message".
Prompt 6 Step 2	Given the following scientific answer, convert it into a clean numeric format with units. If the answer provides multiple values, choose a single value in the most commonly used unit. If the answer is in the form of a fraction, multiplier, or scientific notation, process it into a standard numeric expression. Also, suggest an additional phrase for the question that specifies the unit or format of the answer if needed. Answer: {answer} Respond with: - Numeric Answer: {numeric value with unit} - Question Update: {additional text to specify the expected format of the answer}

Model performance examples

Data availability

The open APBench datasets are available at https://huggingface.co/datasets/woodywu/APBench.

Received: 10 December 2024; Accepted: 18 February 2025
Published online: 07 March 2025

References

1. Mirchandani, S. et al. Large language models as general pattern machines (2023). arXiv preprint arXiv:2307.04721.
2. Xu, Z. et al. DriveGPT4: Interpretable end-to-end autonomous driving via large language model (2024). arXiv preprint arXiv:2310.01412.
3. Radosavovic, I. et al. Humanoid locomotion as next token prediction (2024). arXiv preprint arXiv:2402.19469.
4. Yoshikawa, N. et al. Large language models for chemistry robotics. *Auton. Robot.* **47**, 1057–1086. https://doi.org/10.1007/s10514-023-10136-2 (2023).
5. Wang, L. et al. GenSim: Generating robotic simulation tasks via large language models (2024). arXiv preprint arXiv:2310.01361.
6. Chen, Y. et al. Autotamp: Autoregressive task and motion planning with llms as translators and checkers. in *2024 IEEE International conference on robotics and automation (ICRA)*, 6695–6702, https://doi.org/10.1109/ICRA57147.2024.10611163 (2024).
7. Austin, J. et al. Program synthesis with large language models (2021). arXiv preprint arXiv:2108.07732.
8. Rodriguez-Fernandez, V. et al. Language models are spacecraft operators (2024). arXiv preprint arXiv:2404.00413.
9. Zucchelli, E. M. et al. Fine tuned language models as space systems controllers. In *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference* (Broomfield, CO, 2024).
10. Foutter, M. et al. Adapting a foundation model for space-based tasks (2024). arXiv preprint arXiv:2408.05924v1.
11. Allen-Zhu, Z. & Li, Y. Physics of language models: Part 3.3, knowledge capacity scaling laws (2024). arXiv preprint arXiv:2404.05405.

