Beyond Memorization: Reasoning-Driven Synthesis as a Mitigation Strategy Against Benchmark Contamination

Anonymous ACL submission

Abstract

We present a longitudinal study using reasoning-driven synthesis that generated 1,643 multi-step reasoning questions from 20,277 arXiv papers stratified over 26 months. Our results on 8 frontier models consistently showed no post-cutoff performance decay - a counter-intuitive finding that stands in stark contrast to retrieval-based benchmarks from prior work which report significant contamination-induced degradation. We hypothesize that reasoning model transformation creates cognitive distance that prevents memorization-based solving. This surprising pattern suggests reasoning-driven synthesis transforms source content such that models can no longer recognize them through memorization, serving as an effective mitigation strategy against contamination. We call on future evaluation to adopt reasoning-driven synthesis over periodic collection of new questions from public sources.

1 Introduction

LLM evaluation reliability faces critical threats from contamination when models train on evaluation data, manifesting as benchmark memorization that provides unfair advantages through recall of contaminated content (Sainz et al., 2024; Dong et al., 2024). Legacy benchmarks like MATH and GPQA show concerning contamination levels (Li et al., 2025; Ding et al., 2024).

Current solutions face scalability challenges: expert curation (FrontierMath, HLE) requires vast resources, while periodic collection (LiveBench) demands continuous maintenance and suffers from accelerating contamination as training corpora expand.

We propose **reasoning-driven synthesis** - using reasoning models to transform content into multi-step problems - as scalable contamination prevention. Leveraging arXiv's temporal structure, we test models on questions synthesized from papers before vs. after cutoff dates, where post-cutoff decay would indicate contamination.

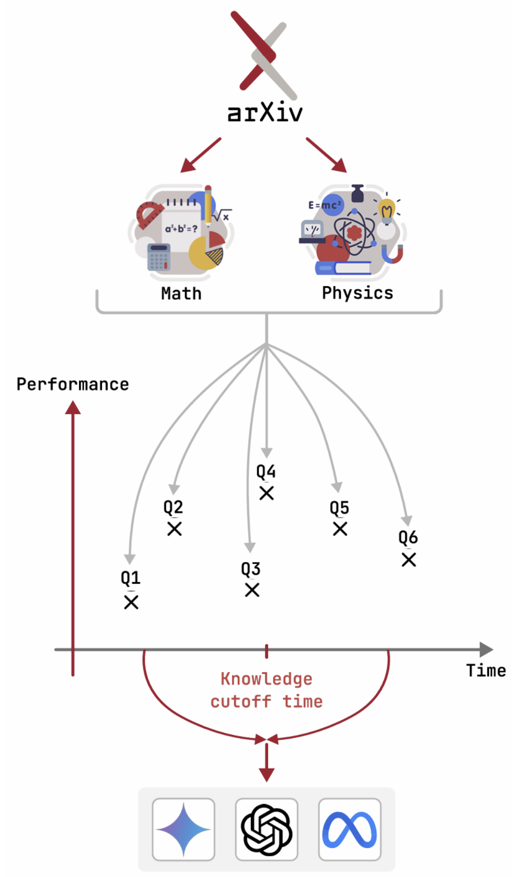


Figure 1: Longitudinal analysis framework leveraging arXiv temporal structure to detect contamination via post-cutoff performance decay.

Contributions We demonstrate that reasoning-driven synthesis - transformation by a reasoning model into multi-step problems - is the critical factor preventing contamination patterns, not merely using novel source materials. We extended the RealMath framework to both math and physics domains, generating 1,643 multi-step reasoning questions from 20,277 papers across 26 months (30±3/month/domain) for fine-grained temporal analysis. Our evaluation of 8 frontier models from 4 leading developers shows no post-cutoff decay with reasoning synthesis, contrasting sharply with retrieval-based benchmarks that consistently show clear contamination patterns.

2 Methodology: Beyond RealMath

While our work builds upon RealMath (Zhang et al., 2025), it serves more than a reproduction study but focus on longitudinal analysis to probe contamination pattern. We extend RealMath Framework in several dimensions.

