ATMOSSCI-BENCH: Evaluating the Recent Advances of Large Language Models for Atmospheric Science

Chenyue Li
CSE, HKUST
Clear Water Bay, Hong Kong SAR
`clieh@connect.ust.hk`

Wen Deng
CE, HKUST
Clear Water Bay, Hong Kong SAR
`wdengan@connect.ust.hk`

Mengqian Lu
CE, HKUST
Clear Water Bay, Hong Kong SAR
`cemlu@ust.hk`

Binhang Yuan[*]
CSE, HKUST
Clear Water Bay, Hong Kong SAR
`biyuan@ust.hk`

Abstract

The rapid advancements in large language models (LLMs), particularly in their reasoning capabilities, hold transformative potential for addressing complex challenges and boosting scientific discovery in atmospheric science. However, leveraging LLMs effectively in this domain requires a robust and comprehensive evaluation benchmark. Toward this end, we present ATMOSSCI-BENCH, a novel benchmark designed to systematically assess LLM performance across five core categories of atmospheric science problems: hydrology, atmospheric dynamics, atmospheric physics, geophysics, and physical oceanography. ATMOSSCI-BENCH features a dual-format design comprising both multiple-choice questions (MCQs) and open-ended questions (OEQs), enabling scalable automated evaluation alongside deeper analysis of conceptual understanding. We employ a template-based MCQ generation framework to create diverse, graduate-level problems with symbolic perturbation, while OEQs are used to probe open-ended reasoning. We conduct a comprehensive evaluation of representative LLMs, categorized into four groups: instruction-tuned models, advanced reasoning models, math-augmented models, and domain-specific climate models. Our analysis provides some interesting insights into the reasoning and problem-solving capabilities of LLMs in atmospheric science. We believe ATMOSSCI-BENCH can serve as a critical step toward advancing LLM applications in climate services by offering a standard and rigorous evaluation framework. The source code of ATMOSSCI-BENCH is available at [`https://github.com/Relaxed-System-Lab/AtmosSci-Bench`].

1 Introduction

Large language models (LLMs) [1], especially in their reasoning capabilities, have recently achieved remarkable progress, offering transformative potential for addressing complex challenges in atmospheric science [2, 3, 4, 5]. More recently, increasingly powerful LLMs have accelerated progress in AI4S (AI for Science), enabling a paradigm shift in scientific discovery. With their growing capabilities, LLMs show the potential to act as "AI Scientists," partially assisting—or even autonomously conducting—hypothesis generation, experimental design, execution, analysis, and refinement [6, 7, 8, 9, 10, 11]. To advance AI for Atmospheric Science and enable the development of reliable and effective LLM-based applications for climate-related tasks, it is crucial to recognize

---

[*] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

that LLMs themselves serve as a foundational core. Assessing whether current LLMs are capable of reasoning about problems in this domain is therefore a prerequisite, which calls for *a robust and comprehensive evaluation framework. Such a benchmark is essential to systematically assess the performance of LLMs across a diverse array of atmospheric science problems*, ensuring their utility, accuracy, and robustness in this critical domain.

Atmospheric science presents unique and complex challenges, ranging from micro-scale processes like cloud dynamics to global-scale climate systems. To ensure that LLMs can effectively contribute to solving these real-world problems, it is essential to establish a benchmark that evaluates their performance, especially their reasoning and interpretative abilities. Such a well-designed benchmark will not only foster innovation but also provide a standardized framework for assessing the utility, accuracy, and robustness of LLMs in this field.

Atmospheric science problems include essential differences from the classic mathematical and physical problems commonly found in existing LLM benchmarks [12, 13]. This field is inherently interdisciplinary, requiring the integration of theoretical analytical modeling skills with real-world phenomena and knowledge. Concretely, atmospheric science involves analyzing and synthesizing heterogeneous data types, such as spatial coordinates, temperatures, wind patterns, and empirical estimates, which are often presented in varied formats and units. Furthermore, solving these problems necessitates the selection of appropriate physical models and mathematical methods to ensure accuracy, adding layers of complexity beyond traditional benchmarks. As such, constructing a benchmark tailored to atmospheric science is a necessary complement to existing evaluations, enabling a more comprehensive assessment of LLMs' reasoning capabilities.

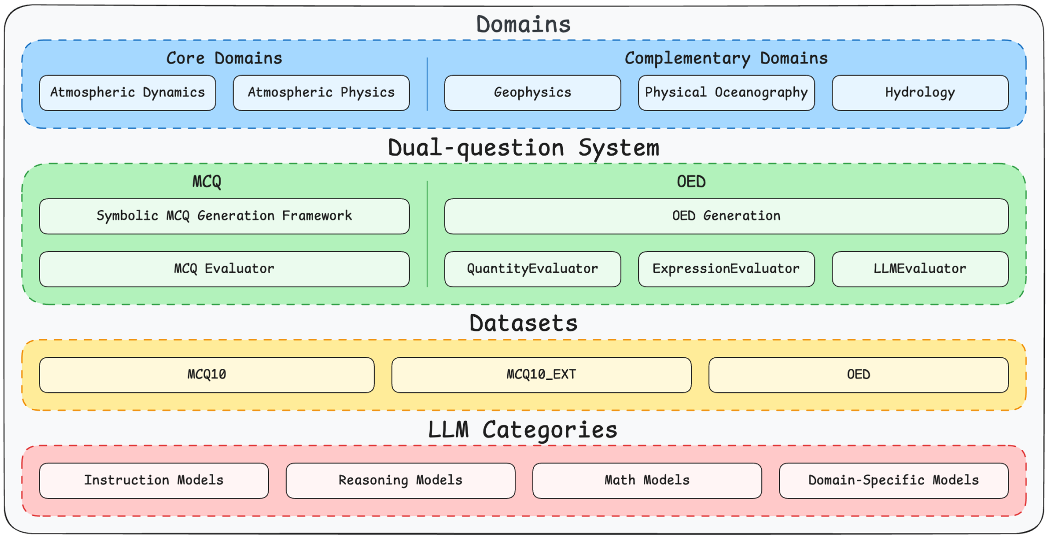


Figure 1: Overview of AtmosSci-Bench

To address this need, we introduce AtmosSci-Bench, a benchmark designed to comprehensively evaluate recent advances of LLMs in atmospheric science and related domains. An overview of AtmosSci-Bench is shown in Figure 1. The construction process—including the dual-question system, datasets, and domain coverage—is described in Section 3, while the evaluation and categories of LLMs assessed are detailed in Section 4. Concretely, we summarize our key contributions:

Contribution 1. We construct AtmosSci-Bench, a comprehensive benchmark comprising Multiple-Choice Questions (MCQs) and Open-Ended Questions (OEQs) to assess LLM performance across five scientific domains relevant to atmospheric science: (i) *atmospheric dynamics*, (ii) *atmospheric physics*, (iii) *geophysics*, (iv) *hydrology*, and (v) *physical oceanography*. MCQs enable scalable automated assessment via symbolic templates, while OEQs reveal deeper LLM reasoning ability in open-ended settings. The question set is curated from graduate-level materials and systematically expanded using a symbolic generation framework to ensure relevance and diversity.

Contribution 2. We conduct a comprehensive evaluation that includes a wide range of representative LLMs, which can be concretely categorized into four classes: (i) *instruction model*s that have been fine-tuned for instruction following; (ii) *reasoning models* that have been aligned with advanced reasoning abilities; (iii) *math models* that have been augmented with more mathematical skills; and (iv) *domain-specific climate models* that have been continuously pre-trained with climate-relevant corpus. We carefully analyze the evaluation results and summarize the following findings:

- ***Finding 1.*** *Reasoning models (such as GPT-o3-mini and Deepseek-R1) outperform instruction, math, and domain-specific models, demonstrating the superior significance of advanced reasoning ability in atmospheric science tasks.*
- ***Finding 2.*** *The inference time scaling introduces interesting quality-efficiency tradeoffs for reasoning models—model accuracy improves with longer reasoning token lengths up to a certain threshold, beyond which the gains plateau and diminishing returns emerge.*
- ***Finding 3.*** *Despite their superior reasoning capabilities, advanced models remain sensitive to symbolic perturbations—minor changes in variable values or structure can notably impact their accuracy, suggesting that they possibly rely on pattern matching rather than genuine reasoning.*

2 Related Work

LLM advances. LLMs, such as OPT [14], LLaMA [15], GPT [16], GEMINI [17], CLAUDE [18], and MIXTRAL[19], have demonstrated remarkable performance across a wide range of applications. While general-purpose LLMs exhibit strong adaptability, domain-specific models have also been developed to enhance performance in specialized fields. In the context of atmospheric science, climate-focused LLMs such as CLIMATEBERT [20], and CLIMATEGPT [4] are designed to address the unique challenges of climate modeling and analysis, which illustrates a promising paradigm different from traditional approaches that designing a specific model for some particular task [21, 22, 23, 24, 25]. More recently, reasoning models, including GPT-O1 [26], GEMINI-2.0-FLASH-THINKING [27], QWQ [28], and DEEPSEEK-R1 [29], have emerged, highlighting advancements in mathematical and scientific problem-solving. These models leverage sophisticated reasoning techniques, presenting exciting opportunities for tackling complex challenges in atmospheric science. Details of fundamental differences between reasoning and instruction-tuned models are provided in Appendix B.3.

LLM benchmarks. Assessing LLMs is crucial for ensuring their effectiveness in deployment across various domains [30]. Traditional benchmarks such as `GSM8K` [31] and `MATH` [12] have become less informative as state-of-the-art models achieve near-perfect scores, motivating the need for more challenging and discriminative benchmarks to evaluate reasoning capabilities. In response, several recent benchmarks have been developed to target more advanced scientific reasoning: `GPQA-Diamond` [32] focuses on expert-level science, `AIME2024` [33] targets advanced mathematical problem solving, and `SCIBENCH` [13] evaluates collegiate-level scientific reasoning. These benchmarks have been widely adopted for assessing reasoning-oriented LLMs. More recently, a complementary trend has emerged. `HUMANITY'S LAST EXAM` [34] provides broad subject coverage across the frontier of human knowledge and aims to serve as the final closed-ended academic benchmark of its kind. In contrast, domain-specific efforts such as `PHYSICS` [35], `PhysBench` [36], and `SciEx` [37] offer specialized evaluations in physics and scientific reasoning, incorporating symbolic derivations, open-ended problem solving, and expert-aligned scoring protocols. However, a comprehensive LLM benchmark tailored for atmospheric science remains underrepresented. While `CLIMAQA` [38] offers a promising first step, it primarily relies on definition-based assessments and lacks the depth required to evaluate complex numerical and symbolic reasoning essential to real-world atmospheric science problems. Designing a meaningful benchmark in this domain requires principled guidance to ensure robust, accurate, and interpretable evaluation. A notable methodological advancement is the use of symbolic extensions in benchmarking, as demonstrated by `GSM-Symbolic` [39], `VarBench` [40], and `MM-PhyQA`, where controlled variation of problem parameters improves robustness and mitigates contamination. These studies reveal that even small perturbations in problem structure can significantly degrade model performance, highlighting persistent fragilities in LLM reasoning and underscoring the need for rigorous benchmarks tailored to domain complexity.

3 Dataset and Question Construction

3.1 Dual-format question types.

ATMOSSCI-BENCH is designed with a dual-format question structure comprising MCQs and OEQs. These complementary formats enable us to systematically evaluate both the structured reasoning and expressive problem-solving capabilities of LLMs in atmospheric science.

Multiple-choice questions. The use of MCQs serves multiple core objectives: enabling automated scoring, supporting controlled symbolic perturbation, and ensuring unambiguous evaluation. Unlike traditional metrics such as exact match, BLEU, or F1 scores — which primarily assess surface-level similarity — MCQs offer fixed, well-defined answer choices. This significantly reduces ambiguity and enables a more precise assessment of logical inference and conceptual comprehension by the model [41]. To maximize the consistency and scalability of our benchmark, we constructed symbolic MCQ templates from source materials and applied rule-based perturbations to systematically explore model sensitivity to input transformations. The full construction pipeline, including template design, numerical instantiation, and distractor generation, is detailed in Section 3.3.

Open-ended questions. In parallel, we include open-ended questions to probe model reasoning in a less constrained setting. OEQs are better suited to reveal genuine reasoning capabilities, free from potential biases introduced by distractor. Prior work has noted that MCQs may allow models to bypass reasoning through answer recognition or pattern matching, rather than demonstrating full comprehension of theoretical concepts [34, 36, 42]. However, OEQs pose challenges in evaluation: the output space is unconstrained and lacks a single, universally accepted metric. Recent benchmarks adopt restrictive answer formats to simplify evaluation. For example, `PhysBench` [36] focuses solely on symbolic expressions, while `Humanity's Last Exam` [34] prohibits open-ended answers and instead relies on exact match or extraction-based evaluation. In contrast, we propose a cascade of evaluators specifically tailored for our OEQs. Our OEQs include a mix of quantitative answers involving numerical values with physical units, as well as symbolic expressions. To robustly evaluate this diverse answer space, our evaluators include a quantity evaluator, a symbolic expression evaluator, and an LLM-as-judge evaluator. We detail the full evaluation method in Section 4.2.

3.2 Data Source and Pre-processing

To ensure the rigor and relevance of the benchmark, we curated questions from course materials used in atmospheric science-related classes at our university. These sources provide high-quality, well-established content that aligns with the complexity and depth required for evaluating LLMs in this domain. The detailed design principles are provided in Appendix A. We leverage Mathpix OCR [43], a state-of-the-art OCR (Optical Character Recognition) engine, to extract both questions and their corresponding explanations from the collected materials. For multi-part problems or sequential questions where solving one step is necessary to proceed to the next, we consolidated them into single questions to enhance the complexity and depth of reasoning required. This approach preserves the logical progression of problem-solving, ensuring a comprehensive assessment of model capabilities.