12. Yu, L. et al. MetaMath: Bootstrap your own mathematical questions for large language models (2024). arXiv preprint arXiv:2309.12284.
13. Fedorenko, E., Piantadosi, S. T. & Gibson, E. A. Language is primarily a tool for communication rather than thought. *Nature* **630**, 575–586. https://doi.org/10.1038/s41586-024-04375-1 (2024).
14. Allen-Zhu, Z. & Li, Y. Physics of language models: Part 3.1, knowledge storage and extraction (2024). arXiv preprint arXiv:2309.14316.
15. Cobbe, K. et al. Training verifiers to solve math word problems (2021). arXiv preprint arXiv:2110.14168.
16. Hendrycks, D. et al. Measuring mathematical problem solving with the MATH dataset (2021). arXiv preprint arXiv:2103.03874.
17. Chen, W. et al. TheoremQA: A theorem-driven question answering dataset (2023). arXiv preprint arXiv:2305.12524.
18. Ting, Y.-S. et al. Astromlab 1: Who wins astronomy jeopardy!? (2024). arXiv preprint arXiv:2407.11194.
19. Xie, Q. et al. FinBen: A holistic financial benchmark for large language models (2024). arXiv preprint arXiv:2402.12659.
20. Rein, D. et al. GPQA: A graduate-level Google-proof Q &A benchmark (2023). arXiv preprint arXiv:2311.12022.
21. Leishman, J. G. *Introduction to Aerospace Flight Vehicles* (Pressbooks, 2023) (**Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International**).
22. Oser, S. M. Phd qualifying examination 2016 (2016).
23. Braeunig, R. A. Rocket & space technology (2017).
24. George, L. *Introduction to Orbital Mechanics* (Pressbooks, 2021) (**Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International**).
25. Tewari, A. *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB® and Simulink®* (Birkhäuser, 2007).
26. Bate, R. R., Mueller, D. D. & White, J. E. *Fundamentals of Astrodynamics* (Dover Publications, 1971).
27. Vallado, D. A. & McClain, W. D. *Fundamentals of Astrodynamics and Applications* 3rd edn. (Microcosm Press/Springer, 2007).
28. Valtonen, M. & Karttunen, H. *The Three-Body Problem* (Cambridge University Press, 2006).
29. Curtis, H. D. *Orbital Mechanics for Engineering Students* 3rd edn. (Butterworth-Heinemann, 2013).
30. Chegg. Orbital mechanics for engineering students 3rd edition solutions (2024). Solutions available with a subscription.
31. Kluever, C. A. *Space Flight Dynamics* 2nd edn. (Wiley, 2018).
32. Schaub, H. & Junkins, J. L. *Analytical Mechanics of Space Systems* 4th edn. (American Institute of Aeronautics and Astronautics, 2018).
33. Srivastava, A. & et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models (2023). arXiv preprint arXiv:2206.04615.
34. Lu, P. et al. Learn to explain: Multimodal reasoning via thought chains for science question answering (2022). arXiv preprint arXiv:2209.09513.
35. de Haan, T. et al. Astromlab 3: Achieving gpt-4o level performance in astronomy with a specialized 8b-parameter large language model (2024). arXiv preprint arXiv:2411.09012.
36. Carruba, V., Aljbaae, S., Smirnov, E. & Caritá, G. Vision transformers for identifying asteroids interacting with secular resonances. *Icarus* **425**, 116346. https://doi.org/10.1016/j.icarus.2024.116346 (2025).
37. Amodei, D. et al. Concrete problems in AI safety (2016). arXiv preprint arXiv:1606.06565.
38. Zhang, Y. et al. Siren's song in the AI ocean: A survey on hallucination in large language models (2023). arXiv preprint arXiv:2309.01219.
39. Sharma, M. et al. Towards understanding sycophancy in language models (2023). arXiv preprint arXiv:2310.13548.
40. Chen, Y., Jhamtani, H., Sharma, S., Fan, C. & Wang, C. Steering large language models between code execution and textual reasoning (2024). arXiv preprint arXiv:2410.03524.
41. Maslej, N. et al. Artificial intelligence index report 2024 (2024). arXiv preprint arXiv:2405.19522.
42. Azerbayev, Z. et al. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics (2023). arXiv preprint arXiv:2302.12433.
43. Lewkowycz, A. et al. Solving quantitative reasoning problems with language models (2022). arXiv preprint arXiv:2206.14858.
44. Zheng, K., Han, J. M. & Polu, S. MiniF2F: A cross-system benchmark for formal Olympiad-level mathematics (2022). arXiv preprint arXiv:2109.00110.
45. Trinh, T. et al. Solving olympiad geometry without human demonstrations. *Nature* **625**, 476–482. https://doi.org/10.1038/s41586-023-06747-5 (2024).
46. Azerbayev, Z. et al. Llemma: An open language model for mathematics (2024). arXiv preprint arXiv:2310.10631.
47. Fang, M., Wan, X., Lu, F., Xing, F. & Zou, K. MathOdyssey: Benchmarking mathematical problem-solving skills in large language models using Odyssey math data (2024). arXiv preprint arXiv:2406.18321.
48. He, C. et al. OlympiadBench: A challenging benchmark for promoting agi with Olympiad-level bilingual multimodal scientific problems (2024). arXiv preprint arXiv:2402.14008.
49. Yue, X. et al. MAmmoTH: Building math generalist models through hybrid instruction tuning (2023). arXiv preprint arXiv:2309.05653.
50. Song, X. et al. CS-Bench: A comprehensive benchmark for large language models towards computer science mastery (2024). arXiv preprint arXiv:2406.08587.
51. Wang, X. et al. SciBench: Evaluating college-level scientific problem-solving abilities of large language models (2024). arXiv preprint arXiv:2307.10635.
52. Lu, P. et al. Learn to explain: Multimodal reasoning via thought chains for science question answering (2022). arXiv preprint arXiv:2209.09513.
53. Ghazal, A. et al. BigBench: towards an industry standard benchmark for big data analytics. in *ACM SIGMOD Conference* (2013).
54. Suzgun, M. et al. Challenging BIG-Bench tasks and whether chain-of-thought can solve them (2022). arXiv preprint arXiv:2210.09261.
55. Sun, L. et al. Scieval: A multi-level large language model evaluation benchmark for scientific research (2024). arXiv preprint arXiv:2308.13149.
56. Arora, D., Singh, H. G. & Mausam. Have LLMs advanced enough? A challenging problem solving benchmark for large language models (2023). arXiv preprint arXiv:2305.15074.
57. Zhong, W. et al. AGIEval: A human-centric benchmark for evaluating foundation models (2023). arXiv preprint arXiv:2304.06364.
58. Norheim, J. et al. Challenges in applying large language models to requirements engineering tasks. *Des. Sci.* **10**, e16. https://doi.org/10.1017/dsj.2024.8 (2024).
59. Lim, S. C. *A Case for Pre-Trained Language Models in Systems Engineering*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2022).
60. Park, C. F., Okawa, M., Lee, A., Lubana, E. S. & Tanaka, H. Emergence of hidden capabilities: Exploring learning dynamics in concept space (2024). arXiv preprint arXiv:2406.19370.
61. Allen-Zhu, Z. & Li, Y. Physics of language models: Part 3.2, knowledge manipulation (2024). arXiv preprint arXiv:2309.14402.
62. Lu, P. et al. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning (2021). arXiv preprint arXiv:2105.04165.
63. Anthropic. Claude 3.5 Sonnet model card addendum (2024). Accessed: 2024-12-02.
64. Nguyen, T. D. et al. AstroLLaMA: Towards specialized foundation models in Astronomy (2023). arXiv preprint arXiv:2309.06126.
65. Weng, Y. et al. Large language models are better reasoners with self-verification (2023). arXiv preprint arXiv:2212.09561.

66. Foutter, M. et al. Adapting a foundation model for space-based tasks (2024). arXiv preprint arXiv:2408.05924.
67. Reimers, N. & Gurevych, I. Sentence-BERT: Sentence embeddings using siamese bert-networks. in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (2019).

Acknowledgements
Research was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Author contributions
D.W., E.M.Z., and Y.C. conceived the project. D.W. and R.Z. built the benchmark evaluation code. D.W. built the benchmark curation code and the benchmark datasets, carried out experiments, analyzed the results, and drafted the manuscript. E.M.Z. and Y.C. helped with manuscript structure. D.W., E.M.Z., and R.L. worked on benchmark source file selection, model selection, and experiment design. All authors reviewed the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-91150-5.

Correspondence and requests for materials should be addressed to D.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.