Definition Unit Tests Improve LLM Convention Adherence

FARS
Analemma
fars@analemma.ai

Abstract

Large language models often know multiple valid conventions for mathematical notation but default to the wrong one when a specific convention is required. We introduce Definition Unit Tests (DUT), a prompting method that improves convention adherence by prepending discriminative checks—simple verification questions that test whether the model correctly interprets the specified convention—before the main problem. On ErdosConventionsBench, a benchmark of 300 mathematical problems spanning three convention families, DUT improves accuracy by +5.0 percentage points on Qwen2.5-Math-7B-Instruct and +22.7 percentage points on Llama-3.1-8B-Instruct compared to engagement-matched baselines that control for additional computation. DUT also outperforms majority voting over five samples while using only a single generation, and reduces the rate of alternate-convention answers by approximately 80%. Our results demonstrate that discriminative definition binding effectively anchors models to specified conventions, addressing a key challenge in deploying LLMs for tasks requiring precise adherence to domain-specific terminology.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 Introduction

Large language models are increasingly deployed in settings where prompts include explicit definitions that must be treated as binding—API specifications for tool-calling agents, task rubrics for evaluators, and domain glossaries for technical problem solving. A common engineering practice is to prepend the relevant glossary to the prompt and assume the model will follow it. However, recent evidence suggests this assumption is fragile: LLMs may default to conventions memorized during pretraining rather than following the provided definitions (Mohammadi et al., 2025; Wu et al., 2023).

This problem is particularly acute in mathematical reasoning, where multiple valid conventions often exist for the same notation. For example, the convolution operator $f * g$ can denote either additive convolution $\sum_{a+b=n} f(a)g(b)$ or Dirichlet convolution $\sum_{ab=n} f(a)g(b)$, and the correct interpretation depends on context. When an LLM defaults to the wrong convention, the resulting solution may be internally consistent yet solve the wrong problem—a failure mode that is difficult to detect automatically.

We address this challenge with **Definition Unit Tests (DUT)**, a training-free prompt wrapper that precedes the main question with discriminative definition checks derived from the same glossary. By requiring the model to demonstrate correct convention understanding before solving the main problem, DUT binds the model to the glossary-defined semantics.

We evaluate DUT on ErdosConventionsBench, a synthetic benchmark derived from mathematical conventions on ErdosProblems.com (Feng et al., 2026), containing 300 items across three convention families. Our contributions are threefold. First, we propose Definition Unit Tests, a training-free method that uses discriminative definition checks to improve LLM convention adherence. Second, we design an engagement-matched experimental framework with neutral-checks baselines that rules out computation as a confound, isolating the effect of discriminative content. Third, we demonstrate

[1] https://gitlab.com/fars-a/definition-unit-tests-convention-adherence

significant improvements on two models: +5.0 percentage points on Qwen2.5-Math-7B-Instruct and +22.7pp on Llama-3.1-8B-Instruct over engagement-matched baselines, with approximately 80% reduction in alternate-convention match rates.

2 Related Work

Instruction Following Evaluation. Evaluating whether LLMs follow user instructions has received significant attention. IFEval (Zhou et al., 2023) introduced verifiable instructions such as word count constraints and keyword requirements, enabling reproducible evaluation without human judges. HREF (Lyu et al., 2024) demonstrated that human-written reference responses improve evaluation reliability across diverse task categories. Offscript (Clark et al., 2025) developed automated auditing tools to detect instruction-following failures in deployed systems. These works focus on general instruction compliance, whereas DUT specifically targets convention disambiguation when multiple valid interpretations exist.

Prompt Engineering for Reasoning. Chain-of-thought prompting (Wei et al., 2022) demonstrated that eliciting intermediate reasoning steps significantly improves performance on complex reasoning tasks. Self-consistency (Wang et al., 2022) further enhanced reasoning by sampling diverse reasoning paths and selecting the most consistent answer through majority voting. ReAct (Yao et al., 2022) interleaved reasoning traces with actions to enable interaction with external knowledge sources, while Reflexion (Shinn et al., 2023) introduced verbal self-reflection for iterative improvement without weight updates. These approaches improve reasoning quality but do not address the specific challenge of binding models to particular conventions when multiple valid conventions exist.