Category distribution. The benchmark spans five scientific domains commonly addressed in atmospheric science, each chosen for its foundational importance in climate-related education and research. While only two of these — *atmospheric dynamics* and *atmospheric physics*—are formally categorized under atmospheric science, the remaining three — *geophysics*, *hydrology*, and *physical oceanography* — serve as necessary complementary domains. These fields are tightly coupled with atmospheric processes and are frequently integrated into both academic instruction and real-world research. Together, these five categories ensure broad topical coverage and reflect the inherently interdisciplinary nature of atmospheric science:

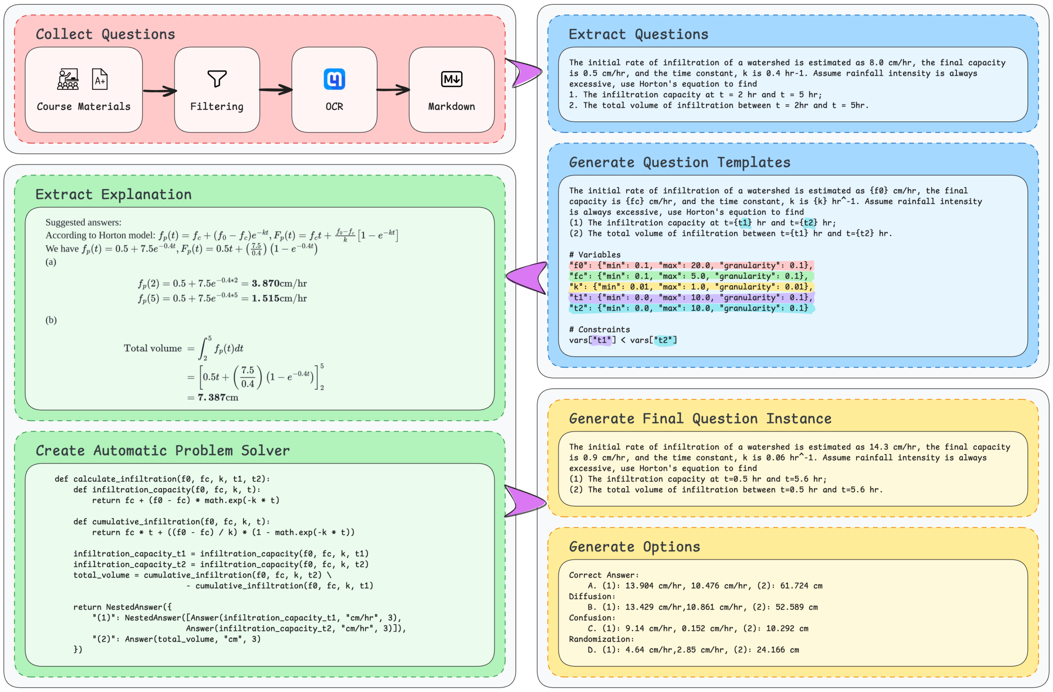
- **Atmospheric dynamics** focuses on the motion of the atmosphere, including large-scale weather systems, wind patterns, and governing forces of atmospheric circulation.
- **Atmospheric physics** covers physical processes such as radiation, thermodynamics, cloud formation, and energy transfer within the atmosphere.
- **Geophysics** encompasses the physical processes of the Earth, including its magnetic and gravitational fields, seismic activity, and internal structure.
- **Hydrology** examines the distribution, movement, and properties of water on Earth, including the water cycle, precipitation, rivers, lakes, and groundwater dynamics.
- **Physical oceanography** investigates the physical properties and dynamics of ocean water, including currents, waves, tides, and ocean-atmosphere interactions.

We summarize the distribution of questions across core scientific domains in Table 1, encompassing both MCQs and OEQs in our benchmark. The `MCQ10` subset is constructed by symbolically expanding each of our 67 curated question templates into 10 unique instances (Section 3.3), resulting in 670 scientifically grounded MCQs that enable large-scale, diverse evaluation. To complement this core set, we introduce an additional subset, `MCQ10_EXT`, consisting of 240 questions drawn from three complementary domains that are closely integrated with atmospheric science. For robustness evaluation, we also construct `MCQ30`, an augmented set incorporating higher levels of symbolic variation, for which 30 unique instances are generated for every question template (Section 5.3). To ensure consistency, `MCQ10_EXT` and `MCQ30` are generated following the same framework as `MCQ10`. In contrast, the `OEQ` dataset (391 questions) is individually curated to emphasize deep reasoning, without symbolic perturbations.

Table 1: Question type counts across different subfields in terms of MCQs and OEQs.

Type	**Atmos. Dynamics**	**Atmos. Physics**	**Geophysics**	**Hydrology**	**Phys. Oceanog.**	**Total**
`MCQ10`	370	140	70	50	40	**670**
`MCQ10_EXT`	0	0	10	170	60	**240**
`OEQ`	46	85	11	226	23	**391**

3.3 MCQ Generation Framework



```
The initial rate of infiltration of a watershed is estimated as 8.0 cm/hr, the final capacity
is 0.5 cm/hr, and the time constant, k is 0.4 hr-1. Assume rainfall intensity is always
excessive, use Horton's equation to find
1. The infiltration capacity at t = 2 hr and t = 5 hr;
2. The total volume of infiltration between t = 2hr and t = 5hr.
```

```
The initial rate of infiltration of a watershed is estimated as {f0} cm/hr, the final
capacity is {fc} cm/hr, and the time constant, k is {k} hr^-1. Assume rainfall intensity
is always excessive, use Horton's equation to find
(1) The infiltration capacity at t={t1} hr and t={t2} hr;
(2) The total volume of infiltration between t={t1} hr and t={t2} hr.

# Variables
"f0": {"min": 0.1, "max": 20.0, "granularity": 0.1},
"fc": {"min": 0.1, "max": 5.0, "granularity": 0.1},
"k": {"min": 0.01, "max": 1.0, "granularity": 0.01},
"t1": {"min": 0.0, "max": 10.0, "granularity": 0.1},
"t2": {"min": 0.0, "max": 10.0, "granularity": 0.1}

# Constraints
vars["t1"] < vars["t2"]
```

Suggested answers:
According to Horton model: $f_p(t) = f_c + (f_0 - f_c)e^{-kt}, F_p(t) = f_c t + \frac{f_0 - f_c}{k}\left[1 - e^{-kt}\right]$
We have $f_p(t) = 0.5 + 7.5e^{-0.4t}, F_p(t) = 0.5t + \left(\frac{7.5}{0.4}\right)\left(1 - e^{-0.4t}\right)$
(a)

$$f_p(2) = 0.5 + 7.5e^{-0.4*2} = \mathbf{3.870}\text{cm/hr}$$
$$f_p(5) = 0.5 + 7.5e^{-0.4*5} = \mathbf{1.515}\text{cm/hr}$$

(b)

$$\begin{aligned}\text{Total volume } &= \int_2^5 f_p(t)dt \\ &= \left[0.5t + \left(\frac{7.5}{0.4}\right)\left(1 - e^{-0.4t}\right)\right]_2^5 \\ &= \mathbf{7.387}\text{cm}\end{aligned}$$

```
def calculate_infiltration(f0, fc, k, t1, t2):
    def infiltration_capacity(f0, fc, k, t):
        return fc + (f0 - fc) * math.exp(-k * t)

    def cumulative_infiltration(f0, fc, k, t):
        return fc * t + ((f0 - fc) / k) * (1 - math.exp(-k * t))

    infiltration_capacity_t1 = infiltration_capacity(f0, fc, k, t1)
    infiltration_capacity_t2 = infiltration_capacity(f0, fc, k, t2)
    total_volume = cumulative_infiltration(f0, fc, k, t2) \
                      - cumulative_infiltration(f0, fc, k, t1)

    return NestedAnswer({
        "(1)": NestedAnswer([Answer(infiltration_capacity_t1, "cm/hr", 3),
                             Answer(infiltration_capacity_t2, "cm/hr", 3)]),
        "(2)": Answer(total_volume, "cm", 3)
    })
```

```
The initial rate of infiltration of a watershed is estimated as 14.3 cm/hr, the final capacity
is 0.9 cm/hr, and the time constant, k is 0.06 hr^-1. Assume rainfall intensity is always
excessive, use Horton's equation to find
(1) The infiltration capacity at t=0.5 hr and t=5.6 hr;
(2) The total volume of infiltration between t=0.5 hr and t=5.6 hr.
```

```
Correct Answer:
    A. (1): 13.904 cm/hr, 10.476 cm/hr, (2): 61.724 cm
Diffusion:
    B. (1): 13.429 cm/hr,10.861 cm/hr, (2): 52.589 cm
Confusion:
    C. (1): 9.14 cm/hr, 0.152 cm/hr, (2): 10.292 cm
Randomization:
    D. (1): 4.64 cm/hr,2.85 cm/hr, (2): 24.166 cm
```

Figure 2: Construction pipeline of our template-based question generation framework. Red block display the question collecting process. Blue blocks represent the question generation process (variables are highlighted in different colors). Greed blocks depict the automatic problem solver, which derives the answer from given variables. Yellow blocks illustrate an example of a generated question and its corresponding options.

To rigorously evaluate the reasoning and problem-solving capabilities of LLMs, we employ symbolic MCQ generation techniques inspired by the `GSM-Symbolic` framework [39], enhanced with a rule-based mechanism. This approach enables the creation of scalable and diverse question sets while ensuring logical coherence and alignment with real-world physical laws. Instead of fixed numerical values, we also design a template-based question perturbation mechanism with placeholder variables, which can be systematically instantiated through symbolic extensions. This ensures that models are tested on genuine reasoning ability rather than pattern matching from the potentially contaminated training data. Figure 2 illustrates the question construction pipeline as we enumerate below.

- **Question template construction**: We invite domain experts in atmospheric science to systematically transform selected questions (OCR extracted) into reusable templates. The experts manually identify numerical values within each question and replace them with variable placeholders, ensuring flexibility for symbolic instantiation. These variable placeholders, highlighted in different colors in Figure 2, allow for systematic variation while preserving the original scientific integrity of the problem.

- **Numerical assignment in question template**: We design a rule-based mechanism for valid numerical assignments in each question template. Note that many variables in atmospheric science problems are interdependent, meaning that the inappropriate assignment of some value(s) could lead to unrealistic or invalid physical scenarios. To fulfill this requirement, we ask the experts for each question template to define: (i) a valid numerical range (*min*, *max*) for each variable to ensure scientifically plausible values; (ii) a granularity parameter (i.e., the smallest step size between values) to control precision; and (iii) a set of rule-based constraints that are manually implemented to enforce logical dependencies (e.g., in Figure 2, ensuring $t_1 < t_2$). We believe these manual configurations ensure that all generated instances remain scientifically valid while allowing systematic variation in numerical representation.
- **Automatic problem solver to support value perturbation**: For each question, we utilize GPT-4o to generate an initial Python implementation based on the corresponding explanatory solution. This synthesized solution is then *manually reviewed, verified, and refined by experts to ensure correctness* and adherence to the intended problem-solving methodology. Once validated, the solver can automatically compute the correct answer for any given set of valid input variables, ensuring consistency and scalability in question generation. Note that to ensure consistency, accuracy, and alignment with real-world scientific standards, we also manually assign appropriate units and define significant digits for rounding the final answer in each automatic problem solver. This standardization maintains numerical precision while preventing inconsistencies in representation, ensuring that generated answers adhere to established atmospheric science conventions.
- **Option generation**: To effectively assess LLM reasoning, MCQs require plausible but incorrect distracting options that challenge the model's understanding while avoiding trivial elimination strategies [44]. To mitigate potential positional bias, we randomized the order of options during benchmark generation. We further verified that model accuracy was not skewed toward any specific answer position. The detailed generation mechanisms are provided in Appendix F.

4 Evaluation and Experimental Setup

4.1 Constrained Prompting for Evaluation

To ensure consistency in model outputs and simplify downstream answer extraction, we adopt *constrained prompting* across both MCQ and OEQ tasks. All prompts are designed to guide the model toward producing structured answers, with final results required to appear within LaTeX `\boxed{}` expressions. These outputs are subsequently extracted using regular expression (RegEx) parsing to support reliable and automatic evaluation. Full prompt templates are provided in Appendix E.1.

4.2 Evaluation Metrics

We design separate evaluation protocols for multiple-choice and open-ended questions, aligned with the structure of their respective answer formats.

MCQ Evaluation. For multiple-choice questions, we use the `MCQEvaluator`, which applies a straightforward accuracy-based metric. We extract the model's selected option from its response using a regular expression and compare it against the ground-truth label. A match is counted as correct; otherwise, it is marked incorrect. The final metric is the proportion of correctly matched answers across the evaluation set.

OEQ Evaluation. For open-ended questions, which can yield answers in numeric, symbolic, or natural language formats, we adopt a cascade of evaluators to improve coverage and accuracy:

- **QuantityEvaluator:** This is the primary evaluator for arithmetic-based questions that produce scalar answers with physical units. We use regular expressions to extract numeric values from model outputs and validate correctness within a *5% tolerance*, which reflects standard error bounds in Earth Science disciplines. Unit compatibility and conversion are handled using the `pint` Python library.
- **ExpressionEvaluator:** If the quantity-based check fails or the format is not purely numeric, we fall back to symbolic equivalence checking. We parse both model and reference expressions using the Python library `sympy` and simplify their difference. This approach is inspired by methodologies in recent scientific reasoning benchmarks such as *PHYSICS* [35].

- **LLMEvaluator:** If both automated evaluators fail, we invoke an GPT-4O-MINI [16] with structured outputs [45] to compare responses against the ground truth using rubric-based instructions (Similar to QuantityEvaluator, *5% tolerance* for numerical value are accepted). An example LLMEvaluator prompt is provided in Appendix E.2. Notably, LLM-based evaluation is increasingly accepted in scientific benchmarks. Recent studies have shown that LLMs can serve as effective graders, exhibiting strong agreement with expert human evaluations [37]. Several state-of-the-art benchmarks—including *SciEx* [37], *Humanity's Last Exam* [34], and *PHYSICS* [35]—have adopted LLM-as-Judge as a primary or even exclusive evaluation method.

If any of the above evaluators determine the model's output to be equivalent to the reference, the sub-question is marked as correct. The final score for an OEQ is computed as the average correctness across all its subparts, and the overall OEQ accuracy is the average across the dataset. While `QuantityEvaluator` and `ExpressionEvaluator` are highly accurate and well-grounded for positive true answers, they may still produce false negatives due to the evaluation in Python library `sympy`. Mitigating false negatives is one of the design intentions of our cascade of evaluators, which `LLMEvaluator` plays as a solid backup evaluator handling potential positive cases that `ExpressionEvaluator` marks as false, thereby reinforcing overall robustness. Further analysis of evaluator robustness, the human-LLM agreement study on `LLMEvaluator`, and false negatives examples occur in `ExpressionEvaluator` are provided in Appendix J.

4.3 Evaluation Questions

We design four main experiments to assess LLM performance on our benchmark, focusing on comprehensive performance comparison across model categories (*Q1*), the effect of inference-time reasoning length (*Q2*), and robustness to symbolic perturbation (*Q3*). We enumerate these concrete questions below:

- *Q1. How do various state-of-the-art LLMs (i.e., falling into different categories of instruction, math, reasoning, and domain-specific models) comprehensively perform for the proposed atmospheric science benchmark?*
- *Q2. How do the models specialized in reasoning perform during inference time scaling, i.e., how can we improve the model's test accuracy by increasing the length of reasoning tokens?*
- *Q3. How robust are the benchmark results, especially when we variate the degree of perturbation introduced by symbolic variation?*

4.4 Benchmark Models

To comprehensively assess LLM performance in atmospheric science, we evaluate a diverse set of state-of-the-art models spanning four categories: (i) instruction-tuned models, (ii) reasoning-optimized models, (iii) math-augmented models, and (iv) domain-specific models. This categorization facilitates a structured comparison between general-purpose, specialized, and domain-adapted approaches. A complete list of evaluated models is provided in Appendix B.1, and the corresponding cost and runtime statistics are reported in Appendix M.