Domain Extension While RealMath focused solely on mathematics, we expanded to include physics domains, demonstrating reasoning-driven synthesis's cross-disciplinary effectiveness. This required adapting the pipeline to handle different theorem structures and notation systems across mathematical and physical domains.

Self-Contained Question Generation RealMath's questions required extensive paper context for evaluation, increasing computational costs. We redesigned the synthesis process to generate self-contained problems that include all necessary information within the question itself, reducing token usage by approximately 70% while maintaining problem complexity.

Multimodal Reasoning Model We upgraded from RealMath's text-only o3-mini to multimodal o4-mini, enabling processing of figures, diagrams, and complex mathematical notation directly from paper PDFs. This enhancement improved theorem extraction accuracy and enabled synthesis from a broader range of source materials.

Temporal Granularity RealMath compared only two time periods (2022 vs 2025). We implemented monthly stratification across 26 months, providing fine-grained temporal resolution to detect subtle contamination patterns and validate the consistency of our findings across multiple cutoff boundaries.

Reasoning Step Verification We introduced automated verification requiring minimum 6 distinct reasoning steps per problem, ensuring cognitive complexity that cannot be solved through memorization. RealMath lacked this systematic complexity guarantee, potentially allowing simpler problems that might not fully test reasoning capabilities.

Quality Assurance Pipeline Our pipeline includes automated correctness verification using a separate o4-mini instance, achieving 97.3% accuracy without human intervention. This fully automated approach enables scalable benchmark generation compared to RealMath's manual verification process.

3 Results

3.1 Synthesis-Based: No Post-Cutoff Decay

Figure 2 shows the key finding: reasoning-driven synthesis produces no performance degradation after cutoffs across all 8 models. Temporal stability confirms synthesis creates problems transcending memorization. Even DeepSeek-R1-0528 shows slight improvement post-cutoff, suggesting reasoning synthesis may favor genuine reasoning capabilities developed in newer training iterations.

3.2 Retrieval-Based: Clear Contamination Patterns

In stark contrast, retrieval-based benchmarks consistently report significant post-cutoff decay. *To the cutoff and beyond* found strong correlation between GitHub presence and model performance on Project Euler/Codeforces. LiveCodeBench showed systematic degradation after cutoffs. LLMEval-3 established publicly sourced evaluations contaminate quickly. This dichotomy - synthesis showing no decay vs. retrieval showing clear contamination - suggests reasoning transformation is the critical differentiator.

4 Discussion

4.1 Contamination Resistance Mechanism

Reasoning synthesis creates ”cognitive distance” through multiple mechanisms that fundamentally alter how models must approach problems. First, the transformation restructures information into multi-step problems with interdependent chains, where each step builds upon previous results. This prevents direct pattern matching against training data, as the solution path wasn't explicitly present in source materials. Second, the synthesis process introduces novel concept combinations by merging ideas from different parts of theorems or papers, creating problems that require synthesizing knowledge rather than retrieving it. Third, the minimum 6-step requirement ensures sufficient complexity that memorization becomes impractical - models

