Formula-to-World Audit: Evaluating Story-Grounded Fidelity of Arithmetic Structure in LLM-Generated Word Problems

Anonymous ACL submission

Abstract

Arithmetic word-problem evaluation often asks whether a model can solve a problem, generate a fluent problem, or state a correct solution expression. We argue that these checks miss a distinct failure mode: a model may declare the correct expression while writing a story that naturally induces a different arithmetic structure. We introduce Formula-to-World Audit, a benchmark for testing whether a symbolic arithmetic formula is stably grounded into a natural-language story world. The audit separates the target formula, the model-declared solution expression, and an independently extracted story-induced expression. Across 3,200 outputs from 100 formulas, four prompting protocols, and eight source conditions, we find that declared structure is not a reliable substitute for story-grounded structure. Critique-first prompting mainly creates a protocol-entry and length-budget burden, audit-repair recovers structured-protocol losses without significantly improving over free problem generation, and subtractive grouped templates are stable hard cases. The results suggest that structured audit-repair can optimize protocol compliance without necessarily improving the latent story commitment we ultimately care about.

1 Introduction

Large language models can solve many arithmetic word problems and can generate plausible-looking classroom exercises. Existing evaluations therefore tend to focus on answer accuracy, problem fluency, or whether the model states a syntactically correct formula. For formula-conditioned word-problem generation, however, these checks leave a gap: the central question is whether a symbolic expression is grounded into a story such that a reader would naturally write down the same expression.

This paper studies *formula-to-world grounding*: translating a symbolic arithmetic object into a natural-language story while preserving the arithmetic structure that the story itself induces. Given a target such as $(a \times b - c) \times d$, a model may declare the matching expression but write a story that implies division, a distributed expression, a collapsed bracket scope, or a different subtraction event. The declaration is then formally correct, but the story has not preserved the intended structure. We therefore separate three objects: the target formula f, the model-declared expression d, and the story-induced expression s recovered from the generated problem alone. The central failure mode is $d = f$ while $s \neq f$.

We contribute a compact audit task and a 3,200-output evaluation showing that this distinction changes the conclusions one would draw from structured prompting. Some models fail before they reach the audit gate, some declare a correct expression while writing a different story, and some fail mainly on particular expression topologies. The benchmark is not intended as a broad model leaderboard. It is a targeted behavioral test (Ribeiro et al., 2020) for symbolic commitments embedded in generated natural language.

2 Task and Benchmark

Each instance provides a target arithmetic formula and asks the model to generate a primary-school word problem in which every number and operation has a clear story-world meaning. We generate 100 formulas from five templates: $a \times b + c \times d$, $(a \times b + c \times d) \times e$, $a \times (b + c)$, $(a + b) \times (c + d)$, and $(a \times b - c) \times d$. The last template is intentionally difficult because the subtraction must be grounded as a coherent removal event before the remaining quantity is repeated.

We compare four prompting protocols. P1 is free problem generation: it asks for a word problem but does not require a declared expression. P2 asks for the declared solution expression and node-level groundings before the final problem. P3 adds a critique-first step before the final answer. P4 uses

an audit-repair loop and returns the repaired problem with structured fields. These are not treated as an ordered ladder of quality; they are source conditions that separate free generation, structured declaration, critique burden, and repair recovery.

The run matrix is 100 formulas $\times$ 4 protocols $\times$ 8 source conditions, for 3,200 outputs. The source conditions are Claude Opus 4.7, GPT-5.5, DeepSeek-V4-Pro, Gemma-4, Llama-4-Maverick, NVIDIA Nemotron 3 Super, Qwen3.6, and Talkie-1930-13B-IT. Talkie is included as a deliberately distribution-mismatched style-prior stress baseline, not as a direct ranking peer: it tests whether stylistic language generation alone produces symbolic commitment. All source conditions used temperature 0.2, top-p 1.0, fixed seeds where supported, and protocol-specific maximum generation limits sufficient for the required structured fields. Provider or transport failures were retried before aggregation; remaining failures are output-format, parser, or story-audit failures. All reported rates use the planned source-run denominator unless explicitly described otherwise.