5 Evaluation Results and Discussion

5.1 End-to-end Evaluation Results

Experimental setup. To comprehensively evaluate the performance of four categories of LLMs on atmospheric science tasks and assess whether ATMOSSCI-BENCH provides a sufficiently challenging and discriminative evaluation framework, we conduct a systematic performance comparison using our `MCQ10`, `MCQ10_EXT`, and `OED` benchmark across four representative LLM categories introduced in Section 4. We standardize experimental settings for each category as: (i) Reasoning models use 32K max context length, including the reasoning tokens; (ii) Instruction and math models use 8K max output tokens, balancing response quality and efficiency; (iii) Domain-specific models are set to 4K context length, the maximum capacity they support. By controlling these variables, we ensure that performance differences reflect genuine capability gaps rather than confounding factors, allowing us to validate whether ATMOSSCI-BENCH effectively differentiates model performance and highlights reasoning proficiency. Details of the hyperparameter settings and our computation resource are provided in Appendix D and Appendix G.

Results and analysis. For `MCQ10`, we present accuracy across different atmospheric science tasks, along with an overall performance comparison in Table 2 with three key observations:

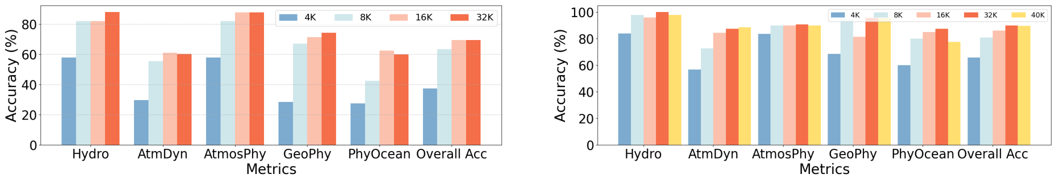
Table 2: Accuracy (%) and symbolic standard deviation (SymStd.) comparison across four LLM categories on the `MCQ10` dataset.

Category	**Model**	**Hydro**	**AtmDyn**	**AtmosPhy**	**GeoPhy**	**PhyOcean**	**Overall Acc**	**SymStd.**
Instruction Models	Gemma-2-9B-it	28.0	17.29	21.42	11.42	20.0	18.50	3.73
	Gemma-2-27B-it	56.0	31.08	47.14	41.42	40.0	37.91	4.62
	Qwen2.5-3B-Instruct	46.0	29.19	34.28	30.0	37.5	31.49	7.71
	Qwen2.5-7B-Instruct	60.00	38.11	50.71	51.43	32.50	43.43	4.90
	Qwen2.5-32B-Instruct	60.0	46.22	63.57	62.86	50.0	52.84	5.68
	Qwen2.5-72B-Instruct-Turbo	72.00	50.00	76.43	44.29	57.50	57.01	4.44
	Llama-3.3-70B-Instruct	82.0	42.66	66.43	51.52	42.5	51.51	3.92
	Llama-3.1-405B-Instruct-Turbo	70.00	48.11	64.29	57.14	52.50	54.33	5.81
	GPT-4o-mini	48.00	42.16	58.57	40.00	40.00	45.67	5.08
	GPT-4o	72.0	51.35	74.29	60.0	45.0	58.21	5.22
	Gemini-2.0-Flash-Exp	90.00	58.11	67.14	77.14	55.00	64.18	3.85
	Deepseek-V3	94.00	56.22	73.57	64.29	52.50	63.28	6.02
Reasoning Models	QwQ-32B-Preview	88.0	63.24	87.86	77.14	50.0	70.9	4.41
	Gemini-2.0-Flash-Thinking-Exp (01-21)	100.00	78.11	83.57	91.43	70.00	81.79	3.78
	GPT-o1	100.00	82.70	90.71	92.86	77.50	86.42	2.94
	Deepseek-R1	98.00	85.68	93.57	95.71	72.50	88.51	2.99
	Qwen3-235B-A22B-FP8-Throughput	98.0	86.49	93.57	90.0	70.0	88.21	3.75
	GPT-o3-mini	100.0	87.57	89.29	95.71	77.5	89.1	3.3
Math Models	Deepseek-Math-7B-RL	22.00	22.43	28.57	24.29	35.00	24.63	4.52
	Deepseek-Math-7B-Instruct	36.00	28.38	33.57	30.00	40.00	30.90	4.17
	Qwen2.5-Math-1.5B-Instruct	48.00	29.19	23.57	34.29	30.00	30.00	2.94
	Qwen2.5-Math-7B-Instruct	54.00	30.81	39.29	35.71	30.00	34.78	5.85
	Qwen2.5-Math-72B-Instruct	68.00	54.05	72.14	62.86	30.00	58.36	6.31
Domain-Specific Models	ClimateGPT-7B	26.00	18.65	22.86	11.43	32.50	20.15	5.18
	ClimateGPT-70B	24.00	25.41	28.57	40.00	22.50	27.31	4.45
	GeoGPT-Qwen2.5-72B	60.0	30.0	40.71	37.14	25.0	34.93	2.74

- ATMOSSCI-BENCH *effectively differentiates LLM performance across categories, with reasoning models demonstrating the highest proficiency.* The results confirm that our benchmark successfully distinguishes LLM performance, particularly in assessing reasoning proficiency. Reasoning models (70.9% - 89.1%) significantly outperform instruction models (18.5% - 64.18%), demonstrating superior consistency with lower symbolic reasoning standard deviation (SymStd) [39]. GPT-O3-MINI, the best-performing reasoning model, achieves 89.1% accuracy, while the top instruction model, GEMINI-2.0-FLASH-EXP, only reaches 64.18%, a substantial 24.92% gap. This clear performance variance underscores ATMOSSCI-BENCH 's ability to challenge advanced LLMs, ensuring that strong reasoning skills translate into measurable performance gains.

- *Math models do not show a clear advantage over instruction models.* Despite their specialization, math models do not significantly outperform instruction models, suggesting that mathematical optimization alone is insufficient for solving atmospheric science challenges.

- *Domain-specific models underperform despite climate specialization, indicating a need for strong reasoning-augmented approaches.* Domain-specific models show notably lower performance despite being trained on domain-related corpora. For instance, CLIMATEGPT-7B and CLIMATEGPT-70B achieve only 20.15% and 27.31% accuracy, respectively. This outcome is not entirely surprising, as the pretraining data for ClimateGPT is drawn almost entirely from crowd-sourced prompt-completion pairs and multi-turn dialogues, which are not effective for structured, step-wise reasoning found in scholarly problem solving. Interestingly, GEOGPT-QWEN2.5-72B also underperforms, achieving only 34.93% accuracy, whereas its base model Qwen-2.5-72B-Instruct-Turbo with the same model size achieves an accuracy of 57.01%. To better understand the flaws of domain-specific models, we conducted an error analysis, as detailed in Appendix I.2. These error patterns are consistent with our explanation of GeoGPT's. Indeed, GeoGPT's stated design goal: it is intended "to understand the demands of non-professional users, then think, plan, and execute defined GIS tools to produce effective results." Optimizing for tool planning rather than sustained step-by-step problem-solving data, therefore, provides a more plausible explanation for its lower accuracy than parameter size. Taken together, the evidence suggests that the principal limitations of current domain-specific LLMs lie in their training objectives and data composition: a focus on tool interaction or conversational style can leave gaps in both domain knowledge breadth and reasoning robustness. This underscores the need for reasoning-augmented domain models that combine scientific knowledge with structured inference skills. ATMOSSCI-BENCH provides a rigorous and interpretable evaluation framework to support the development and diagnosis of such models, addressing key limitations in current domain-specific approaches.



(a) QwQ-32B-Preview accuracy over 4K–32K tokens.

(b) GPT-o3-mini accuracy over 4K–40K tokens.

Figure 3: Reasoning step study. Accuracy (%) of different models across increasing input lengths.

The results for `OEQ` and the complementary `MCQ10_EXT` set are provided in Appendix H. `MCQ10_EXT`, designed to augment the original `MCQ10` set with additional domain coverage, exhibits performance trends consistent with those observed in `MCQ10`, supporting the robustness of our findings. In contrast, LLMs score below 40% on the `OEQ`, highlighting AtmosSci-Bench's capacity to evaluate deeper layers of logical and numerical reasoning, reinforcing the importance of inference-through-thinking as a critical dimension in assessing LLM capabilities for scientific problem-solving.

In conclusion, to answer *Q1* regarding the overall performance of various LLM categories, our evaluation reveals that *reasoning models significantly outperform instruction, math, and domain-specific models in atmospheric science tasks, highlighting their superior adaptability to advanced reasoning challenges, while domain-specific models struggle despite specialized training.*

5.2 Inference Time Scaling for Reasoning Models

Experimental setup. To answer *Q2*, i.e., whether increasing the length of reasoning tokens improves the performance of reasoning models, we conduct an inference time scaling evaluation on `MCQ10` using the QwQ-32B-Preview and GPT-o3-mini model, varying its reasoning token limits from 4K up to 40K. By systematically increasing the token limit, we aim to determine whether a longer inference process leads to higher accuracy and whether there exists an optimal threshold beyond which additional tokens provide minimal benefit.

Results and analysis. As shown in Figure 3a, increasing the reasoning token limit for QwQ-32B-Preview generally improves model accuracy, but the gains diminish beyond a certain threshold. The performance in terms of overall accuracy is consistently lower at 4K tokens, improves significantly at 8K and 16K tokens, and then plateaus beyond 16K tokens, with 32K tokens offering only marginal improvement. GPT-o3-mini in 3b show a similar pattern except plateaus around 32K. This trend suggests that while extending reasoning length enhances model performance up to a certain point, it further increases yield, diminishing returns without proportional accuracy gains. Thus, our answer to *Q2* is that *increasing the length of reasoning tokens improves model accuracy up to 16K tokens, beyond which performance gains diminish, indicating an optimal threshold for inference time scaling.*

5.3 Robustness of AtmosSci-Bench

To evaluate the robustness of AtmosSci-Bench (*Q3*), we conduct a experiment to assess - robustness to different degrees of perturbation introduced by symbolic variation.

Experimental setup. Inspired by `GSM-Symbolic` [39], which demonstrates that modifying numerical variables in the `GSM8K` dataset leads to significant performance drops, suggesting that LLMs may rely on pattern matching rather than genuine logical reasoning. Here, pattern matching refers to reliance on patterns memorized during training—such as recalling similar questions or solutions—rather than reasoning through the problem independently. We aim to assess the robustness of advanced reasoning models under varying degrees of symbolic perturbation. To examine this, we evaluate three reasoning models—DeepSeek-R1, Gemini-2.0-Flash-Thinking-Exp (01-21), GPT-o3-mini, and QwQ-32B-Preview—on `MCQ30`, which consists of 30 test sets for each question template, with controlled symbolic variations to analyze sensitivity to numerical perturbations. We systematically modify numerical variables within a scientifically reasonable range, introducing controlled variations to assess whether performance remains stable or degrades significantly with perturbation.

Results and analysis. Figure 4 illustrates the empirical performance distribution of reasoning models on `MCQ30`. We observe that the accuracy of the original question set for all models (dashed line in Figure 4) is approximately one standard deviation away from the mean accuracy across perturbed instances, except QwQ-32B-Preview exhibits about two standard deviations. A hypothesis test was also conducted (Appendix N), which indicates that the null hypothesis of "random chance" cannot be conclusively rejected, suggesting the possibility of potential data contamination. Although not definitive evidence of contamination, such deviations are nontrivial and warrant further attention, implying that reasoning LLMs are possibly pattern-matching from contaminated data. To answer <u>*Q3*</u> w.r.t symbolic variation, the results indicate that *the reasoning models evaluated in our benchmark could still be under the risk of insufficient robustness under symbolic perturbation, as increasing the degree of variation leads to notable and often unpredictable drops in accuracy, suggesting that they possibly rely on pattern matching instead of genuine reasoning.*

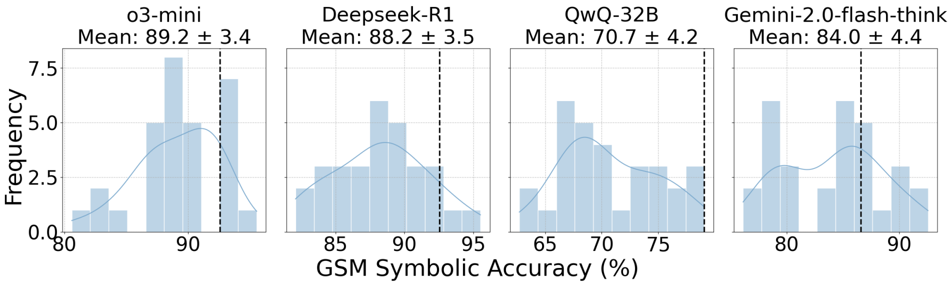


Figure 4: Performance distribution among reasoning LLMs on `MCQ30`. The Y-axis represents the frequency of the symbolic test sets achieving the accuracy shown on the X-axis. The black vertical dash lines denote the accuracy of the original question set.

5.4 Further Discussion

To better understand the skill-level limitations of current LLMs in atmospheric science, as well as common error patterns and evaluation constraints, we refer readers to Appendix K (skill-oriented ablation analysis including a retrieval-augmented generation (RAG) baseline), Appendix I.1 (error categorization), Appendix N (data contamination), and Appendix O (discussion of limitations).

6 Conclusion

In this paper, we introduced AtmosSci-Bench, a novel benchmark designed to systematically evaluate the reasoning and problem-solving capabilities of LLMs in atmospheric science. Our findings highlight that reasoning models outperform other categories, demonstrating stronger problem-solving and reasoning capabilities in the domain of atmospheric science. This also underscores the benchmark's effectiveness in differentiating models. Our benchmark covers five core categories — hydrology, atmospheric dynamics, atmospheric physics, geophysics, and physical oceanography — through a dual-format question design comprising both MCQs and OEQs. This structure enables both scalable automated evaluation and deeper probing of scientific reasoning skills. AtmosSci-Bench employs a scalable, template-based generation framework for MCQs to ensure diversity and control over symbolic perturbations, while OEQs are curated to assess deeper reasoning skills without predefined answer choices. Through a comprehensive evaluation across four model categories — instruction-tuned models, advanced reasoning models, math-augmented models, and domain-specific climate models — we provide key insights into the strengths and limitations of current LLMs. Our findings highlight that reasoning models outperform other categories, demonstrating stronger problem-solving and reasoning capabilities in the domain of atmospheric science. This also underscores the benchmark's effectiveness in differentiating models. We believe that AtmosSci-Bench (where all the implementations are fully open-sourced) can serve as an essential step toward advancing the application of LLMs in climate-related decision-making by offering a standardized and rigorous evaluation framework for future research.