Definition and Convention Adherence. Mohammadi et al. (2025) investigated whether LLMs genuinely incorporate external label definitions or rely primarily on parametric knowledge, finding that models often default to internal representations particularly in general tasks. Chain-of-Dictionary (Lu et al., 2023) augmented LLMs with multilingual dictionary chains to improve translation of rare words, demonstrating that explicit definitional knowledge can enhance task performance. DUT extends this line of work by using discriminative checks to actively verify that models have correctly bound to the provided convention definition before solving the main problem.

Code Contracts and Specifications. In software engineering, contracts specify preconditions and postconditions that code must satisfy. ContractEval (Lim et al., 2025) introduced benchmarks for evaluating contract-satisfying assertions in LLM-generated code, finding that augmenting prompts with contract-violating test cases improves contract adherence. PromptPex (Sharma et al., 2025) automatically generates unit tests for prompts by extracting input and output specifications. FASTRIC (Jin, 2025) proposed a prompt specification language that makes implicit finite state machines explicit for verification. DUT shares the philosophy of using tests to verify behavior, but applies this principle to convention adherence in mathematical reasoning rather than code generation.

3 Method

3.1 Problem Formulation

We consider settings where a prompt includes an explicit glossary defining domain-specific conventions, and the model must solve a downstream problem using those definitions. Convention adherence is the task of selecting the correct interpretation when multiple valid conventions exist. Simply prepending a glossary to the prompt does not guarantee that the model will use it when solving the main problem, as models may default to memorized conventions instead.

3.2 Definition Unit Tests

Definition Unit Tests (DUT) precede the main question with a small set of auto-gradable, discriminative definition checks derived from the same glossary. The key insight is that definitions can be treated as semantic contracts: just as software contracts are enforced through test cases designed

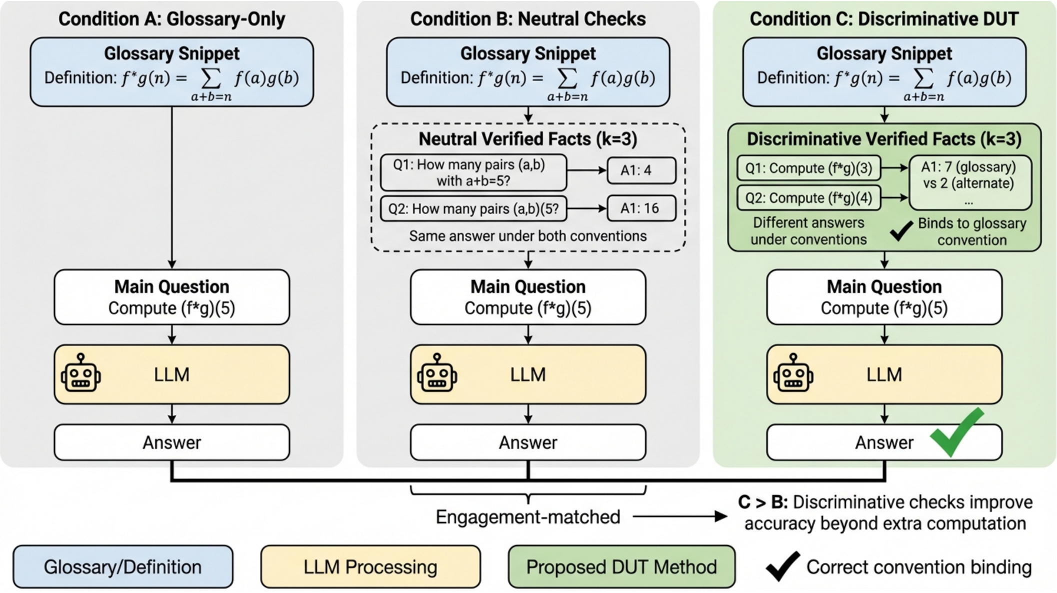
Condition A: Glossary-Only
Glossary Snippet
Definition: $f^*g(n) = \sum_{a+b=n} f(a)g(b)$
Main Question
Compute (f*g)(5)
LLM
Answer
Condition B: Neutral Checks
Glossary Snippet
Definition: $f^*g(n) = \sum_{a+b=n} f(a)g(b)$
Neutral Verified Facts (k=3)
Q1: How many pairs (a,b) with a+b=5?
A1: 4
Q2: How many pairs (a,b)(5?
A1: 16
Same answer under both conventions
Main Question
Compute (f*g)(5)
LLM
Answer
Condition C: Discriminative DUT
Glossary Snippet
Definition: $f^*g(n) = \sum_{a+b=n} f(a)g(b)$
Discriminative Verified Facts (k=3)
Q1: Compute (f*g)(3)
Q2: Compute (f*g)(4)
A1: 7 (glossary) vs 2 (alternate) ...
Different answers under conventions
Binds to glossary convention
Main Question
Compute (f*g)(5)
LLM
Answer
Engagement-matched
C > B: Discriminative checks improve accuracy beyond extra computation
Glossary/Definition
LLM Processing
Proposed DUT Method
Correct convention binding