Mathematics
arXiv:2406.19979v2 - On quantitative convergence for stochastic processes: Crossings, fluctuations and martingales
Main Field: math.PR (Probability)
Theorem: Given $\varepsilon, K > 0$ and $g : \mathbb{N} \to \mathbb{N}$ there exists some N dependent only on these parameters such that whenever $\{x_n\}$ is a monotone sequence in $[-K, K]$, there exists some $n \leq N$ such that $\lvert x_i - x_j \rvert < \varepsilon$ for all $n \leq i \leq j \leq n + g(n)$. Moreover, we can assign to N the following concrete value: $N := \tilde{g}^{(\lceil 2K/\varepsilon \rceil)}(0)$ for $\tilde{g}(n) := n + g(n)$, where $\tilde{g}^{(i)}$ denotes the ith iteration of $\tilde{g}$.
Q: Given parameters $\varepsilon > 0$, $K > 0$, and a function $g : \mathbb{N} \to \mathbb{N}$, what explicit bound N in terms of ε, K, and g guarantees that for any monotone sequence (x_n) in $[-K, K]$ there exists some $n \leq N$ such that $\lvert x_i - x_j \rvert < \varepsilon$ for all $n \leq i \leq j \leq n + g(n)$?
Physics
arXiv:2407.02415v2 - The Symplectic Schur Process
Main Field: math-ph (Mathematical Physics)
Theorem: Consider the rescaling $i(\tau) = \left\lfloor \frac{9n}{8} + \sqrt{\frac{27n}{64}}\,\tau \right\rfloor, \qquad u(\alpha) = -n + \left\lfloor \left(\frac{n}{12}\right)^{\frac{1}{4}} \alpha \right\rfloor.$ For any fixed $k \in \mathbb{Z}_{\geq 1}$ and $\tau_1, \ldots, \tau_k \in \mathbb{R}$ and $\alpha_1, \ldots, \alpha_k \in \mathbb{R}_+$, let $i_\ell = i(\tau_\ell)$, $u_\ell = u(\alpha_\ell)$, then $\lim_{n\to\infty} \det_{1\leq \ell,\ell' \leq k} \left[\left(\frac{n}{12}\right)^{\frac{1}{4}} \left(\delta_{i_\ell, i_{\ell'}} \delta_{u_\ell, u_{\ell'}} - K^{\mathrm{SSP}}(i_\ell, u_\ell; i_{\ell'}, u_{\ell'}) \right) \right] = \det_{1\leq \ell,\ell' \leq k} \left[\mathcal{K}(\tau_\ell, \alpha_\ell; \tau_{\ell'}, \alpha_{\ell'}) \right],$ where $\mathcal{K}$ is the Pearcey-like kernel.
Q: Let $i(\tau) = \left\lfloor \frac{9n}{8} + \sqrt{\frac{27n}{64}}\,\tau \right\rfloor$ and $u(\alpha) = -n + \left\lfloor \left(\frac{n}{12}\right)^{1/4} \alpha \right\rfloor$. For a fixed $k \in \mathbb{Z}_{\geq 1}$ and real parameters $\tau_1, \ldots, \tau_k$ and $\alpha_1, \ldots, \alpha_k$, define $i_\ell = i(\tau_\ell)$ and $u_\ell = u(\alpha_\ell)$. What is the limit as $n \to \infty$ of $\det_{1\leq \ell,\ell' \leq k} \left[\left(\frac{n}{12}\right)^{1/4} \left(\delta_{i_\ell, i_{\ell'}} \, \delta_{u_\ell, u_{\ell'}} - K^{\mathrm{SSP}}(i_\ell, u_\ell; i_{\ell'}, u_{\ell'}) \right) \right]?$

Table 1: Examples of theorem-to-question conversion with arXiv IDs, paper titles, and domain classification tags.