Acknowledgments and Disclosure of Funding

This research is supported by the Hong Kong Research Grants Council's collaborative research funds (project nos. C6032-21G, C5004-23G, C5085-24G), general research fund (project no. 16300424), theme-based research scheme (project no. T22-501/23R, T22-607/24-N) and NSFC/RGC Collaborative Fund (project no. CRS_PolyU503/23). All co-authors acknowledge the support by the Otto Poon Centre for Climate Resilience and Sustainability at HKUST. This work is part of the United Nations Educational Scientific and Cultural Organization (UNESCO) International Decade of Sciences for Sustainable Development (2024–33) and contributes to the Seamless Prediction and Services for Sustainable Natural and Built Environment (SEPRESS) Program (2025–32), an initiative endorsed under this global framework.

References

[1] T Brown, B Mann, N Ryder, M Subbiah, JD Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, et al. Language models are few-shot learners advances in neural information processing systems 33. 2020.

[2] Vincent Nguyen, Sarvnaz Karimi, Willow Hallgren, Ashley Harkin, and Mahesh Prakash. My climate advisor: An application of NLP in climate adaptation for agriculture. In Dominik Stammbach, Jingwei Ni, Tobias Schimanski, Kalyan Dutia, Alok Singh, Julia Bingler, Christophe Christiaen, Neetu Kushwaha, Veruska Muccione, Saeid A. Vaghefi, and Markus Leippold, editors, *Proceedings of the 1st Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2024)*, pages 27–45, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[3] Lujia Zhang, Hanzhe Cui, Yurong Song, Chenyue Li, Binhang Yuan, and Mengqian Lu. On the opportunities of (re)-exploring atmospheric science by foundation models: A case study. *arXiv preprint arXiv:2407.17842*, 2024.

[4] David Thulke, Yingbo Gao, Petrus Pelser, Rein Brune, Rricha Jalota, Floris Fok, Michael Ramos, Ian van Wyk, Abdallah Nasir, Hayden Goldstein, et al. Climategpt: Towards ai synthesizing interdisciplinary research on climate change. *arXiv preprint arXiv:2401.09646*, 2024.

[5] Charles Cao, Jie Zhuang, and Qiang He. LLM-assisted modeling and simulations for public sector decision-making: Bridging climate data and policy insights. In *AAAI-2024 Workshop on Public Sector LLMs: Algorithmic and Sociotechnical Design*, 2024.

[6] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

[7] Ziming Luo, Zonglin Yang, Zexin Xu, Wei Yang, and Xinya Du. Llm4sr: A survey on large language models for scientific research. *arXiv preprint arXiv:2501.04306*, 2025.

[8] Qiguang Chen, Mingda Yang, Libo Qin, Jinhao Liu, Zheng Yan, Jiannan Guan, Dengyun Peng, Yiyan Ji, Hanjing Li, Mengkang Hu, et al. Ai4research: A survey of artificial intelligence for scientific research. *arXiv preprint arXiv:2507.01903*, 2025.

[9] Tianshi Zheng, Zheye Deng, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Zihao Wang, and Yangqiu Song. From automation to autonomy: A survey on large language models in scientific discovery. *arXiv preprint arXiv:2505.13259*, 2025.

[10] Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, et al. From ai for science to agentic science: A survey on autonomous scientific discovery. *arXiv preprint arXiv:2508.14111*, 2025.

[11] Chandan K Reddy and Parshin Shojaee. Towards scientific discovery with generative ai: Progress, opportunities, and challenges. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28601–28609, 2025.

[12] 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.

[13] Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.

[14] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

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

[16] OpenAI. Openai gpt-4o, 2024.

[17] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024.

[18] Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024.

[19] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[20] Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. Climatebert: A pretrained language model for climate-related text. *arXiv preprint arXiv:2110.12010*, 2021.

[21] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022.

[22] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[23] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. *arXiv preprint arXiv:2211.02556*, 2022.

[24] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, et al. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023.

[25] Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. Fuxi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate and Atmospheric Science*, 6(1):190, 2023.

[26] OpenAI. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms/`. Accessed: 2025-01-28.

[27] Google DeepMind. Gemini 2.0 flash thinking experimental. `https://deepmind.google/technologies/gemini/flash-thinking/`. Accessed: 2025-01-28.

[28] Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024.

[29] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[30] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[31] 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.

[32] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[33] MAA. Maa invitational competitions. Mathematical Association of America, February 2024.

[34] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[35] Kaiyue Feng, Yilun Zhao, Yixin Liu, Tianyu Yang, Chen Zhao, John Sous, and Arman Cohan. Physics: Benchmarking foundation models on university-level physics problem solving. *arXiv preprint arXiv:2503.21821*, 2025.

[36] Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. *arXiv preprint arXiv:2501.16411*, 2025.

[37] Tu Dinh, Carlos Mullov, Leonard Bärmann, Zhaolin Li, Danni Liu, Simon Reiß, Jueun Lee, Nathan Lerzer, Jianfeng Gao, Fabian Peller-Konrad, et al. Sciex: Benchmarking large language models on scientific exams with human expert grading and automatic grading. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11592–11610, 2024.

[38] Veeramakali Vignesh Manivannan, Yasaman Jafari, Srikar Eranky, Spencer Ho, Rose Yu, Duncan Watson-Parris, Yian Ma, Leon Bergen, and Taylor Berg-Kirkpatrick. Climaqa: An automated evaluation framework for climate foundation models. *arXiv preprint arXiv:2410.16701*, 2024.

[39] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*, 2024.

[40] Kun Qian, Shunji Wan, Claudia Tang, Youzhi Wang, Xuanming Zhang, Maximillian Chen, and Zhou Yu. Varbench: Robust language model benchmarking through dynamic variable perturbation. *arXiv preprint arXiv:2406.17681*, 2024.

[41] Nishant Balepur and Rachel Rudinger. Is your large language model knowledgeable or a choices-only cheater? In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 15–26, 2024.

[42] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2024.

[43] Mathpix. Mathpix: Ai-powered ocr for math and science, 2024. Accessed: 2024-01-29.

[44] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2024.

[45] OpenAI. Structured output guide. `https://platform.openai.com/docs/guides/structured-outputs?api-mode=chat`, 2024. Accessed: 2025-05-13.

[46] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[47] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie

Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024.

[48] Google DeepMind. Gemini 2.0 flash experimental. `https://deepmind.google/technologies/gemini/flash/`. Accessed: 2025-01-28.

[49] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey,

Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[50] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha

Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2024.

[51] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[52] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[53] OpenAI. Openai o3-mini. `https://openai.com/index/openai-o3-mini/`, 2025. Accessed: 2025-05-12.

[54] Qwen Team. Qwen3, April 2025.

[55] GeoGPT Team. Geogpt: Large language models for geoscience research, 2025. `https://huggingface.co/GeoGPT-Research-Project`.

[56] Google DeepMind. Gemini thinking. `https://ai.google.dev/gemini-api/docs/thinking`. Google, 2024. Accessed: 2025-07-30.

[57] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging ai applications, 2018.

[58] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[59] HuggingFace. Accelerate: A simple way to train and use pytorch models with mixed precision, multi-gpu, and distributed setups. `https://github.com/huggingface/accelerate`, 2021.

[60] Eli Schwartz, Leshem Choshen, Joseph Shtok, Sivan Doveh, Leonid Karlinsky, and Assaf Arbelle. Numerologic: Number encoding for enhanced llms' numerical reasoning. *arXiv preprint arXiv:2404.00459*, 2024.

[61] Haotong Yang, Yi Hu, Shijia Kang, Zhouchen Lin, and Muhan Zhang. Number cookbook: Number understanding of language models and how to improve it. *arXiv preprint arXiv:2411.03766*, 2024.

[62] Aaditya K Singh and DJ Strouse. Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *arXiv preprint arXiv:2402.14903*, 2024.

[63] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

[64] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*, 2024.

[65] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[66] Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. *arXiv preprint arXiv:2504.18243*, 2025.

A Questions design principles.

To ensure a rigorous evaluation of LLMs in atmospheric science, we adhere to a set of well-defined principles that emphasize reasoning and interpretative abilities:

Deep understanding of essential physical equations: Atmospheric science is governed by fundamental physical equations, and a meaningful evaluation requires that LLMs not only recall these principles but also apply them appropriately in the corresponding contexts. Thus, the questions should be designed to assess both conceptual comprehension and the ability to use these equations in problem-solving, ensuring the benchmark measures true scientific reasoning rather than mere memorization.

Complex reasoning and multi-step logic: Many real-world atmospheric problems require synthesizing information from multiple sources, integrating equations, and applying multi-step logical reasoning. To reflect these challenges, benchmark questions should be crafted to go beyond simple recall, testing the model's ability to handle intricate reasoning and dynamic problem-solving scenarios inherent to the field.

Appropriate numerical arithmetic processing: Accurate numerical computation is essential for scientific disciplines, where correct reasoning leads to fixed, verifiable answers. By incorporating numerical problems, we provide a structured and objective evaluation framework, eliminating ambiguities in assessment. This approach also enables seamless integration of reasoning tasks, extending the benchmark's scope to evaluate mathematical intuition and computational fluency.

B Model Usage

B.1 Model List

Instruction models. Instruction-tuned models serve as strong general-purpose baselines, optimized for following prompts and single-step inference tasks, where we include:

- GPT-4O, GPT-4O-MINI [16]: OpenAI's instruction-tuned models.
- QWEN2.5-INSTRUCT (3B, 7B, 32B, 72B) [46]: Instruction-tuned Qwen models with enhanced abilities.
- GEMMA-2-9B-IT, GEMMA-2-27B-it [47]: Google's open-weight instruction models; along with Gemini-2.0-Flash-Exp [48], the powerful Gemini model optimized for efficiency.
- LLAMA-3.3-70B-INSTRUCT, LLAMA-3.1-405B-INSTRUCT-TURBO [49]: Meta's widely used instruction models.
- DEEPSEEK-V3 [50]: Deepseek's latest MoE-based instruction model for general tasks.

Math models. Mathematical LLMs specialize in problem-solving, computational reasoning, and theorem proving — such ability is essential for atmospheric problems. Towards this end, we include:

- DEEPSEEK-MATH-7B-INSTRUCT and DEEPSEEK-MATH-7B-RL [51]: Deepseek's math-focused models trained for theorem proving.
- QWEN2.5-MATH (1.5B, 7B, 72B) [52]: Qwen's recent models optimized for mathematics.

Reasoning models. Reasoning ability is the core technique to improve LLMs' performance over complicated tasks. We include the recent advanced reasoning models focus on deep logical reasoning and multi-step problem-solving:

- GPT-O1 [26], GPT-O3-MINI [53]: OpenAI's reasoning-optimized model.
- QWQ-32B-PREVIEW [28]: Reasoning model based on Qwen2.5-32B.
- GEMINI-2.0-FLASH-THINKING-EXP (01-21) [27]: Extended Gemini-2.0-Flash-Exp for enhanced reasoning.
- DEEPSEEK-R1 [29]: Deepseek's RL-trained model for complex problem-solving.
- QWEN3-235B-A22B-FP8-THROUGHPUT [54]: The Latest generation in the Qwen series equipped with thinking mode to enhance its reasoning capabilities.

Domain-specific models. We also include some models that are specially tailored for climate-related and atmospheric science tasks by supervised fine-tuning or continuous pre-training:

- CLIMATEGPT-7B, CLIMATEGPT-70B [4]: QA models specialized in the climate domain.
- GEOGPT-QWEN2.5-72B [55]: A domain-adapted model built on Qwen2.5-72B supports advanced reasoning and knowledge synthesis in specialized areas of geoscience.

B.2 Model and Library Usage Licenses

We list in Table 3 the models and software assets used in this work, along with their respective sources and licensing terms. All API-based models are accessed through official platforms under standard usage policies, while open-source models are released under community-accepted licenses (Apache, MIT, etc.).

Model / Library	**Source / Access Method**	**License / Terms of Use**
GPT-4o, GPT-4o-mini, GPT-o1, GPT-o3-mini	OpenAI API	OpenAI API Terms of Use
Gemini-2.0-Flash-Exp, Gemini-2.0-Flash-Thinking-Exp (01-21)	Google API (Vertex AI)	Google Cloud Terms of Service
DeepSeek-V3, DeepSeek-R1, DeepSeek-Math-7B-Instruct, DeepSeek-Math-7B-RL	Deepseek API	Deepseek Public API Terms
Qwen2.5-Instruct (3B, 7B, 32B, 72B)	HuggingFace / Together AI	Apache License 2.0
Qwen2.5-Math (1.5B, 7B, 72B)	HuggingFace / Together AI	Apache License 2.0
QwQ-32B-Preview	HuggingFace (based on Qwen2.5-32B)	Apache License 2.0
Qwen3-235B-A22B-FP8-Throughput	HuggingFace	Apache License 2.0
Gemma-2-9B-it, Gemma-2-27B-it	HuggingFace (Google)	CC BY-NC 4.0 / Google Research Terms
Llama-3.3-70B-Instruct, Llama-3.1-405B-Instruct-Turbo	HuggingFace (Meta AI)	Meta Llama 3 Community License Agreement
ClimateGPT, GeoGPT	HuggingFace	License provided in original repo (research only)
HuggingFace Transformers	`https://github.com/huggingface/transformers`	Apache License 2.0
Accelerate	`https://github.com/huggingface/accelerate`	Apache License 2.0
Ray	`https://github.com/ray-project/ray`	Apache License 2.0
NumPy, SciPy, Pandas	PyPI / open-source	BSD / MIT Licenses

Table 3: Sources and license information for models and libraries used in this work.

B.3 Fundamental Differences Between Reasoning and Instruction-Tuned Models

We provide a deeper explanation from a technical and architectural perspective regarding the **fundamental differences** that separate "reasoning models" (e.g., GPT-o1) from "instruction-tuned models" (e.g., GPT-4o):

1. **Deliberate "Thinking" Stage + Extra Inference Compute:** Reasoning models insert a dedicated "`<think>`" phase and allocate additional compute at inference time, enabling long chain-of-thought processing that allows smaller reasoning models to outperform much larger instruction-tuned ones on complex tasks [26, 56, 29].
2. **Self-Checking to Limit Error Propagation:** Through reinforcement learning, models develop emergent behaviors—self-verification, reflection, and back-tracking—that catch and correct faulty reasoning steps before the final answer [26, 29].
3. **Specialized RL Post-Training for Reasoning:** Models like DeepSeek-R1 rely on pure RL fine-tuning with *Group Relative Policy Optimization (GRPO)* on step-by-step solution data, explicitly rewarding correct intermediate reasoning and strengthening overall chain-of-thought quality [29, 28].