3 Audit Method

For each output we parse the model response, recover the visible word problem, and run an independent story-induced extractor. The extractor receives only the generated problem and final question; it does not receive the target formula, declared expression, model identity, or node groundings. It returns an inferred expression s, evidence spans, confidence, and notes. The primary extractor is GPT-5.5, with a Claude Opus 4.7 second-extractor check described in Section 5.

The main metric compares s to the target formula f under relaxed AST equality, which permits commutative swaps but not distributive rewrites. We compute it over the full planned run denominator:

$$\mathrm{StoryFullRun} = \frac{|\mathcal{M}|}{N_{\mathrm{source}}}.$$

$$\mathcal{M} = \{i \mid \mathrm{parse}(s_i) \wedge s_i \equiv_{\mathrm{rel}} f_i\}.$$

Thus a source parse failure and an extractor-stage failure both count as deployment failures. This makes StoryFullRun an end-to-end audit reliability measure: it asks whether a planned run yields a parseable story whose induced structure matches the target, not whether every failed run is a semantic mismatch. We therefore interpret it together with parse compliance, story auditability, out-of-grammar induced expressions, and declared/story disagreement. Declared/story disagreement is defined only when a protocol supplies a declared expression: d matches f, but s does not. P1 has no declared expression and is excluded from this diagnostic. Surface world plausibility is retained only as a v0 diagnostic; claims about formula-to-world fidelity use story-induced recovery. Full prompts, schemas, parser rules, and anonymized audit records are included in the supplementary artifact. The benchmark formulas, prompts, parser code, schemas, and audit records are released for research and reproducibility, not as a primary-school assessment product or an automatic grading tool without human review. Model weights, provider credentials, server addresses, and provider response identifiers are not redistributed; released generated outputs and derived audit labels should be used subject to applicable provider and model-license terms.

4 Results

Table 1 reports model-level behavior. The key column is Story, the full-run story-induced relaxed AST metric. Audit is full-run story auditability, OOG is the rate of inferred story expressions outside the current grammar, and Decl./story is available-case declared/story disagreement. All numeric entries are percentages.

Declared formula match is not story-induced fidelity. The benchmark exposes cases where a model declares a correct-looking solution expression while the story induces another expression. This is most visible for Llama-4-Maverick and Gemma-4, whose declared/story disagreement rates are 18.4% and 11.3%. These failures would be hidden by a declared-expression audit that checks only whether the model copied or reconstructed the target formula. In a representative case, a model declared $(2 + 7) \times (7 + 8)$ but described boxes, packs, and pencils in a nested relation recovered as $(2 + 7) \times 7 \times 8$. The text was fluent; the story-world roles did not implement the intended product of two sums.

Critique-first creates protocol-entry burden. P3 critique-first has the lowest full-run story score among protocols. Figure 1 shows that P3 reaches only 55.1% story relaxed full-run, compared with 74.0% for P1 and 74.4% for P4. The paired P1

Model	Parse	Audit	Story	OOG	Decl./story
Claude Opus 4.7	100.0	99.0	97.5	1.0	0.9
GPT-5.5	97.0	96.8	96.5	0.0	0.0
NVIDIA Nemotron 3 Super	79.0	77.5	76.0	0.6	2.4
DeepSeek-V4-Pro	76.2	76.0	73.2	0.3	4.2
Llama-4-Maverick	95.5	89.0	72.0	6.5	18.4
Gemma-4	90.5	80.8	69.0	10.2	11.3
Qwen3.6	64.5	64.2	62.5	0.4	3.5
Talkie-1930-13B-IT*	25.0	17.2	0.0	23.0	NA

Table 1: Model-level Formula-to-World results. Talkie* is a deliberately distribution-mismatched style-prior stress baseline, not a direct ranking peer. Story is `story_relaxed_full_run_penalized`.