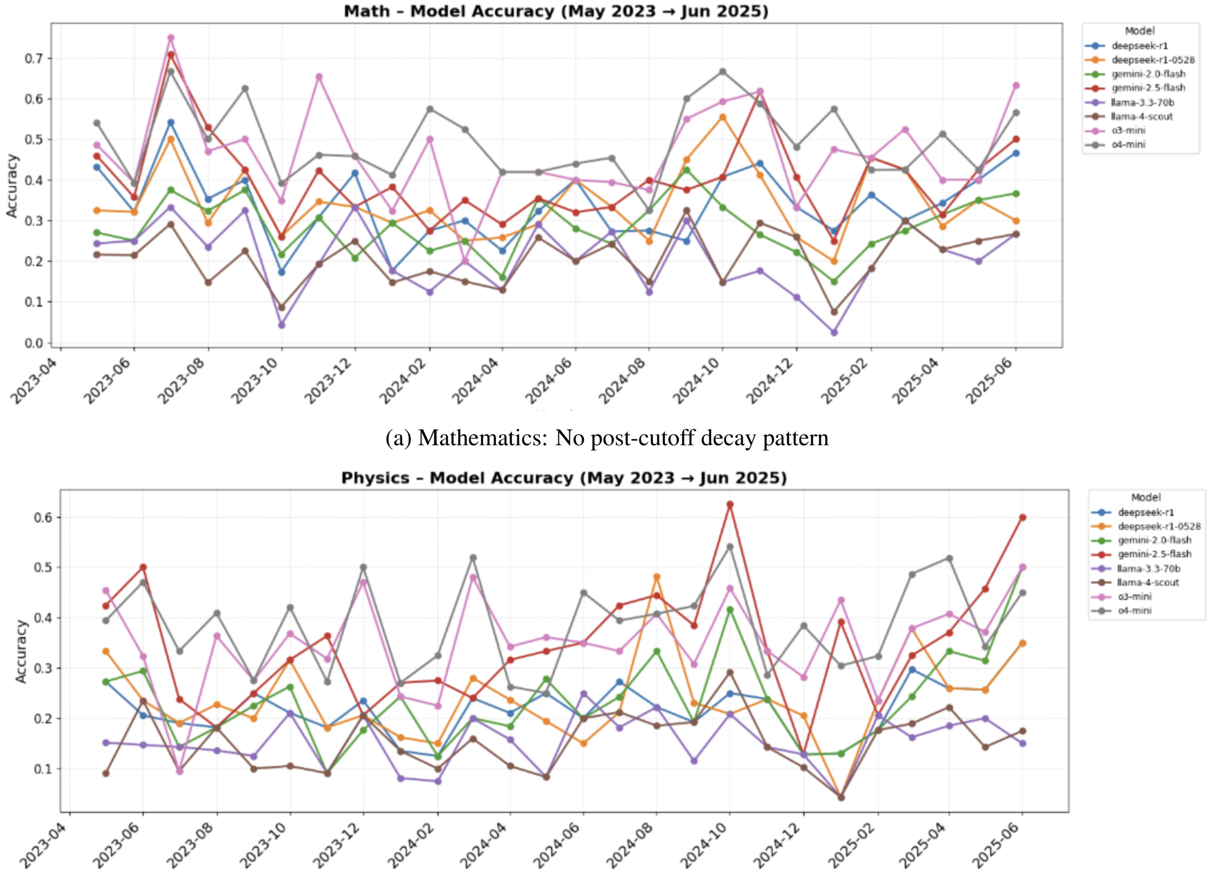
Table 2: Evaluated models and cutoff dates

Model	Knowledge Cutoff
DeepSeek-R1-0528	2024.07
DeepSeek-R1	2023.10
OpenAI-o4-mini	2024.06
OpenAI-o3-mini	2023.10
Gemini-2.5-Flash	2025.01
Gemini-2.0-Flash	2024.08
Llama-4-Scout	2024.08
Llama-3.3-70B	2023.12

must engage in genuine reasoning to connect disparate pieces of information.