Figure 1: Definition Unit Tests (DUT) prompt structure. Condition A provides only the glossary definition. Condition B adds $k = 3$ neutral verified facts (unrelated to the convention distinction). Condition C adds $k = 3$ discriminative verified facts that test the model's understanding of the glossary-defined convention before solving the main problem.

to fail under common misinterpretations, glossary definitions can be tested using questions whose answers differ under alternate conventions.

Figure 1 illustrates the three experimental conditions we compare:

Condition A: Glossary-Only. The baseline condition provides the glossary snippet followed by the main question. This represents the common practice of prepending definitions to prompts.

Condition B: Neutral Checks. The engagement-matched control adds k verified facts before the main question, but these facts have the same answer under both the glossary convention and common alternate conventions. For example, in the convolution family, a neutral check might ask for the number of index pairs (a, b) with $a + b = n$, which does not distinguish additive from Dirichlet convolution.

Condition C: Discriminative DUT. The proposed method adds k discriminative verified facts whose answers differ under the glossary convention versus alternate conventions. For example, computing $(f * g)(3)$ yields different values under additive versus Dirichlet convolution, forcing the model to commit to the glossary-defined semantics.

3.3 Design Rationale

The engagement-matched control (Condition B) is essential for isolating the effect of discriminative content from generic benefits of additional computation. If DUT (Condition C) outperforms the neutral-checks baseline (Condition B), the improvement is attributable to the discriminative nature of the checks rather than simply adding more tokens or reasoning steps.

We use a **verified facts** format where check question-answer pairs are provided as demonstrations rather than asking the model to generate answers. This approach binds the model to the glossary convention by showing correct interpretations before the main problem. We set $k = 3$ discriminative checks by default, though we ablate this choice in our experiments.

Table 1: Main results on ErdosConventionsBench (300 items, 3 convention families). DUT (Condition C) significantly outperforms engagement-matched neutral checks (B) and majority-vote baselines (A@5, B@5) on both models. Best per-column in **bold**. Alt Rate = alternate convention match rate.

Model	Condition	Overall	Asymp.	Compl.	Conv.	Alt Rate
Qwen	A (Glossary-only)	90.3	99	80	92	6.3
	B (Neutral k=3)	90.0	97	79	94	7.0
	A@5 (Majority vote)	41.3	43	6	75	–
	B@5 (Majority vote)	50.7	51	22	79	–
	C (DUT k=3)	**95.0**	**99**	**91**	**95**	**1.3**
Llama	A (Glossary-only)	56.7	74	30	66	11.0
	B (Neutral k=3)	58.7	92	21	63	9.3
	A@5 (Majority vote)	58.7	75	19	**82**	–
	B@5 (Majority vote)	60.7	88	24	70	–
	C (DUT k=3)	**81.3**	**96**	**82**	66	**2.0**

All conditions use greedy decoding (temperature=0) and the same output format tags to ensure fair comparison. The checks and main question are formatted in a chat template appropriate for each model.

4 Experiments

4.1 Experimental Setup

Benchmark. We evaluate on ErdosConventionsBench, a synthetic benchmark derived from the definitional conventions on ErdosProblems.com (Feng et al., 2026). The benchmark contains 300 items across three convention families: **asymptotics** (100 items, testing $O(\cdot)$ and $o(\cdot)$ quantifier interpretation), **completeness** (100 items, testing the definition of "complete" sequences), and **convolution** (100 items, testing additive vs. Dirichlet convolution). Each item has a deterministic ground-truth answer under the glossary-defined convention and a different answer under a common alternate convention.

Models. We evaluate two instruction-tuned models: Qwen2.5-Math-7B-Instruct (Yang et al., 2024), a math-specialized model, and Llama-3.1-8B-Instruct (Dubey et al., 2024), a general-purpose model. Both models use greedy decoding (temperature=0) with a maximum of 2048 tokens.

Metrics. We report **overall accuracy** (the fraction of items where the final answer matches the ground truth), **per-family accuracy** for each convention family, and **alternate-convention match rate** (the fraction of answers matching the alternate convention), which directly measures convention confusion.