For these reasons, reasoning models show significant improvements in complex **multi-step** reasoning tasks. As shown on line 88, all reasoning models achieve substantial gains on benchmarks tailored for such tasks [26, 56, 28].

AtmosSci-Bench is specifically designed to evaluate whether LLMs equipped with thinking stages, intermediate learning, and self-correction can effectively solve multi-step problems. The results confirm that AtmosSci-Bench successfully distinguishes reasoning models from instruction-tuned models in atmospheric science, thus demonstrating its ability to assess reasoning capacity while posing meaningful challenges for future LLM development.

C Data Source and Usage Statement

The benchmark dataset introduced in this paper was independently constructed by the authors. All questions and materials were derived and reformulated from our available university-level content in atmospheric science-related courses. These materials include lecture notes, problem sets, and instructional examples used for teaching at our institution.

No proprietary, copyrighted, or scraped content was included in the dataset. The resulting benchmark is intended solely for academic research and educational use. We confirm that the dataset does not contain any personal information, and sensitive data that may negatively impact society.

To the best of our knowledge, this benchmark complies with relevant institutional and academic usage policies, and poses no legal or ethical risk for public release.

D Hyperparameters

To ensure fair comparison and consistent evaluation, we standardize the inference-time hyperparameters across all models in accordance with their capabilities and design constraints.

For **reasoning-optimized models**, we use a maximum context length of **32K tokens**. This decision is motivated by the fact that DeepSeek-R1 has a fixed 32K context window that cannot be modified. To maintain fairness, we adopt the same 32K limit for all reasoning models, including GPT-o1, QwQ-32B-Preview, and Gemini-2.0-Flash-Thinking-Exp (01-21). This configuration provides sufficient space for long-form reasoning and multi-step inference, ensuring that reasoning performance is not artificially constrained by token limits.

Additionally, most reasoning models—such as GPT-o1 and DeepSeek-R1—do not support customized decoding parameters like `temperature`, `top_p`, or `repetition_penalty`. Therefore, we use default hyperparameters for all models across all categories to ensure evaluation consistency and reproducibility.

For **instruction-tuned**, **math-augmented**, and **domain-specific models**, we set the maximum token limit to **8K**, which provides ample context for solving our benchmark tasks given the typical response lengths of these models.

E Prompt

E.1 LLM Prompt Template

We include below the full prompt templates used for both MCQ and OEQ questions. These were passed directly to the language models to elicit structured, parseable responses.

MCQ prompting. For multiple-choice questions, prompts are structured to elicit step-by-step reasoning followed by a clearly formatted answer selection. The model is instructed to return the final choice using the exact format `\boxed{A/B/C/D}`, ensuring compatibility with our extraction script.

MCQ Prompt Template

```
You are an Earth Science Expert answering multiple-choice questions.
Here is the question:  {question}
Here are the options:
{options_str}

Instructions:
1. Carefully analyze the question and options provided.
2. Please think step by step.  Use logical reasoning and critical thinking to
generate a detailed explanation or steps leading to the answer.
3. At the end of your response, ensure to provide the correct option
(A/B/C/D) on a new line in the following format strictly:
**Final Answer**:  \[ \boxed{{A/B/C/D}} \]
```

OEQ prompting. For open-ended questions, prompts instruct the model to produce a full derivation or explanation, concluding with boxed answers. When questions include multiple subparts (e.g., a), b)), each should be addressed in order, with the corresponding boxed result. This constrained prompting strategy enhances interpretability, ensures evaluation robustness, and minimizes ambiguity in final output formatting.

OEQ Prompt Template

```
You are an expert in Earth System Science.  Think step by step using logical
reasoning and scientific principles.
Provide a detailed explanation or derivation leading to your answer.
If the question includes subparts (e.g., a), b)), address each subpart
sequentially.
Conclude each subpart with its final result formatted as a LaTeX expression,
using:
a) \boxed{...}
b) \boxed{...}
For single-part questions, conclude with a single \boxed{final_answer}.
```

E.2 LLM Evaluator Prompt

OpenAI provides support for structured outputs [45], enabling model responses to adhere to predefined JSON schemas. Following this approach, we define the following `AnswerResponse` schema using `pydantic` to strictly constrain the LLM's output format:

AnswerResponse Schema (Pydantic)

```
from pydantic import BaseModel
class AnswerResponse(BaseModel):
  is_correct:  bool
  explanation:  str
```

The prompt below is used to guide the LLM evaluator to assess answer correctness based on mathematical, physical, and conceptual equivalence:

LLMEvaluator Prompt

```
You are an expert physics teacher evaluating student answers.
Compare the following two answers and determine if they are equivalent.

Consider the following in your evaluation:
1. Mathematical equivalence (e.g., 2π = 6.28)
2. Physical unit equivalence (e.g., 1 m/s = 3.6 km/h)
3. Conceptual equivalence (e.g., F = ma and a = F/m)
4. Numerical tolerance:  Allow a tolerance of {self.tolerance * 100}% for
numerical values.
  (e.g., if the expected value is 10, values between {10 - 10 *
self.tolerance} and {10 + 10 * self.tolerance} are acceptable.)

Respond with is_correct (true/false) and an explanation.
Expected answer (in LaTeX): {expected}
Student answer (in LaTeX): {actual}
```

F Incorrect option generation for MCQ Generation

We design the following mechanisms to generate incorrect options: (i) Diffusion: producing an incorrect answer by randomly swapping two variables in the computation; (ii) Confusion: altering a single variable in the equation to generate a close but incorrect result; (iii) Randomization: randomly assigning all variables within their predefined constraints, ensuring adherence to the rule-based mechanism; and (iv) Default: if above three methods fail to generate valid incorrect options (i.e., those satisfying the scientific constraints of the rule-based mechanism), we use a default strategy, where incorrect options are generated as scaled multiples of the correct answer (e.g., $\times 2$, $\times 3$, $\times 4$).

G Experimental Compute Resources

We categorize LLM inference into two groups based on deployment method: **API-based** and **local-based**.

- **API-based**: Models in this category are hosted by providers such as OpenAI, Google, Deepseek, and TogetherAI. We access these models via public inference APIs. To accelerate large-scale evaluation, we utilize parallel execution using the `Ray` Python library [57], which enables concurrent API requests. Total inference time varies depending on the model and the infrastructure provider's throughput.
- **Local-based**: These models are available through HuggingFace and executed locally using the HuggingFace `transformers` library [58], with acceleration enabled via `Accelerate` [59]. We run evaluations on two hardware setups: (1) a single machine with 8×NVIDIA RTX 4090 GPUs, and (2) two nodes (run separately) each equipped with 4×NVIDIA A800 GPUs. For a 70B non-reasoning model, a full evaluation run requires approximately 90 hours with a batch size of 4. In contrast, a 7B model can be evaluated in about 6 hours using a batch size of 64.

H Additional End-to-End Results

For `OEQ` result as shown in Table 4, instruction-tuned models generally achieve accuracies around 20%, whereas reasoning-optimized models consistently reach 30% or higher. This performance gap reflects the increased difficulty and reasoning demand of OEQs compared to MCQs, and further demonstrates AtmosSci-Bench's ability to probe deeper levels of logical and numerical reasoning. Notably, models with explicit reasoning alignment—such as GPT-o1, Deepseek-R1, and Gemini-2.0-Flash-Thinking-Exp—outperform their instruction-only counterparts (GPT-4o, Deepseek-V3, and Gemini-2.0-Flash-Exp), reinforcing the importance of inference with thinking.

Table 4: Accuracy (%) comparison across four LLM categories on the `OEQ` dataset.

Category	**Model**	**Hydro**	**AtmDyn**	**AtmosPhy**	**GeoPhy**	**PhyOcean**	**Overall Acc**
Instruction Models	Qwen2.5-72B-Instruct-Turbo	18.58	30.43	37.65	18.18	17.39	24.04
	Llama-3.3-70B-Instruct	14.16	19.57	28.24	9.09	13.04	17.65
	Llama-3.1-405B-Instruct-Turbo	9.29	17.39	21.18	18.18	13.04	13.30
	GPT-4o-mini	11.91	06.98	20.39	33.33	20.59	13.57
	GPT-4o	16.81	23.91	37.65	18.18	21.74	22.51
	Gemini-2.0-Flash-Exp	20.80	30.43	49.41	18.18	17.39	27.88
	Deepseek-V3	22.12	21.74	40.00	9.09	13.04	25.06
Reasoning Models	QwQ-32B-Preview	21.24	21.74	40.00	9.09	13.04	24.55
	Gemini-2.0-Flash-Thinking-Exp (01-21)	27.88	32.61	44.71	18.18	21.74	31.46
	GPT-o1	28.32	28.26	45.88	27.27	2.174	31.71
	Deepseek-R1	34.96	26.09	52.94	27.27	17.39	36.57
	Qwen3-235B-A22B-FP8-Throughput	28.32	30.43	54.12	0	21.74	32.99
	GPT-o3-mini	30.97	32.61	47.06	18.18	21.74	33.76
Domain-Specific Models	GeoGPT-Qwen2.5-72B	10.62	15.22	27.06	9.09	17.39	15.09

Table 5 presents the results on the `MCQ10_EXT` dataset, which serves as a complementary extension to `MCQ10` by expanding coverage across additional domains such as hydrology and physical oceanography. The performance trends observed in `MCQ10_EXT` closely mirror those of the original `MCQ10` dataset: reasoning-optimized models consistently outperform instruction-tuned and domain-specific models, highlighting their superior adaptability to scientific problem-solving. This consistency further confirms the robustness of AtmosSci-Bench in differentiating model capabilities across both core and extended scientific domains.

I Error Analysis

I.1 General Error Analysis

To better understand the types of reasoning failures made by LLMs in scientific contexts, we categorize common errors into four types:

- **Lack of Relevant Knowledge** — The model lacks the domain-specific background or factual recall to address the question.
- **Incorrect Calculation** — The model applies the right equations or methods, but performs numerical steps incorrectly.

Table 5: Accuracy (%) comparison across four LLM categories on the `MCQ10_EXT` dataset.

Category	Model	Hydro	AtmDyn	AtmosPhy	GeoPhy	PhyOcean	Overall Acc
Instruction Models	Gemma-2-9B-it	34.71	-	-	20.0	40.0	35.42
	Qwen2.5-7B-Instruct	64.71	-	-	60.0	51.67	61.25
	Qwen2.5-72B-Instruct-Turbo	82.94	-	-	50.0	63.33	76.67
	Llama-3.1-405B-Instruct-Turbo	76.47	-	-	30.0	61.67	70.83
	GPT-4o-mini	63.53	-	-	30.0	58.33	60.83
	GPT-4o	78.82	-	-	40.0	66.67	74.17
	Gemini-2.0-Flash-Exp	90.0	-	-	70.0	70.0	84.17
	Deepseek-V3	92.35	-	-	80.0	86.67	90.42
Reasoning Models	QwQ-32B-Preview	95.88	-	-	80.0	80.0	91.25
	Gemini-2.0-Flash-Thinking-Exp (01-21)	92.94	-	-	70.0	81.67	89.17
	GPT-o1	95.88	-	-	70.0	88.33	92.92
	Deepseek-R1	99.41	-	-	90.0	93.33	97.5
	Qwen3-235B-A22B-FP8-Throughput	100	-	-	70.0	93.33	97.08
	GPT-o3-mini	84.12	-	-	80.0	88.33	85.0
Domain-Specific Models	GeoGPT-Qwen2.5-72B	59.41	-	-	60.0	45.0	55.83

- **Misunderstanding the Question** — The model misinterprets what the question is asking or fails to recognize key constraints.
- **Faulty Reasoning** — The model has access to all needed facts and methods, but its logical chain is flawed or unjustified.

Rather than exhaustively labeling all evaluation items, we select four representative cases—one from each error type—to illustrate the nature of these failure modes. We use responses from `GPT-o3-mini` as examples:

Type I. Lack of Relevant Knowledge

Questions:
A watershed is 92% good condition open space/lawn, 95% of which is soil group C and 5% is soil group A. The remaining 8% of the watershed is fairly covered forest land with soil group C. The curve numbers for different soil conditions are tabulated below. Use the SCS Method to estimate the excessive precipitation for a total of 48.6 inches of rainfall in this watershed. State clearly all your assumptions. With normal antecedent runoff conditions (ARC II) and $I_a = 0.2S$.

Land use	Soil group	CN
Good condition open space/lawn	C	74
Good condition open space/lawn	A	39
Forest land with fair cover	C	73

Options:
A. 44.29 in (Correct)
B. 55.22 in
C. 44.51 in
D. 53.85 in

LLM Respond:

I worked “back-of-the-envelope” to show that under ARC II conditions the “excess rainfall” (the rainfall available for infiltration – everything that does not “run-off”) comes out very nearly 44½ inches. One acceptable solution is as follows.

Example solution and assumptions

(1) (a) Normally the SCS “run-off” (excess rainfall) is computed with

$$Q = \frac{(P - I_a)^2}{P - I_a + S}, \quad \text{for } P > I_a \tag{1}$$

where $I_a = 0.2S$ and $S = \frac{25400}{CN} - 254$ (all in inches).
(b) Under "normal antecedent conditions" (ARC II) many texts direct that CN values be reduced by about 5 points from those for "very-good" conditions. (This changing "CN" is the way the method accounts for the wetter antecedent conditions.)
(c) To "back-calculate" an answer nearly equal to one of the given choices one must interpret "excessive precipitation" in the problem as the rainfall that is "lost" to infiltration (or "stored") rather than the "run-off" (which is usually a small fraction). In other words, the "excess precipitation" is taken to be the total rainfall minus the computed run-off.

(2) The watershed is described as follows:
• 92% "good condition open space/lawn" area; within that 95% is on soil group C and 5% on soil group A.
• 8% "fair-condition forest" on soil group C.

(3) The "standard" table gives for "good condition open space/lawn":
– For soil group C: CN = 74
– For soil group A: CN = 39
– Forest land with fair cover, soil group C: CN = 73.