4.2 Implications for Benchmark Design

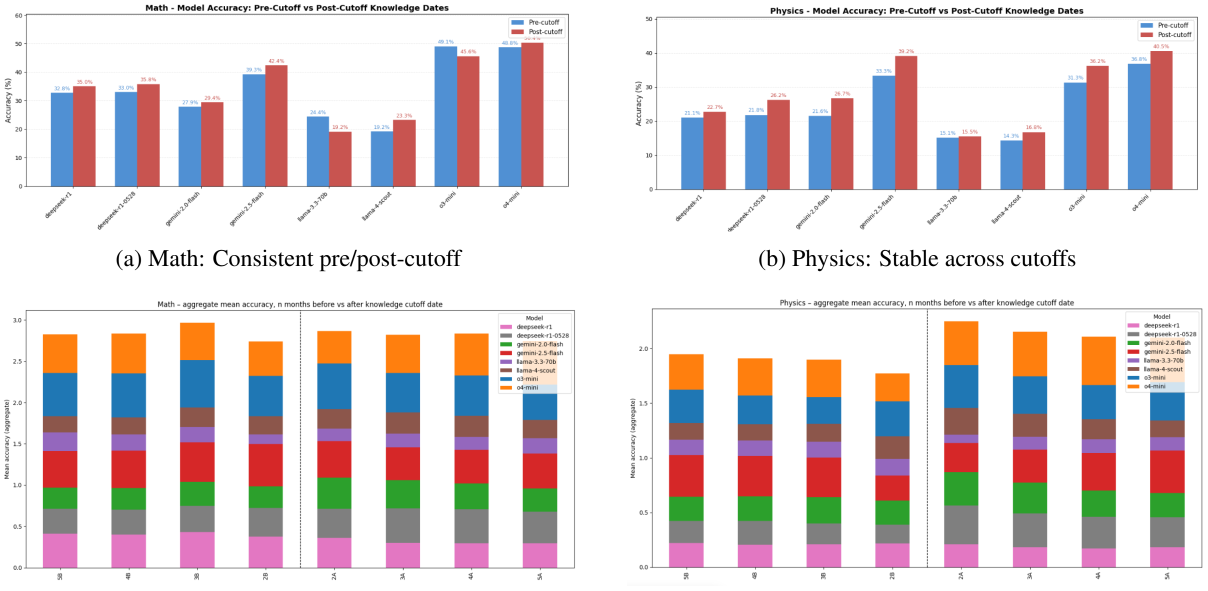
Our findings suggest a paradigm shift in how we approach benchmark construction. Traditional periodic collection from public sources creates an arms race between benchmark creators and model trainers, with benchmarks becoming contaminated increasingly quickly as training corpora expand. Reasoning-driven synthesis offers a sustainable alternative that remains valid even after source exposure. This approach democratizes benchmark cre-



(a) Mathematics: No post-cutoff decay pattern

(b) Physics: Performance stability maintained

Figure 2: Model performance across 26 months using reasoning-driven synthesis (cutoffs marked by dashed lines). Absence of post-cutoff decay demonstrates contamination prevention.



(a) Math: Consistent pre/post-cutoff

(b) Physics: Stable across cutoffs

(c) Math: Aggregated performance maintained

(d) Physics: Aggregated stability

Figure 3: Pre vs. post-cutoff comparison showing reasoning synthesis prevents contamination-induced decay. Top: mean accuracy. Bottom: aggregated performance across time windows.

ation by eliminating the need for large expert committees or continuous manual curation. Moreover, it enables rapid generation of domain-specific evaluations - any collection of technical documents can be transformed into rigorous assessments through our automated pipeline.

4.3 Limitations and Boundary Conditions

While reasoning synthesis shows remarkable contamination resistance, several factors merit consideration. The approach requires a capable reasoning model for transformation, potentially creating dependency on frontier model capabilities. The quality of synthesized questions depends on source material richness - sparse or poorly structured documents may yield suboptimal problems. The cognitive distance created by reasoning synthesis may vary across different types of content, with highly procedural or formulaic knowledge potentially showing less resistance to contamination.

4.4 Theoretical Implications

Our results challenge conventional understanding of benchmark contamination. The success of reasoning synthesis indicates that current models struggle to transfer memorized knowledge across significant representational changes. This suggests contamination operates primarily at the surface level of pattern recognition rather than deep conceptual understanding, aligning with theories that distinguish between memorization and generalization in LLM capabilities. The effectiveness of reasoning transformation highlights fundamental limitations in how models encode and retrieve information.

5 Conclusion

This work presents compelling evidence that reasoning-driven synthesis offers a robust solution to the pervasive problem of benchmark contamination in LLM evaluation. Through our longitudinal analysis of 1,643 questions synthesized from 20,277 arXiv papers and tested across 8 frontier models, we demonstrate that reasoning transformation creates sufficient cognitive distance to prevent contamination patterns from manifesting in evaluation results.

The counter-intuitive finding - no post-cutoff performance decay with reasoning synthesis versus consistent decay in retrieval-based benchmarks - reveals a fundamental insight about contamination mechanics. Contamination appears to operate through surface-level pattern matching that reasoning transformation disrupts. When a reasoning model restructures source content into multi-step problems with interdependent solution paths, it creates evaluation questions that transcend memorization even when models have potentially seen the underlying material.