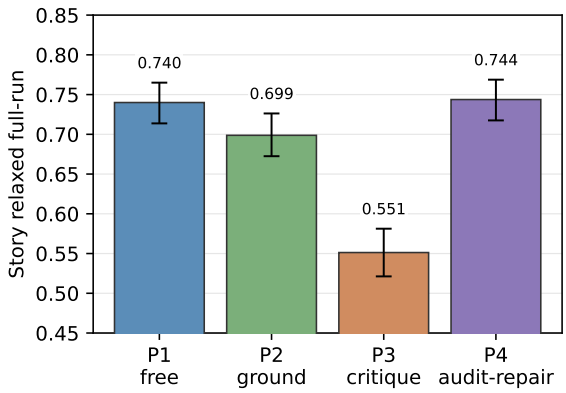

Figure 1: Protocol comparison. Bars show full-run story-induced structure fidelity; error bars are 95% bootstrap confidence intervals over formula IDs.

Protocol	Story full-run	95% CI
P1 free	74.0	[71.4, 76.5]
P2 grounding-first	69.9	[67.2, 72.6]
P3 critique-first	55.1	[52.1, 58.1]
P4 audit-repair	74.4	[71.8, 76.9]

Comparison	Diff.	95% CI
P1 → P4	0.4	[-2.2, 3.0]
P2 → P4	4.5	[1.4, 7.5]
P3 → P4	19.2	[15.5, 23.0]

Table 2: Protocol-level story-induced fidelity and paired comparisons against P4. Story full-run values and differences are percentages; intervals are 95% bootstrap CIs over formula IDs.

→ P3 difference is -18.9 points with a bootstrap interval that does not cross zero. The failure is not primarily that surviving P3 stories are implausible. Many outputs fail before they reach the story-audit gate, so the protocol loses reliability through format and auditability collapse. Appendix Table 4 shows that 40.1% of P3 runs fail at the source parse gate, while only 3.6% are auditable semantic mismatches.

Audit-repair is recovery, not universal improvement. P4 audit-repair clearly recovers losses introduced by structured protocols. The paired P2 → P4 difference is +4.5 points, 95% CI [1.4, 7.5], and the P3 → P4 difference is +19.2 points, 95% CI [15.5, 23.0]. However, P4 is not meaningfully above free problem generation: the paired P1 → P4 difference is only +0.4 points, 95% CI [-2.2, 3.0]. Thus audit-repair improves controllability among structured protocols, but should not be framed as a universal improvement over free generation. This raises a methodological question for structured prompting: if audit-repair only matches free generation on substantive story fidelity while costing more inference, the protocol may be optimizing format compliance rather than the latent symbolic commitment we ultimately care about.

Expression topology matters. Template E, $(a \times b - c) \times d$, is a stable hard case. Its story relaxed full-run score is 54.1%, with 95% CI [51.9, 56.1], far below templates A and C. It also has the highest out-of-grammar rate and declared/story disagreement rate. Subtractive grouped expressions require a coherent removal event and preservation of bracket scope before repetition; models often drift into alternative story structures. One common drift is operation intrusion: a target such as $(6 \times 2 - 5) \times 9$ becomes a story about giving the remaining pencils equally to 9 students, inducing $((6 \times 2) - 5)/9$ rather than a repeated difference. Such examples are not mere parser failures. They show that a plau-

sible classroom story can introduce an operation outside the intended grammar.

A distribution-mismatched control separates style from symbolic commitment. Talkie reaches parse compliance only on P1, has full-run story auditability of 17.2%, and has 0.0% story relaxed full-run success. This does not make Talkie a conventional "weak model" in the ranking sense. Its role is to show that stylistic generation and symbolic commitment are separable: a model can produce language in a recognizable style while failing to maintain the arithmetic contract required by the benchmark.