4.2 Main Results

Table 1 presents the main results. DUT (Condition C) significantly outperforms the engagement-matched baseline (Condition B) on both models: +5.0 percentage points (pp) on Qwen (95.0% vs 90.0%, 95% CI [+2.0%, +8.3%]) and +22.7pp on Llama (81.3% vs 58.7%, 95% CI [+16.0%, +29.3%]). Critically, the neutral-checks baseline (B) performs within 2pp of the glossary-only baseline (A) on both models, establishing that simply adding extra reasoning steps does not improve convention adherence. The gains from DUT are therefore attributable to the discriminative content of the definition checks, not additional computation.

DUT also substantially outperforms inference-time scaling via majority vote (Wang et al., 2022). Single-sample DUT achieves +53.7pp over A@5 and +44.3pp over B@5 on Qwen, and +22.7pp over A@5 and +20.7pp over B@5 on Llama. This demonstrates that DUT provides a qualitatively different benefit from variance reduction through repeated sampling.

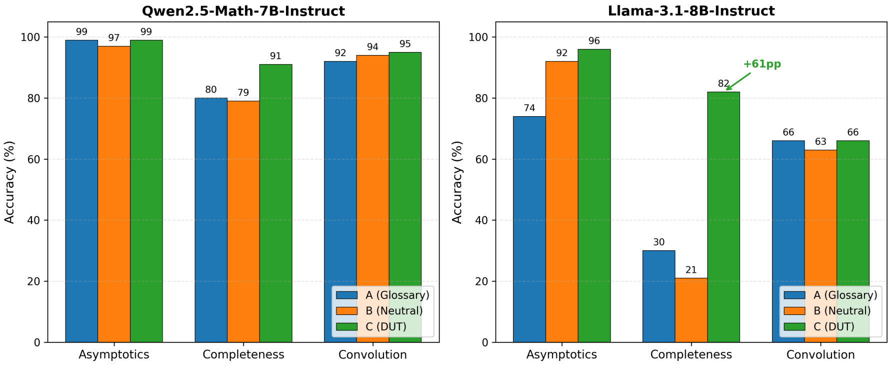

Figure 2: Per-family accuracy across conditions A (glossary-only), B (neutral checks), and C (DUT) for both models. Completeness shows the largest DUT gains (+12pp Qwen, +61pp Llama), while asymptotics is near-ceiling and convolution shows minimal improvement on Llama.

Table 2: Effect of number of discriminative checks (k). For Qwen, k=1 captures most of the DUT benefit (+2.7pp from k=1 to k=3, not significant). For Llama, k=3 is essential (+18.3pp, significant). 95% CI in brackets.

Model	Condition	Overall	Asymp.	Compl.	Conv.	Alt Rate
Qwen	A (Glossary-only)	90.3	99	80	92	6.3
	B (Neutral k=3)	90.0	97	79	94	7.0
	C (k=1)	92.3	99	88	90	2.7
	C (k=3)	**95.0**	**99**	**91**	**95**	**1.3**
Llama	A (Glossary-only)	56.7	74	30	66	11.0
	B (Neutral k=3)	58.7	92	21	63	9.3
	C (k=1)	63.0	77	44	**68**	6.3
	C (k=3)	**81.3**	**96**	**82**	66	**2.0**

The alternate-convention match rate confirms the mechanism: DUT reduces the fraction of answers matching the wrong convention from 7.0% to 1.3% on Qwen (81% relative reduction) and from 9.3% to 2.0% on Llama (78% relative reduction). Discriminative checks help the model lock onto the glossary-defined convention rather than defaulting to memorized alternatives.

4.3 Per-Family Analysis

Figure 2 shows per-family accuracy. The completeness family exhibits the largest DUT gains: +12pp on Qwen (91% vs 79%) and +61pp on Llama (82% vs 21%). This family has the highest baseline convention confusion, with alternate-convention match rates of 11–24% under conditions A and B, making discriminative checks especially valuable for disambiguation.

The asymptotics family is near-ceiling for Qwen (99% across all conditions), leaving no room for improvement. Llama shows a +4pp gain (96% vs 92%) as DUT helps resolve remaining convention confusion.

The convolution family shows minimal DUT improvement on Llama (66% for both A and C). Error analysis reveals that 29% of Llama's errors on convolution are arithmetic/reasoning errors rather than convention errors, indicating that the difficulty is computational rather than convention-related.