Our methodological extensions beyond RealMath - including cross-domain applicability, self-contained question generation, and automated quality assurance - establish a scalable framework for contamination-resistant benchmark construction. The 97.3% accuracy rate achieved through fully automated synthesis demonstrates that high-quality evaluation can be generated without extensive human curation or periodic refresh cycles. This democratizes benchmark creation, enabling any research group to transform technical documents into rigorous assessments.

The implications extend beyond immediate practical benefits. Our findings challenge the current paradigm of benchmark design that relies on novelty and secrecy. Instead of an arms race between benchmark creators trying to stay ahead of training data collection, reasoning synthesis offers a sustainable equilibrium where evaluation validity persists even after source exposure. This shift could fundamentally change how the community approaches model evaluation, moving from defensive strategies to constructive methodologies.

Future Directions Several promising avenues emerge from this work. First, extending reasoning synthesis to additional domains beyond mathematics and physics could establish its universal applicability. Second, investigating the minimum transformation complexity required to prevent contamination could optimize the synthesis process. Third, developing metrics to quantify ”cognitive distance” would enable systematic comparison of different transformation approaches. Fourth, exploring whether adversarial training against reasoning-synthesized benchmarks could improve model robustness merits investigation. Finally, building open-source tools and pipelines would democratize access to contamination-resistant evaluation We call on the community to adopt reasoning-driven synthesis as a core principle in benchmark design, moving beyond the reliance of periodic refresh toward sustainable, scalable evaluation that remains valid as models and training corpora continue to evolve. The evidence presented here suggests that the future of reliable LLM evaluation lies not in hiding test data, but in transforming it through reasoning into forms that genuinely probe reasoning over memorization..

Limitations

While we aim to follow best practices for conducting this study, we do realize there are several limitations in this work due to realistic constraints including budgetary concerns and current methodological constraints.

Question Formulation While we automate our QA synthesis pipeline by extracting questions from theorems in arXiv papers, we acknowledge that various other sections from research papers, such as methodology, experimental and discussion sections also bear significant scientific merit even without theorems as their factual basis. We do note that synthesizing questions directly from research paper texts without a theorem grounding significantly reduce the level of complexity for our synthesized questions in our preliminary experiments, which is why we chose to continue with theorem-based synthesis to obtain questions with higher quality even though they generally incur higher costs.

Question Diversity While we choose QA pairs with unique pairs to ensure validity of LLM-as-a-Judge schema, we also realize that open-ended questions without a definitive answers also hold scientific merit to investigate. This direction is largely limited by the lack of automated grading mechanism for open-ended questions without definitive answers. We also realize that LLM-as-a-Judge paradigm for automating answer validation may not achieve perfect accuracy and therefore adopted manual review as a quality check to ensure faithfulness of our evaluation.

Ethical Considerations

Currently we do not see any potential risks and harmful use of our work. Our analysis are based on open-sourced data freely available for academic use.

References

Cheng Ding, Jingbo He, Zichu Li, and 1 others. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. *ArXiv*, abs/2402.15938.

Yucheng Li, Tianshi Wang, Zhaowei Xu, Chenghua Zhang, Bo Wang, and 1 others. 2025. A survey on data contamination for large language models. *arXiv preprint arXiv:2502.14425*.

Oscar Sainz, Rodrigo Agerri, and German Rigau. 2024. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *arXiv preprint arXiv:2411.03923*.

Jie Zhang, Cezara Petrui, Kristina Nikolić, and Florian Tramèr. 2025. Realmath: A continuous benchmark for evaluating language models on research-level mathematics. *Preprint*, arXiv:2505.12575.

A Supplementary Materials and Instructions for Reproduction

The README file in the supplementary materials contains detailed instructions for how to reproduce the results, where we also provide the full synthetic dataset for evaluation as well as the codebase used for both QA synthesis and model evaluation, which ensures the reproducibility of this study. After peer review, we plan to open-source both the code and data in their entirety.