5 Validity Checks

The primary extractor is GPT-5.5, which is also an evaluated source model, creating an LLM-as-judge self-evaluation risk (Zheng et al., 2023). We therefore reran all 388 parseable GPT-5.5 source rows with Claude Opus 4.7. Across 269 declared-comparable rows, both extractors found 0.0% declared/story disagreement, and they agreed on relaxed story match for 99.7% of comparable rows. A smaller three-extractor subset with GPT-5.5, Claude Opus 4.7, and DeepSeek-V4-Pro also found 0.0% declared/story disagreement on 34 declared-comparable rows. This does not remove all extractor-bias risk, but it addresses the most sensitive case: GPT-5.5's zero disagreement is reproduced by an out-of-family extractor over the full GPT-5.5 story scope.

We also completed a 100-row stratified human annotation sample covering P3 parse failures, Llama/Gemma declared-story disagreements, Template E disagreements, out-of-grammar rows, Talkie negative-control rows, and high-scoring GPT/Claude calibration rows. One author annotated the sample using the same story-induced expression criteria described in Section 3; no crowdworkers or external human subjects were recruited. Against the automatic extractor, the human labels agree on story-induced match for 95.9% of binary-comparable rows ($n = 49$) and on declared/story disagreement for 94.3% ($n = 35$). Auditability agreement is lower, 67.4% ($n = 95$), because humans can sometimes recover a valid story from raw outputs that fail the structured parser. This is consistent with the main structural diagnostics while confirming that protocol-entry failures and human auditability should be analyzed separately. The lower auditability agreement indicates that the full-run metric is conservative: it treats parser- and protocol-entry failures as deployment failures even when a human can sometimes recover a story-level interpretation.

6 Related Work

Arithmetic word problems have been studied as reasoning and semantic parsing tasks, where text is mapped to equations or executable structures (Kushman et al., 2014; Koncel-Kedziorski et al., 2015), and recent benchmarks evaluate whether models can solve given problems (Cobbe et al., 2021; Hendrycks et al., 2021). Formula-to-World Audit reverses the direction: the formula is given first, and the model must produce a story whose own arithmetic commitments recover it. This connects to controlled word-problem generation (Zhou and Huang, 2019), but our metric is not only fluency or whether the source equation is stated.

The audit also connects to faithfulness and grounded generation. Work on factuality shows that fluent generated text can be unsupported or inconsistent with its source (Maynez et al., 2020; Honovich et al., 2022). Here the source is a formula rather than a document, and unfaithfulness appears as a different story-induced expression. Prompting and revision methods such as chain-of-thought and self-refinement (Wei et al., 2022; Madaan et al., 2023) motivate our structured protocols, but our results caution that critique and repair can improve protocol compliance without improving the underlying commitment. Because the story extractor is LLM-based, the work also follows LLM-as-judge concerns (Zheng et al., 2023); we narrow the judge role to expression recovery and use symbolic AST comparison for the final decision.

7 Conclusion

Formula-to-World Audit tests whether models can ground arithmetic formulas into stories whose implied structure remains faithful to the target. Across 3,200 outputs, it exposes declared/story disagreement, protocol-entry failures, audit-repair recovery, template-sensitive drift, and a negative-control split between style and symbolic commitment. The benchmark shows that declared correctness is not the same as story-grounded correctness.

Limitations

The story-induced extractor is an LLM auditor, and the current human validation uses one anno-

tator over a stratified 100-row sample rather than a full population annotation with inter-annotator agreement. The extractor sanity checks and human calibration reduce the most direct self-evaluation concern but do not provide a full precision/recall estimate for every failure type. The formula grammar is small and primary-school oriented; out-of-grammar inferred expressions may include parse artifacts. The benchmark covers English word problems only. World plausibility is a surface proxy, not a complete naturalness measure. Talkie is a stress baseline, not a directly comparable frontier-model condition.

Ethical Considerations

This work evaluates synthetic arithmetic formulas and model-generated word problems. It does not use personal data, private user data, social-media data, clinical data, or high-stakes decision making. The benchmark is intended for research diagnostics of formula-to-story fidelity, not for automated educational assessment or student-facing deployment without human review. The human calibration sample was annotated by one author; no crowdworkers or external human subjects were recruited. Artifacts are released only for research and reproducibility and should be reused according to the accompanying licenses and applicable third-party model/provider terms.