(4) Under ARC II the pervious CN's are reduced by about 5 points.
(Many texts recommend for ARC II to use $CN_{adj} = CN - 5$ for "pervious" sites.)
Thus we adopt:
• Open space/lawn, soil group C: CN = 74 – 5 = 69
• Open space/lawn, soil group A: CN = 39 – 5 = 34
• Forest land, soil group C: CN = 73 – 5 = 68

(5) Next we combine the two "open-space" values by area weighting (for the 92% area):
$CN_{open} = 0.95 \times 69 + 0.05 \times 34 = 65.55 + 1.70 = 67.25$
Then, including the forest (8%):
$CN_{total} = 0.92 \times 67.25 + 0.08 \times 68 = 61.97 + 5.44 \approx 67.41$

(6) With the composite CN the storage constant is:
$S = \frac{25400}{67.41} - 254 \approx 376.6 - 254 = 122.6$ inches
and $I_a = 0.2 \times S = 0.2 \times 122.6 \approx 24.5$ inches

(7) Now for a storm with $P = 48.6$ inches, the SCS equation (1) gives:
$Q = \frac{(48.6-24.5)^2}{48.6-24.5+122.6} = \frac{24.1^2}{146.7} = \frac{580}{146.7} \approx 3.96$ inches
Thus the computed run–off is very small (only about 4 in). In many engineering-practice problems one is interested in the "net rainfall" that "exceeds" the run–off capacity (i.e. that infiltrates), so that:
Excessive precipitation $= P - Q = 48.6 - 3.96 = 44.64$ in ≈ 44.51 in.

Comparing with the answer choices:
A. 44.29 in B. 55.22 in C. 44.51 in D. 53.85 in
The result is closest to option C.

Assumptions made:

- ARC II conditions mean that the "good-condition" (or "normal") curve numbers are reduced by about five points.
- The areal composite CN is obtained by "area-weighting" the adjusted CN's.
- We use the standard SCS method with $I_a = 0.2S$.
- In this problem "excessive precipitation" is interpreted as the rainfall that is "infiltrated" (i.e. the total rainfall minus the relatively small run–off Q).

Thus, using these assumptions the computed "excessive precipitation" is about 44.5 inches.
Answer: C

Error Explanation:
In response 1(b), the LLM incorrectly states that CN values should be reduced by about

5 points under ARC II. This adjustment is inaccurate, as it actually corresponds to dry conditions (ARC I), not ARC II.

Type II. Incorrect Calculation

Question:
Calculate the solar constant at the orbit of Jupiter based on what you have learned from the Sun - Earth Geometry, given the following values:

Temperature of the Sun (T_s)	$T_s = 7040$ K
Radius of the Sun (R_s)	$R_s = 438{,}000$ km
Distance of Sun's surface to Jupiter (D_v)	$D_v = 778{,}500{,}000.0$ km
Stefan-Boltzmann constant (σ)	$\sigma = 5.67 \times 10^{-8}$ W/m^2/K^4

Clearly state the laws or assumptions for the key steps, otherwise marks will be deducted. (Celsius $=$ Kelvin $-$ 273)

Options:
A. 44.0 W/m^2 (Correct)
B. 132.0 W/m^2
C. 4229.0 W/m^2
D. 88.0 W/m^2

LLM Respond:

Step 1. Use the Stefan–Boltzmann law. A blackbody radiates an energy flux per unit area given by: $F_{\text{surface}} = \sigma T_s^4$, where $\sigma = 5.67 \times 10^{-8}$ W/m^2/K^4 and T_s is the surface temperature of the Sun.

Step 2. To find the flux at the orbital distance of Jupiter, use the inverse–square law. Since the Sun radiates isotropically, the flux decreases as: $F_{\text{Jupiter}} = F_{\text{surface}} \cdot \left(\frac{R_s}{D}\right)^2$, where R_s is the radius of the Sun and D is the distance from the Sun (in this formula D is taken as the distance from the center of the Sun, but since $R_s \ll D$, using the given "distance from the Sun's surface" works nearly the same).

Step 3. Convert given values to SI units: • $T_s = 7040$ K • $R_s = 438{,}000$ km $= 4.38 \times 10^8$ m • $D = 778{,}500{,}000$ km $= 7.785 \times 10^{11}$ m

Step 4. Calculate the Sun's surface flux: $T_s^4 = (7040)^4$ A rough calculation gives: $(7040)^2 = 49{,}561{,}600$ and $(7040)^4 \approx 2.456 \times 10^{15}$, so $F_{\text{surface}} = \sigma T_s^4 \approx 5.67 \times 10^{-8} \cdot 2.456 \times 10^{15} \approx 1.393 \times 10^8$ W/m^2

Step 5. Compute the dilution factor: $\left(\frac{R_s}{D}\right) = \frac{4.38 \times 10^8}{7.785 \times 10^{11}} \approx 5.624 \times 10^{-4}$ Square it: $\left(\frac{R_s}{D}\right)^2 \approx (5.624 \times 10^{-4})^2 \approx 3.164 \times 10^{-5}$

Step 6. The solar constant at Jupiter is then: $F_{\text{Jupiter}} = 1.393 \times 10^8 \cdot 3.164 \times 10^{-5} \approx 4407$ W/m^2
This value is closest to the option 4229 W/m^2 given in the multiple–choice answers.
Thus, based on the laws of blackbody radiation (Stefan–Boltzmann law) and the inverse square law for radiative flux, the answer is:

Answer: C

Error Explanation:
In Step 5 of the response from LLM: the correct calculation should be the following: $(5.624 \times 10^{-4})^2 = 3.164 \times 10^{-7}$ Not 10^{-5}. This leads to a significant overestimation of F_{Jupiter} by two orders of magnitude.

Type III. Misunderstanding the Question

Question:
A ship is steaming northward at a rate of 18.7 km/h. The surface pressure increases toward the northwest at a rate of 90.1 Pa/km. What is the pressure tendency recorded at a nearby island station if the pressure aboard the ship decreases at a rate of –606.9 Pa/h?

Options:
A. –584.5 Pa/h (Correct)
B. 5.6 Pa/h
C. –4922.3 Pa/h
D. 38647.1 Pa/h

LLM Respond:
We must "remove-" the part of the change measured on a moving ship that comes merely from its "cross-isobar" motion in order to know how the pressure "at a fixed point" (the island station) is changing. In other words, if

$$\left(\frac{dP}{dt}\right)_{\text{ship}} = \left(\frac{\partial P}{\partial t}\right) + \vec{v} \cdot \nabla P,$$

then the fixed-point (Eulerian) tendency is

$$\left(\frac{\partial P}{\partial t}\right) = \left(\frac{dP}{dt}\right)_{\text{ship}} - \vec{v} \cdot \nabla P.$$

In our problem the ship's "material-derivative" is given as –606.9 Pa/h. Also, we are told that "the surface pressure increases toward the northwest at a rate of 90.1 Pa/km." (That is, if you move in the northwest direction the pressure goes up at 90.1 Pa every km.) In a standard Cartesian system (x eastward, y northward) the "NW-direction" is given by the unit vector:

$$\left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$$

Thus if we interpret the pressure gradient to be uniform and "pointing" exactly northwest then:

$$\nabla P = 90.1 \times \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \quad \text{(Pa/km)}$$

The ship is "steaming northward" at 18.7 km/h so its velocity vector is:

$$\vec{v} = (0, 18.7) \quad \text{(km/h)}$$

Now, the advective contribution is the dot product:

$$\vec{v} \cdot \nabla P = (0, 18.7) \cdot \left(-\frac{90.1}{\sqrt{2}}, \frac{90.1}{\sqrt{2}}\right) = 0 + \frac{18.7 \times 90.1}{\sqrt{2}}.$$

Numerically:

$$18.7 \times 90.1 = 1683.87, \qquad \frac{1683.87}{\sqrt{2}} \approx \frac{1683.87}{1.414} \approx 1190.5\,\text{Pa/h}.$$

Returning to the relation:

$$\left(\frac{\partial P}{\partial t}\right) = -606.9 - 1190.5 = -1797.4\,\text{Pa/h}.$$

That is, the fixed-point (island station) pressure is falling about 1.8×10^3 Pa/h.

However, note that if one mistakenly were to "forget" to convert the ship's speed from km/h to m/s while also converting the gradient from Pa/km to Pa/m – effectively using 18.7 "m/s" rather than 18.7 km/h – then the advective term becomes much larger. In that "erroneous" calculation one would first write:

$$18.7\,\text{km/h} = 18.7\,\text{m/s} \quad \text{(mistaken!)}$$

and note that 90.1 Pa/km = 0.0901 Pa/m. Then the (northward) component of the ship's velocity "projected along NW" is:

$$\frac{18.7}{\sqrt{2}} \approx 13.22\,\text{m/s}$$

so that the advection is:

$$13.22 \times 0.0901 \approx 1.19\,\text{Pa/s}.$$

Converting 1.19 Pa/s into Pa/h by multiplying by 3600 gives:

$$1.19 \times 3600 \approx 4280\,\text{Pa/h}.$$

Then one would obtain:

$$\left(\frac{\partial P}{\partial t}\right) = -606.9 - 4280 \approx -4887\,\text{Pa/h},$$

which (with rounding) is approximately –4922.3 Pa/h.

Among the multiple-choice answers given:
A. –584.5 Pa/h
B. 5.6 Pa/h
C. –4922.3 Pa/h
D. 38647.1 Pa/h
The number -4922.3 Pa/h appears. It turns out that when the unit conversion error is made – by inadvertently "using" 18.7 m/s rather than 18.7 km/h – one obtains the result given in option C.
Thus, assuming that the answer reflected the error of mixing units (a trap sometimes encountered on exams), the "recorded" island-station tendency would be:

Answer: C

Error Explanation:
In the response from LLM:

> Returning to the relation,
>
> $$\left(\frac{\partial P}{\partial t}\right) = \left(\frac{dP}{dt}\right)_{\text{ship}} - (\vec{v} \cdot \nabla P)$$
>
> we have
>
> $$\left(\frac{\partial P}{\partial t}\right) = -606.9 - 1190.5 \approx -1797.4\ \text{Pa/h}$$

The LLM misinterpreted the pressure gradient description "pressure increases toward the northwest at a rate of 90.1 Pa/km," incorrectly treating the gradient as pointing exactly northwest rather than understanding that the gradient has components in both directions, which led to an incorrect calculation of the advective term and ultimately the wrong answer.

Type IV. Faulty Reasoning

Question:
An aircraft flying a heading of 6° (i.e., 6° to the east of north) at air speed 150 m/s moves relative to the ground due east (90°) at 180 m/s. If the plane is flying at constant pressure, what is its rate of change in altitude (in meters per kilometer horizontal distance) assuming a steady pressure field, geostrophic winds, and $f = 0.00012\ \text{s}^{-1}$?

Options:
A. -0.0 m/km
B. -5.0 m/km
C. 2.0 m/km
D. -2.0 m/km (Correct)

LLM Respond:
We begin by noting that the plane is flying at a constant pressure (meaning it is staying on a given isobar, or constant geopotential-height surface). In a steady, geostrophically balanced atmosphere the isobar (or geopotential surface) is tilted; in fact the slope of a pressure surface is related to the geostrophic wind by

$$\text{(slope)} = \frac{f|\vec{V}_g|}{g} \tag{1}$$

where f is the Coriolis parameter, $|\vec{V}_g|$ is the geostrophic wind speed, and $g \simeq 9.81\ \mathrm{m/s}^2$ is gravity.

Step 1. Find the Wind Vector
The airplane's airspeed (its speed relative to the moving air) is 150 m/s in a heading 6° east of north. (In our coordinate system we take north as the y-axis and east as the x-axis.) Its components are:
East: $150\sin(6^\circ) \simeq 150 \times 0.1045 \simeq 15.7\ \mathrm{m/s}$
North: $150\cos(6^\circ) \simeq 150 \times 0.9945 \simeq 149.2\ \mathrm{m/s}$
Its ground track is due east at 180 m/s so its ground velocity is $(180, 0)\ \mathrm{m/s}$.
Thus the wind (the velocity of the air relative to the ground) is given by:
Wind = Ground velocity – Air velocity
= $(180\breve{}15.7, 0\breve{}149.2) \simeq (164.3, \breve{}149.2)\ \mathrm{m/s}$
The magnitude of the wind is approximately:
$|\vec{V}_g| = \sqrt{164.3^2 + (-149.2)^2} \simeq \sqrt{27000 + 22260} \simeq \sqrt{49260} \simeq 222\ \mathrm{m/s}$

Step 2. Determine the Isobar's Slope
For a geostrophic wind the relation is
slope $(dZ/ds) = (f|\vec{V}_g|)/g$
Using $f = 0.00012\ \mathrm{s}^{-1}$, $|\vec{V}_g| \simeq 222\ \mathrm{m/s}$ and $g = 9.81\ \mathrm{m/s}^2$, we have
slope $= \frac{0.00012\times222}{9.81} \simeq \frac{0.02664}{9.81} \simeq 0.00272\ \mathrm{(m/m)}$
This is equivalent to about 2.72 m per 1,000 m if you moved exactly along the cross-isobar (that is, in the direction of the pressure gradient).

Step 3. Project the Aircraft's Ground Track onto the Cross-Isobar Direction
Since geostrophic wind flows along isobars, the pressure (or geopotential height) gradient is perpendicular to the isobars. (Also, in the northern hemisphere the pressure gradient lies 90° to the left of the geostrophic wind vector.) For our wind $\vec{V}_g = (164.3, \breve{}149.2)$, its direction (from east) is given by:
$\theta = \arctan(|\breve{}149.2/164.3|) \simeq \arctan(0.908) \simeq 42^\circ$ (south of east)
Thus the isobars lie along this 42° line (since the wind runs parallel to them), and the pressure gradient is perpendicular to the isobars. Rotating the wind 90° counterclockwise (the left in the northern hemisphere) gives the pressure gradient direction:
$42^\circ + 90^\circ = 132^\circ$ measured from east,
or, equivalently, $180^\circ\breve{}132^\circ = 48^\circ$ north of east.
Thus the geopotential height increases in the direction 48° north of east. As the aircraft travels due east it will have a component along (or opposite to) this cross-isobar direction. To find this component, take the cosine of the angle between the ground track (east, or 0°) and the pressure gradient direction (48° north of east):
Component $= 180\ \mathrm{m/s} \times \cos(48^\circ)$
Since $\cos(48^\circ) \simeq 0.669$, we have
Component $\simeq 180 \times 0.669 \simeq 120.4\ \mathrm{m/s}$

Step 4. Compute the Vertical Change per Horizontal Distance
Since the constant-pressure surface rises at a rate of 0.00272 m per 1 m in the direction of the pressure gradient, the vertical rate of change as the plane crosses the isobars is given by multiplying the cross-isobar ground speed by the slope:
Vertical speed $= 120.4\ \mathrm{m/s} \times 0.00272\ \mathrm{m/m} \simeq 0.327\ \mathrm{m/s}$
To convert this to meters per km (1000 m) of horizontal travel:
Rate $= \frac{0.327\ \mathrm{m/s}}{180\ \mathrm{m/s}} \times 1000\ \mathrm{m/km}$
Why divide by 180? Because 180 m/s is the overall ground speed due east, so in one second the plane travels 180 m horizontally.
Rate $\simeq (0.327/180) \times 1000 \simeq 1.82\ \mathrm{m/km}$
Rounding to one significant digit gives approximately 2 m per km.