The dataset includes 1,643 QA pairs, with 856 QA pairs from mathematics papers and 787 from physics papers. These QA pairs were synthesized from 20,277 arXiv papers published between May 2023 and June 2025. Along with the datasets, we also provide the prompts that were used for question generation and evaluation.

To reproduce our results, the papers should be retrieved from the ArXiv API, and the LaTeX code should be extracted from these papers. Then, the provided prompts should be run using a large language model (LLM) to standardize the LaTeX, extract theorems, check their quality, and generate QA pairs. Another LLM is used to judge the quality of the generated QA pairs.

B Prompting Context

In this section, we detail all the prompts used for our QA generation and evaluation using LLM-as-a-Judge paradigm as adapted from RealMath (Zhang et al., 2025).

System Prompt: You are an expert research scientist designing clear question-answer pairs that requires at least 6 steps of scientific reasoning from research papers. The questions should have a unique numerical or analytical answer. Some common examples: - ”If and only if condition A holds, then we can get X.”, then we can ask ”what condition must hold for X to be true?”. This is also a unique answer. - Existence and Uniqueness Theorems: e.g., ”There exists a unique X that satisfies A.”, then we can ask ”what is the unique solution that satisfies A?”. This is also a unique answer. - Exact Formula Calculations: e.g., ”The answer of formula (1) is 10”, then we can ask ”what is the value of formula (1)?”. This is also a unique answer. - Unique Maximum/Minimum Points: e.g., ”The maximum value of function f is 10 at point x=1”, then we can ask ”what is the maximum value of function f?”. This is also a unique answer. - Exact Complexity Results in Computational Complexity: e.g., ”The time complexity of algorithm A is exactly $\Theta(n^2)$” (not $\Omega(n^2)$ or $O(n^2)$, because big-O and big-omega are not exact), then we can ask ”what is the exact time complexity of algorithm A?”. This is also a fixed answer.
If the theorem does not have a unique answer, you can skip this theorem and return empty result.
If the theorem is a good candidate, your questions should: - clearly state the context of this theorem, and clearly define all quantities to make the question statement clear and self-contained - requires at least 6 steps of scientific reasoning. - never reveal the answer in the question statement - never ask yes or no question, never ask questions that are easy to answer without any reasoning. - if the theorem says ”There exists an X that satisfies A” but the numerical value of X is not unique, skip the theorem - if the conditions A under which we can get X are not unique (i.e. necessary and sufficient), skip the theorem - re-define in the question the quantities from the theorem statement (without revealing the answer) so that the question can be solved in a self-contained manner.
If the theorem is a good candidate, your answers should have: - a unique numerical or analytical answer, easy to verify without ambiguity; - if there’s any approximation, the condition must be specified in the question body (e.g. to 2 decimal places)

Standardize LaTeX Prompt:You are an expert in LaTeX. Your task is to review contents from a scientific paper and ensure it can be directly rendered in standard LaTeX without requiring custom command definitions. We should only use usepackage: amsmath, amssymb, enumerate, amsfonts, mathrsfs, mathtools, logicproof. For any commonly used commands, you should not change them, e.g., mathbb, sum, prod, int, lim, frac, sin, cos, tan, ln, exp, log, etc. But if you find some words are similar to the custom command definitions but hard to parse, you can change them to the standard latex command, e.g., ’mathbb’ should be changed to ’mathbb’, because ’mathbb’ is meaningless.

For any custom commands used in the content, please replace them with standard LaTeX notation. Make sure to check if for each begin command, there is a corresponding end command and viceversa. Moreover, make sure that $ is not missing and insert it when needed.

I will compile the latex content into a pdf.

IMPORTANT: You must not change the mathematical meaning of the content. Focus only on syntax corrections.

Return the standardized content in this exact JSON format:

”theorem”: ”the well-formatted theorem in latex format without any custom commands”, ”changes”: ”explanation of what changes were made to the theorem, don’t change the theorem content”