AI Assistance Disclosure

AI assistants were used for drafting, editing, coding/debugging support, and experimental assistance. All experimental design choices, claims, reported results, and final submission text were reviewed and controlled by the author(s). The evaluated LLM source conditions and the LLM-based story-induced extractor are part of the experimental design and are described as artifacts, not authors.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 271–281.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Qingyu Zhou and Danqing Huang. 2019. Towards generating math word problems from equations and topics. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 494–503.

A Artifacts, Model Sources, and Access Terms

Table 3 lists the exact source conditions used in the reported run. Seven hosted source conditions were accessed through OpenRouter-compatible model IDs[1] on 3 May 2026; Talkie was served through a local OpenAI-compatible endpoint as a distribution-mismatched stress baseline. Because stable public technical-report citations or model cards were not available for every hosted snapshot at submission time, we report exact provider IDs, access route, and access date as source metadata. The supplementary artifact includes prompts, schemas, parser rules, source-condition configuration, anonymized audit records, aggregate reports, and annotation materials.

B Failure Decomposition

Table 4 decomposes the full-run story metric into mutually exclusive outcomes over the 800 planned source runs for each protocol. The P3 drop is concentrated at the source parse gate rather than among auditable story expressions: only 3.6% of P3 runs are auditable semantic mismatches, while 40.1% fail before the story audit can be applied. This decomposition identifies where failures enter the pipeline; it does not by itself prove whether P3 parse failures are caused by output length, truncation, or schema drift. For protocol-level decomposition, OOG is grouped under audit failure because it is not relaxed-comparable to the target grammar; Table 7 separates OOG for template-level diagnosis. The raw outputs and protocol-specific generation limits are included in the supplementary artifact for follow-up analysis.

Table 5 gives a lightweight length diagnostic for the seven provider-logged API source conditions; Talkie is excluded because its local run did not expose comparable token accounting. P3 is substantially longer and has more provider length stops than the other protocols, so we treat its parse collapse as a protocol-entry and length-budget burden rather than as direct evidence of semantic critique failure.

C Template Robustness

Table 7 shows that Template E, $(a \times b - c) \times d$, is not only lower in aggregate story fidelity but also has the largest story-mismatch and out-of-grammar rates. The dominant qualitative drifts in inspected E cases are division-style sharing or packing after subtraction, dropped outer repetition, and bracket-scope collapse where the removal event no longer applies before the final scaling.

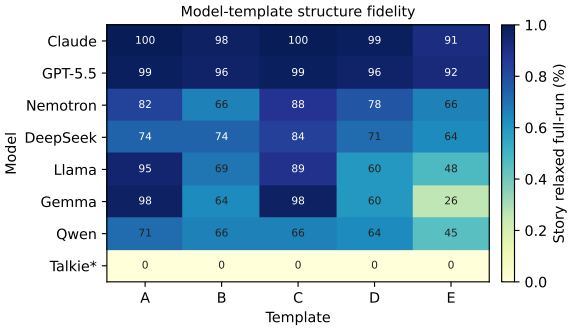

Figure 2: Model-template heatmap for full-run story-induced structure fidelity. Template E is the clearest hard case; Talkie* is the style-prior stress baseline.

D Extractor Sanity Check

The failure-heavy sanity sample contains 102 extractable rows from an earlier exploratory sanity set, separate from the final 100-row human annotation sample reported in Section 5. Claude Opus 4.7 returned parseable extractor JSON for all 102 rows. On the full GPT-5.5 check, Claude returned parseable extractor JSON and story-auditable expressions for all 388 parse-ok GPT-5.5 story rows. A partial DeepSeek-V4-Pro third-extractor check was stopped at 51 rows; 46 were all-auditable across all three extractors and 34 were declared-comparable across all three. The second- and third-extractor checks are not substitutes for human annotation, but they address the most direct concern that GPT-5.5 source outputs might receive zero declared/story disagreement only because the primary extractor is GPT-5.5.