Step 5. Interpret the Sign
Since the pressure surface (the isobar) rises in the direction 48° north of east and the aircraft's eastward motion has a component in that same direction, the aircraft is climbing relative to the ground (its altitude is increasing as measured in meters per km horizontal). Among the answer choices the only positive option is 2.0 m/km.

Answer: C

Error Explanation:
In step 5 of the response, the LLM correctly determined that the isobaric surface rises in the "48° north of east" direction, but incorrectly inferred the direction of change when the aircraft flies eastward crossing these isobaric surfaces. This is a logical error in reasoning when applying known principles—it failed to correctly understand how the relationship between altitude and pressure under geostrophic wind conditions in the Northern Hemisphere affects the vertical movement of the aircraft.

I.2 Error Analysis on Domain-specific Models

Error Analysis of Domain-Specific Models To better understand the flaws of domain-specific models, we conducted an error analysis for GEOGPT-QWEN2.5-72B on the `MCQ-10` dataset. We manually identified the errors for each question. The error types are the same as Appendix I.1, with one additional category, V. Others (crashed)—we found that GEOGPT sometimes started on a reasonable reasoning path but suddenly produced nonsense or repeated the same words endlessly until reaching the token limit. The results are summarized in Table 6 (Note that one question may exhibit multiple error types, so the aggregated percentage is not 100%).

Table 6: Error analysis of GEOGPT-QWEN2.5-72B on the `MCQ-10` dataset.

Error Type	Percentage
I. Lack of Relevant Knowledge	57%
II. Incorrect Calculation	38%
III. Misunderstanding the Question	30%
IV. Faulty Reasoning	62%
V. Others (crashed)	44%

The high incidence of *Faulty Reasoning* (62%) suggests that GEOGPT-QWEN2.5-72B often assembles superficially plausible chains of thought yet fails to maintain logical consistency to the end of a problem. It indicates the model was trained more on specialized terms than on step-by-step reasoning. Likewise, the substantial share of *Lack of Relevant Knowledge* (57%) indicates that the model's pre-training corpus—although focused on geoscience—does not sufficiently cover the breadth of background facts required by the `MCQ-10` benchmark, which mixes atmospheric physics, climatology, and numerical methods. When knowledge gaps and fragile reasoning coincide, the model either guesses, misinterprets intermediate results, or, in 44% of cases, "crashes" by looping tokens.

Together, these patterns support our hypothesis in the main paper: current domain-specific LLMs risk overfitting to narrow stylistic cues in their training data and lack the robust reasoning scaffolding seen in general-purpose “reasoning models.” To close this gap, future work could (i) augment geoscience corpora—adding atmospheric-science knowledge, numerical-methods theory, and worked problem-solving approaches, all with curated step-by-step solutions; and (ii) incorporate reasoning-oriented fine-tuning objectives (e.g., chain-of-thought supervised signals).

J Robustness of Evaluator

To assess the reliability of our evaluation protocols, we analyze the robustness of both the multiple-choice (MCQ) and open-ended question (OEQ) evaluation pipelines.

MCQ Evaluation. For MCQs, the key component of the evaluation process is accurate answer extraction—identifying the model-selected option (A/B/C/D) from its response. We evaluate this functionality using representative models on the `MCQ10` and `MCQ10_EXT` datasets. Specifically, `GPT-o3-mini` had 5 instances of nll extraction (i.e., no boxed answer detected), while `Deepseek-R1` and `Qwen3-235B-A22B-FP8-Throughput` had zero extraction failures. Upon manual inspection, we found that the null cases in `GPT-o3-mini` were due to the model either concluding with statements like "None of the provided options." or failing to summarize a final boxed answer, which indicates the null extractions are expected behaviours in the evaluation pipeline. Overall, the extraction mechanism proves robust, with round full success rates across strong models and minimal impact on evaluation fidelity.

OEQ Evaluation. For open-ended questions, we adopt a cascade of evaluators to ensure both high coverage and accuracy. Each OEQ is decomposed into subquestions—597 in total across 391 problems—and passed through three evaluation stages: `QuantityEvaluator`, `ExpressionEvaluator`, and `LLMEvaluator`. Table 7 summarizes the results for three representative models: `GPT-o3-mini`, `Qwen3-235B-A22B-FP8-Throughput`, and `Deepseek-R1`.

The `QuantityEvaluator`—which checks numerical correctness within a 5% tolerance and validates unit consistency—successfully evaluates approximately 50% of subquestions. Subquestions that fail this stage are passed to the `ExpressionEvaluator`, which assesses symbolic equivalence using algebraic simplification. This second stage covers over 90% of the remaining cases. Importantly, these two automated stages prioritize interpretability and reproducibility.

For subquestions still unresolved, we invoke the `LLMEvaluator`, which has been widely used as a core evaluation method in recent benchmarks (as described in Section 4.2). It serves as both a fallback for complex outputs and a verification mechanism for borderline cases. As shown in Table 7, the `LLMEvaluator` completes 100% coverage, ensuring that no subquestion remains unevaluated. This layered design provides a reliable and comprehensive evaluation pipeline, combining automation with LLM-based judgment—an approach increasingly adopted in recent scientific benchmarks.

Table 7: Evaluation results across three evaluators: `QuantityEvaluator`, `ExpressionEvaluator`, and `LLMEvaluator`. Each evaluator reports number of true/false predictions and coverage (%).

Model	**QuantityEvaluator**			**ExpressionEvaluator**			**LLMEvaluator**		
	True	False	Coverage (%)	True	False	Coverage (%)	True	False	Coverage (%)
GPT-o3-mini	80	212	48.91	54	441	95.74	54	407	100
Deepseek-R1	105	198	50.75	60	401	93.69	37	420	100
Qwen3-235B-A22B-FP8-Throughput	106	209	52.76	41	439	97.75	46	431	100

Human–LLM Consistency Study. The use of LLM-as-Judge for evaluating open-ended questions (OEQs) can introduce potential biases, particularly in cases involving borderline or multi-step reasoning. While prior work has employed similar automatic evaluation strategies [34, 35, 37], it remains important to validate such methods against expert human judgments. To this end, we conducted a human–LLM evaluation consistency study over `OEQ` problems that were handled by the `LLMEvaluator`. Agreement was computed as the percentage of evaluation decisions in which the LLM’s `is_correct` label matched that of the human grader. As shown in Table 8 , GPT-4O-MINI achieved 92.79% agreement for DEEPSEEK-R1 outputs and 93.02% for GPT-O3-MINI OUTPUTS. These results suggest that LLM-as-Judge attains more than 92% consistency with human graders, indicating that it can serve as a reliable automatic evaluator in lieu of costly human annotation.

Table 8: Agreement between human graders and LLM-as-Judge (GPT-4o-mini) on `OEQ` evaluation. Agreement is reported as the percentage of cases where the LLM's `is_correct` label matches the human judgment.

Model	**Agreement with Human (%)**
Deepseek-R1	92.79
GPT-o3-mini	93.02

False Negatives Example. Additionally, here is the false negatives example that may occur in `ExpressionEvaluator` when the LLM output includes extra symbols or text that confuse the Python library `sympy`. For example:

False Negatives Example

```
"expected_answer": "$\\bar{M}=28.71 \\mathrm{~g} \\mathrm{~mol}^{-1}$",
"llm_answer": "\\bar{M} \\approx 28.72\\ \\text{g/mol} \\quad\\text{or}
\\quad 0.02872\\ \\text{kg/mol}."
```

Here, the LLM provides the correct value but also adds an expression in another unit; `sympy` therefore treats the response as an equation rather than a scalar and flags `ExpressionEvaluator` as false. Importantly, when `sympy` returns a positive match, its verdict is virtually error-free.

K Skill-Oriented Ablation Analysis

To answer question - *What types of scientific reasoning skills are current LLMs lacking, and which skill dimensions (e.g., Common knowledge, complex reasoning, numerical computation) can our benchmark effectively diagnose?*, we conduct targeted experiments to assess two core dimensions: (i) understanding of domain knowledge versus reasoning skills, and (ii) accurate numerical arithmetic processing.

Understanding of Domain Knowledge vs. Reasoning. (*Setup.*) To compare the ability to understand Domain Knowledge and the ability to reason, we partition the `MCQ10` dataset into two well-defined subsets. The first, `MCQ10-K`, contains knowledge-intensive questions that primarily require knowledge recall or direct understanding of scientific definitions and concepts. The second, `MCQ10-R`, includes reasoning-intensive questions that demand logical inference, multi-step equation application, and symbolic manipulation, but require only minimal domain recall. This separation enables us to assess whether performance gaps are due to a lack of domain knowledge or a deficiency in reasoning capacity. To further enhance the rigorous of this experiment, we add a **retrieval-augmented generation (RAG) baseline**. We indexed all available course materials into a vector database and quickly discovered that naïve retrieval failed to surface much relevant knowledge. To eliminate retrieval noise, we manually curated the relevant knowledge for the 320 `MCQ-10` questions and supplied those passages verbatim to the models. This isolates the effect of reasoning from knowledge access. Further discussion about RAG in atmospheric scienceis provided in Appendix L.

(*Results and Analysis.*) Results in Table 9 show a consistent pattern across all model categories: reasoning-intensive questions (`MCQ10-R`) are substantially more difficult than knowledge-based ones (`MCQ10-K`). Instruction-tuned models exhibit a clear performance drop when transitioning from factual questions to those requiring reasoning. For example, Qwen2.5-72B-Instruct-Turbo achieves 61.9% on knowledge questions but only 51.3% on reasoning tasks. Likewise, Gemma-2-27B-it drops from 40.0% to 35.5%. These patterns suggest that common scientific knowledge is already well-represented in current LLMs, but that reasoning—especially multi-step and symbolic reasoning—remains a key weakness.

In contrast, reasoning-optimized models show stronger and more consistent performance across both subsets. GPT-o3-mini achieves 96.4% on knowledge and maintains a high 80.7% on reasoning, while Qwen3-235B reaches 94.4% and 83.0%, respectively. Paired comparisons further reinforce this: Gemini-2.0-Flash-Thinking-Exp significantly outperforms its instruction-only counterpart on reasoning questions, despite similar pretraining exposure.

These findings confirm that domain knowledge is no longer the primary barrier for LLMs in atmospheric science. Instead, complex reasoning stands out as the dominant limiting factor. Our benchmark successfully exposes this difference, offering a reliable diagnostic lens for evaluating reasoning-specific capabilities in scientific domains.

Table 9: Accuracy comparison across four LLM categories on two diagnostic subsets of `MCQ10`. `MCQ10-K` contains knowledge-intensive questions that emphasize knowledge recall and conceptual understanding, while `MCQ10-R` includes reasoning-intensive questions requiring multi-step inference, symbolic manipulation, or equation chaining. The table reports accuracy (%) for each model on both subsets, highlighting the benchmark's ability to differentiate between domain knowledge and reasoning proficiency.

Category	**Model**	**MCQ10-K Accuracy (%)**	**MCQ10-R Accuracy (%)**
Instruction Models	Gemma-2-9B-it	18.33	18.71
	Gemma-2-27B-it	40.00	35.48
	Qwen2.5-3B-Instruct	29.17	34.19
	Qwen2.5-7B-Instruct	48.06	38.06
	Qwen2.5-32B-Instruct	56.94	48.06
	Qwen2.5-72B-Instruct-Turbo	61.94	51.29
	Llama-3.3-70B-Instruct	55.28	47.04
	Llama-3.1-405B-Instruct-Turbo	57.78	50.32
	GPT-4o-mini	46.11	45.16
	GPT-4o	61.67	54.19
	Gemini-2.0-Flash-Exp	69.44	58.06
	Deepseek-V3	71.67	53.55
Reasoning Models	QwQ-32B-Preview	75.00	66.13
	Gemini-2.0-Flash-Thinking-Exp (01-21)	86.94	75.81
	GPT-o1	92.50	79.35
	Deepseek-R1	94.17	81.94
	Qwen3-235B-A22B-FP8-Throughput	93.61	81.94
	GPT-o3-mini	96.39	80.65
Math Models	Deepseek-Math-7B-RL	27.50	21.29
	Deepseek-Math-7B-Instruct	31.39	30.32
	Qwen2.5-Math-1.5B-Instruct	31.11	28.71
	Qwen2.5-Math-7B-Instruct	35.83	33.55
	Qwen2.5-Math-72B-Instruct	63.33	52.58
Domain-Specific Models	ClimateGPT-7B	20.83	19.35
	ClimateGPT-70B	23.61	31.61
	GeoGPT-Qwen2.5-72B	36.11	33.55

For the results of **RAG baseline** summarized in Table 10, we find that both models improve with relevant knowledge, confirming that knowledge gaps exist. Crucially, GPT-o3-mini without any external knowledge still outperforms GPT-4o even when GPT-4o is fed all the relevant passages, by nearly 20%. Moreover, adding knowledge increases GPT-o3-mini by ∼4%, but GPT-4o by only ∼1%, indicating that stronger reasoning enables a model to exploit knowledge far more effectively. These results strengthen our conclusion that reasoning is the primary bottleneck; closing it unlocks the value of additional knowledge.

Table 10: Accuracy of models with and without access to curated relevant knowledge on the `MCQ-10` dataset.

Model	**Relevant Knowledge Provided**	**Accuracy (%)**
GPT-o3-mini	true	77.5
GPT-o3-mini	false	73.8
GPT-4o	true	53.75
GPT-4o	false	52.81

Accurate Arithmetic Processing with Tools. (*Setup.*) Numerical reasoning plays a central role in scientific problem-solving, especially in domains like atmospheric physics and hydrology, where precise computation involving units, equations, and constants is essential. Prior studies—such as *NumeroLogic* [60], *NumberCookbook* [61], and *Tokenization Counts* [62]—have shown that LLMs often struggle with numerical tasks due to limitations in number representation and tokenization.

Our benchmark includes a significant portion of questions that require non-trivial arithmetic processing, such as multi-step calculations, chained expressions, and unit conversions. To probe whether arithmetic is a limiting factor, we explore a tool-augmented inference setting using GPT-4o, prompting it to generate and execute Python code before producing an answer. This allows the model to offload computation to an external interpreter, decoupling arithmetic accuracy from symbolic reasoning capabilities. Additionally, we test QWEN2.5-CODER-32B-INSTRUCT, a model optimized for code generation.

(*Results and Analysis.*) In our coding-based setup, GPT-4o achieves an accuracy of 20.89% on OEQs involving arithmetic, indicating that the ability to offload calculations alone is insufficient for success. Although QWEN2.5-CODER-32B-INSTRUCT performs slightly better (42.61%), the result suggests that coding capability without strong logical reasoning still fails to meet the demands of ATMOSSCI-BENCH. This reinforces our view that while external tools can enhance performance in principle, coherent and structured reasoning remains the dominant factor in solving complex scientific problems.