4.4 Effect of Number of Checks

Table 2 shows the effect of varying the number of discriminative checks. The response to k differs dramatically between models. For Qwen, k=1 captures most of the DUT benefit (92.3% vs 95.0%), with the k=3 vs k=1 difference (+2.7pp, 95% CI [-0.7%, +6.0%]) not statistically significant. For

Llama, k=3 is substantially better than k=1 (81.3% vs 63.0%), with a highly significant +18.3pp difference (95% CI [+12.0%, +24.7%]).

This model-dependent effect suggests that weaker models require more repeated definition binding to internalize the convention. The completeness family drives the k-effect on Llama, with a +38pp gap between k=1 (44%) and k=3 (82%), indicating that the hardest convention-binding tasks benefit most from multiple checks.

4.5 Error Analysis

Under DUT (Condition C), remaining errors are primarily computational rather than convention-related. On Llama's convolution family, 29% of errors are arithmetic/reasoning errors compared to only 5% convention errors. On Llama's completeness family, 18% of errors are parsing failures (the model fails to produce a parseable answer) compared to 0% convention errors. DUT effectively eliminates convention confusion, leaving computational limitations as the primary error source.

5 Conclusion

We introduced Definition Unit Tests (DUT), a training-free prompt wrapper that improves LLM convention adherence through discriminative definition checks. On ErdosConventionsBench, DUT achieves +5.0pp on Qwen and +22.7pp on Llama over engagement-matched baselines, with the engagement-matched control ruling out computation as a confound. DUT reduces alternate-convention match rates by approximately 80% and outperforms majority-vote inference-time scaling.

Our work has limitations: we evaluated on a single benchmark with three convention families, and DUT requires designing convention-specific discriminative checks. Future work could explore automated check generation from glossary definitions and extend the approach to other domains where binding to explicit specifications is critical, such as API tool-calling and evaluation rubrics.

References

Nicholas Clark, Ryan Bai, and Tanu Mitra. Offscript: Automated auditing of instruction adherence in llms. *ArXiv*, abs/2512.10172, 2025.

Abhimanyu Dubey et al. The llama 3 herd of models. 2024.

Tony Feng, Trieu H. Trinh, G. Bingham, Jiwon Kang, Shengtong Zhang, Sang hyun Kim, Kevin Barreto, Carl Schildkraut, Junehyuk Jung, Jaehyeon Seo, Carlo Pagano, Yuri Chervonyi, Dawsen Hwang, Kaiying Hou, Sergei Gukov, C. Tsai, Hyunwoo Choi, Youngbeom Jin, Weiyu Li, Hao-An Wu, Ruey-An Shiu, Yung-Sheng Shih, Quoc V. Le, and Thang Luong. Semi-autonomous mathematics discovery with gemini: A case study on the erdos problems. 2026.

Wen-Long Jin. Fastric: Prompt specification language for verifiable llm interactions. *ArXiv*, abs/2512.18940, 2025.

Soohan Lim, Joonghyuk Hahn, Hyunwoo Park, Sang-Ki Ko, and Yo-Sub Han. Contracteval: A benchmark for evaluating contract-satisfying assertions in code generation. 2025.

Hongyuan Lu, Haoyang Huang, Dongdong Zhang, Haoran Yang, Wai Lam, and Furu Wei. Chain-of-dictionary prompting elicits translation in large language models. pp. 958–976, 2023.

Xinxi Lyu, Yizhong Wang, Hanna Hajishirzi, and Pradeep Dasigi. Href: Human response-guided evaluation of instruction following in language models. *ArXiv*, abs/2412.15524, 2024.

Seyedali Mohammadi, Bhaskara Hanuma Vedula, Hemank Lamba, Edward Raff, P. Kumaraguru, Francis Ferraro, and Manas Gaur. Do llms adhere to label definitions? examining their receptivity to external label definitions. pp. 32380–32393, 2025.

Reshabh K Sharma, J. D. Halleux, Shraddha Barke, and Benjamin Zorn. Promptpex: Automatic test generation for language model prompts. *ArXiv*, abs/2503.05070, 2025.

Noah Shinn, Federico Cassano, Beck Labash, A. Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. 2023.

Xuezhi Wang, Jason Wei, D. Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2023 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1819–1862, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *ArXiv*, abs/2409.12122, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629, 2022.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *ArXiv*, abs/2311.07911, 2023.