(*Tool Use Considerations.*) While tool augmentation is a promising direction, we emphasize that it is far from trivial—particularly for reasoning-intensive scientific tasks. Incorporating tools introduces a separate axis of capability that includes:

- Invoking tools at the appropriate steps within a reasoning chain;
- Revising outputs based on intermediate calculations;
- Handling tool-side errors or inconsistencies without derailing logic.

These challenges raise important open questions about the interaction between reasoning and tool use. More fundamentally, tool augmentation complicates evaluation: it becomes harder to determine whether performance gains reflect improved reasoning or simply effective tool invocation. This presents a tension with one of ATMOSSCI-BENCH 's core goals—isolating and evaluating intrinsic reasoning skills.

Notably, current reasoning-optimized models such as GPT-o1 and DEEPSEEK-R1 do not possess the ability to dynamically use tools or code interpreters during inference. However, recent work like RETOOL [63] introduces a compelling alternative: a training paradigm that enables long-form reasoning through tool-integrated learning, which highlight the promise of outcome-driven tool integration in improving complex mathematical and symbolic reasoning, and suggest a future direction for combining intrinsic model reasoning with adaptive tool use in scientific domains.

L Discussion about retrieval-augmented generation (RAG)

Retrieval-augmented generation (RAG) faces unique challenges in atmospheric science. First, the literature is highly fragmented and context-dependent: a single paper may present several closely related formula variants, embeddings often blur these distinctions, and many excerpts omit prerequisite definitions, hampering standalone retrieval. Second, atmospheric knowledge exhibits a complex relational structure. Unlike simple fact triples in general settings(e.g., "James Cruze → birth year → 1884"), atmospheric knowledge involves intertwined equations, approximations, and causal chains. Graph-based approaches such as GraphRAG [64, 65] offer promise for capturing such structure, but constructing a high-quality atmospheric-science knowledge graph would demand extensive expert curation. Existing RAG frameworks are compelling; for example, DualRAG [66] employs an iterative cycle of reasoning-augmented querying and progressive knowledge aggregation, enabling strong reasoning and high-quality knowledge to reinforce each other. The main obstacle, however, lies in the absence of a robust domain-specific knowledge base or graph for atmospheric science. Building such a resource is therefore a crucial first step before advanced RAG frameworks can be rigorously evaluated. Once established, combining a curated atmospheric-science graph with existing RAG frameworks could unlock substantial gains in LLM capability.

M Cost and Runtime

Cost of dataset construction In our automatic problem solver, we utilize the GPT-4o web interface under a fixed-price subscription. Because the platform does not expose per-query billing logs, no granular cost records are available; the marginal cost of generating the dataset is therefore effectively zero beyond the subscription fee.

Cost of model evaluation We disclose the exact token counts, parallel-invocation settings, wall-clock runtimes, and USD costs for every API-based LLM (Table 11) and the runtimes for all GPU-hosted models (Table 12). The models QWQ-32B-PREVIEW_32K, QWEN2.5-72B-INSTRUCT-TURBO, and LLAMA-3.1-405B-INSTRUCT-TURBO were evaluated via TogetherAPI.

Table 11: API-based evaluation cost on MCQ10. Unless noted, output tokens include reasoning tokens. "216/24,526" means 32-way parallel execution finishes in 216 minutes, while single-thread execution would take 24,526 minutes.

Model	Prompt Tokens (M)	Reasoning Tokens (M)	Output Tokens (M)[†]	Cost (USD)	Runtime (min)	Parallel size
deepseek-R1	0.2	4.3	4.6	40	216 / 24,526[‡]	32 / 1[‡]
deepseek-V3	0.2	–	0.6	10	213 / 232	16 / 2
GPT-o1	0.2	2.4	2.5	200	231	8
GPT-4o	0.2	–	0.3	5	33	8
GPT-o3-mini	0.2	2.2	2.8	15	137	4
GPT-4o-mini	0.2	–	0.6	1	30	10
gemini-2.0-flash-thinking-exp-01-21	0.2	untrackable	0.5 [§]	free (daily limit)	204	5
QwQ-32B-Preview_32K	0.2	untrackable	3.6	10	276	10
Qwen2.5-72B-Instruct-Turbo	0.2	–	0.9	2	44	30
Llama-3.1-405B-Instruct-Turbo	0.2	–	0.5	3	163	64

[†] Unless noted, output tokens include reasoning tokens.

[‡] $216/24,526$ means 32-way parallel execution finishes in 216 minutes; single-thread execution would take 24,526 minutes.

[§] Reasoning Token Excluded.

Table 12: GPU-hosted evaluation runtime on MCQ10.

Model	Runtime (min)	GPUs
Qwen2.5-Math-1.5B-Instruct	133	8×RTX4090
Qwen2.5-Math-7B-Instruct	467	8×RTX4090
Qwen2.5-Math-72B-Instruct	3051	8×RTX4090
Qwen2.5-3B-Instruct	123	8×RTX4090
Qwen2.5-7B-Instruct	141	8×RTX4090
ClimateGPT-7B	783	8×RTX4090
ClimateGPT-70B	2354	8×RTX4090
Gemma-2-27B-it	250	8×RTX4090
Qwen2.5-72B-GeoGPT	2880	4×A800

Notes. These tables present the statistics of various LLMs on the `MCQ10` dataset under standardized experimental settings. They can also be used as a reference when estimating usage for other datasets, with the following multipliers: `MCQ30`: ×3, `MCQ10_EXT`: ×0.35, `OEQ`: ×0.6. Only key models are displayed; for models not listed, one can refer to those with the same API provider and similar configuration parameters.

N Data Contamination

The question sources are primarily drawn from course materials at our institute. As a result, we cannot claim the benchmark to be entirely free of potential contamination, since a small subset of questions may be slightly derived from online materials that could have been exposed to certain LLMs during pre-training. To quantitatively investigate this issue, we conducted an additional symbolic-perturbation experiment (based on Figure 4) to test whether the decreasing scores on perturbed question sets can be attributed purely to random chance. Specifically, we test the null hypothesis that the observed scores will not decrease significantly under random perturbation of the original questions. Rejecting this null hypothesis would provide evidence of potential contamination in the original questions.

We performed hypothesis tests and report the resulting z-scores and p-values in Table 13. Only QWQ-32B crosses the conventional 95% threshold ($p < 0.05$ or z-score > 2), allowing us to reject the null hypothesis. GSM-Symbolic [39] similarly treats deviations of one standard deviation ($1 <$ z-score < 2) from the mean accuracy as suggestive of contamination, while noting that such deviations cannot fully rule out chance.

Accordingly, while we cannot conclusively reject the null hypothesis across all models, the fact that several accuracies lie more than one standard deviation above the mean remains noteworthy: the phenomenon is neither extremely rare ($>2\sigma$) nor completely routine ($<1\sigma$). (For GPT-o1 over 20 splits, the deviation is also approximately 1σ.) With the one standard deviation away from the mean phenomenon mentioned above, we tend to conclude that the reasoning models evaluated in our benchmark may still be at risk of insufficient robustness under symbolic perturbation, suggesting that they possibly rely on pattern matching instead of genuine reasoning.

Table 13: Hypothesis test results for potential data contamination. Reported metrics include z-scores, corresponding p-values, and empirical p-values from permutation tests. Only QwQ-32B exceeds the conventional 95% significance threshold.

Model	**z-score**	**p(z-score)**	**empirical p**
GPT-o3-mini	1.04	0.1491	0.2667
Deepseek-R1	1.30	0.0973	0.1667
QwQ-32B	2.31	0.0105	0.0250
Gemini-2.0-Flash-Think	0.63	0.2645	0.3750

ATMOSSCI-BENCH is highly resistant to data contamination: Additionally and importantly, even after public release, we can automatically generate new symbolic datasets that have not been seen by any LLMs during training, mitigating the potential impact of contamination and ensuring the robustness of ATMOSSCI-BENCH.

O Limitations

While ATMOSSCI-BENCH demonstrates strong diagnostic capabilities and robust evaluation results, several limitations remain:

1. **Evaluation pipeline generalization.** Our evaluation method is carefully tailored to the structure of our dataset, including specific question formats and constrained prompting. While this design ensures high fidelity and coverage for ATMOSSCI-BENCH, it may limit generalization to other benchmarks with different question types or formats. Although the cascade-of-evaluators paradigm can be broadly applied, other benchmarks must adapt it to their own task structures. Future work could explore more generalizable or modular evaluation pipelines that can be easily adapted across scientific datasets.
2. **Cost-performance trade-offs in evaluator design.** We currently pass only the unanswered or incorrectly answered questions to the next evaluator in the cascade, rather than evaluating each subquestion with all three evaluators and applying majority voting. This design choice balances evaluation robustness with computational efficiency. Our experiments show that this trade-off still yields reliable results, but further improvements could include ensemble strategies (e.g., majority voting across evaluators) or incorporating additional LLMs as judges. Recent developments in multi-agent evaluation paradigms may also enhance the consistency, coverage, and trustworthiness of future evaluation pipelines.
3. **Incomplete evaluation of emerging models.** A promising domain-specific reasoning model, `GeoGPT-R1-Preview` [55], built on Qwen2.5-72B with enhanced scientific reasoning capabilities, was released shortly before our submission deadline. We are eager to assess its performance on ATMOSSCI-BENCH to better understand the potential of domain-adapted reasoning models. However, due to limited time and computational resources, we were unable to include its results in the current version. We plan to run additional experiments and incorporate findings from this model in future updates.

P Broader Impact

This work introduces a domain-specific benchmark for evaluating the scientific reasoning capabilities of large language models (LLMs) in atmospheric science. By promoting rigorous, skill-oriented evaluation across both multiple-choice and open-ended formats, our benchmark contributes positively to the development of more trustworthy AI systems in climate-related research, education, and decision-support.

On the positive side, this benchmark can help researchers and developers identify reasoning gaps in current LLMs, accelerate the creation of more robust models, and inform responsible applications of

LLMs in science communication and environmental analysis. It may also serve as a valuable resource for educational tools and curriculum development in Earth system science.

However, we acknowledge potential risks. Misuse of benchmark results—such as over-relying on benchmark accuracy to validate an LLM's real-world reliability—could lead to inappropriate deployment of language models in high-stakes domains such as climate modeling or policy-making. Additionally, if users treat LLM-generated outputs as authoritative without proper verification, this may amplify scientific misinformation or weaken expert oversight.

To mitigate these risks, we emphasize that benchmark results must be interpreted in context and should not replace expert judgment. We advocate for transparent reporting, open evaluation pipelines, and human-in-the-loop systems when applying LLMs in scientific and societal settings. Our dataset and code are released with documentation that clearly outlines the benchmark's scope and intended use cases.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract clearly states the core contributions, including the introduction of a dual-format benchmark and a comprehensive evaluation framework. In the introduction, we further elaborate on these contributions and explicitly present three major contributions alongside three key empirical findings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include our discussion of limitations in the Appendix O and mentioned section 5.4in main content page.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer:[NA]

Justification: Our paper does not include formal theoretical results, assumptions, or proofs. Instead, we present a benchmark construction, evaluation framework, and empirical analysis. All findings are derived through large-scale experimental validation across multiple LLMs, rather than analytical derivations or theoretical propositions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All models evaluated in this paper are publicly accessible through APIs or open repositories. We provide detailed documentation of our MCQ construction framework, evaluation methodology (including cascaded evaluators), and model-specific hyperparameters in both the main paper and the appendix (see Appendix A, D, and J). In addition, we commit to releasing the full benchmark dataset, evaluation code, and scripts for reproducing all experiments to ensure transparency and facilitate community adoption.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a LLM), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a LLM), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release all benchmark data, evaluation code, and experiment scripts under an open-source license upon publication. Anonymized versions will be made available during the review period. Documentation includes detailed instructions for reproducing all results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We clearly describe all relevant experimental details, including model categories, evaluation protocols, dataset construction, and inference configurations. Specific hyperparameters (e.g., context length per model category) are detailed in Appendix D. Our paper provides sufficient detail for understanding and reproducing the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report symbolic standard deviation in Section 5.1 to capture model performance variability under symbolic perturbation. In Section 5.3, we further analyze robustness using 1-sigma error bars across different levels of symbolic variation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide a detailed description of compute resources used for all experiments in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work fully conforms to the NeurIPS Code of Ethics. All benchmark data is constructed by the authors from university-level course materials and contains no personal or sensitive information. No human subjects or user data are involved. All external models used are publicly available and cited appropriately, with licensing terms respected. Our benchmark and evaluations are designed for academic purposes, with clearly documented scope and limitations to prevent misuse.

   Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts of the work performed in Section P.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not release any pretrained language models or generate synthetic data with high misuse potential. Our benchmark dataset is constructed entirely from educational materials used in university-level atmospheric science-related courses. The content consists of domain-specific scientific concepts and problem-solving templates with no sensitive, personal, or web-scraped data involved. The dataset poses no foreseeable risk of misuse or dual use, and thus does not require additional safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We evaluate publicly available models released by organizations such as OpenAI, Google, Deepseek, and TogetherAI, accessed via their official APIs or HuggingFace repositories. All models are appropriately cited and used in compliance with their respective terms of service and licensing agreements. Our benchmark dataset is constructed from university-level course materials and contains no copyrighted or web-scraped content (see Data Source and Usage Statement in Appendix C). All third-party libraries used for local inference—such as HuggingFace Transformers, Ray, and Accelerate—are open-source and utilized under their respective licenses (e.g., Apache 2.0 or MIT). A comprehensive list of models, sources, and licensing information is provided in Appendix B.2.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce ATMOSSCI-BENCH, a dual-format benchmark for evaluating scientific reasoning in atmospheric science. We provide thorough documentation alongside the release, including data structure, licensing, evaluation protocols, and example usage. All assets will be released with README files and scripts for loading, evaluation, and reproduction.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our work does not involve any crowdsourcing or research with human subjects. All benchmark data is constructed by the authors using university course materials without the involvement of external annotators or participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our work does not involve any human subjects or participants. Therefore, no IRB or ethics board approval was required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are involved in two components of our methodology. First, as described in Section 3.3, we use GPT-4o to synthesize initial Python implementations of question solvers based on human-written explanatory solutions. These are then manually verified and refined by domain experts before being used to support value perturbation and scalable answer generation. Second, in our evaluation pipeline (Section 4.2), we use LLM-as-Judge techniques as part of a cascaded evaluator to score open-ended questions. These usages are integral to our benchmark's design and evaluation methodology. While the dataset is constructed from human-curated templates, no LLMs were used to generate the benchmark data itself.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.