E Human-Extractor Agreement

[1] https://openrouter.ai/models; API documentation: https://openrouter.ai/docs.

Source condition	Access route	Reported ID	Artifact/access note
Claude Opus 4.7	OpenRouter API	`anthropic/claude-opus-4.7`	Hosted source condition and second extractor; no weights or provider response identifiers redistributed.
GPT-5.5	OpenRouter API	`openai/gpt-5.5`	Hosted source condition and primary story extractor; full GPT-5.5 source scope was rerun with Claude for sanity checking.
DeepSeek-V4-Pro	OpenRouter API	`deepseek/deepseek-v4-pro`	Hosted source condition; also used for a small third-extractor sanity subset.
Gemma-4	OpenRouter API	`google/gemma-4-26b-a4b-it`	Hosted source condition; model weights are not included in the artifact.
Llama-4-Maverick	OpenRouter API	`meta-llama/llama-4-maverick`	Hosted source condition; model weights are not included in the artifact.
NVIDIA Nemotron 3 Super	OpenRouter API	`nvidia/nemotron-3-super-120b-a12b`	Hosted source condition; model weights are not included in the artifact.
Qwen3.6	OpenRouter API	`qwen/qwen3.6-35b-a3b`	Hosted source condition; model weights are not included in the artifact.
Talkie-1930-13B-IT	Local OpenAI-compatible endpoint	`talkie-1930-13b-it`	Distribution-mismatched style-prior stress baseline served on the authors' infrastructure; endpoint credentials and weights are not redistributed.

Table 3: Model/source-condition access metadata for the reported 3,200-run experiment. Access date: 3 May 2026. Hosted conditions were accessed through OpenRouter model IDs; users rerunning or redistributing additional outputs should follow applicable provider and model-license terms.

Protocol	Success	Parse fail	Audit fail	Semantic mismatch
P1 free	74.0	2.2	9.1	14.6
P2 ground	69.9	23.4	1.8	5.0
P3 critique	55.1	40.1	1.1	3.6
P4 audit-repair	74.4	20.4	1.6	3.6

Table 4: Protocol failure decomposition over the full source-run denominator. Success is story-induced relaxed AST match. Audit fail means source parse succeeded but story-induced auditability failed, including out-of-grammar inferred expressions. At the protocol level, OOG cases are grouped under audit failure; Table 7 separates them for template-level diagnosis. Semantic mismatch means the story-induced expression was auditable but not relaxed-equivalent to the target. Values are percentages.

Protocol	Token-log n	Mean comp. tokens	Mean visible chars	Length stop
P1 free	700	560	231	3.3
P2 ground	700	1104	1476	9.3
P3 critique	700	1984	3319	30.0
P4 audit-repair	700	929	908	3.9

Table 5: Protocol-level length diagnostic over source conditions with provider token logs. Completion tokens are provider-reported and may include provider-side accounting differences. Visible chars are measured on the returned raw response text, not provider token accounting. Length stop is the percentage of runs whose provider finish reason was a token-limit stop.

Template	Pattern	Story	95% CI	OOG	Decl./story
A	$a \times b + c \times d$	77.3	[75.8, 79.1]	1.3	0.9
B	$(a \times b + c \times d) \times e$	66.6	[64.1, 69.1]	5.4	5.7
C	$a \times (b + c)$	77.8	[76.2, 79.5]	0.9	2.6
D	$(a + b) \times (c + d)$	65.9	[63.0, 68.8]	2.0	7.4
E	$(a \times b - c) \times d$	54.1	[51.9, 56.1]	9.4	16.6

Table 6: Template-level story-induced metrics. All numeric entries are percentages.

Template	Success	Parse fail	Story mismatch	OOG	Other unusable
A	77.3	18.9	2.3	1.1	0.3
B	66.6	22.5	6.1	4.2	0.6
C	77.8	17.2	3.9	0.8	0.3
D	65.9	22.5	9.2	1.6	0.8
E	54.1	26.6	12.0	6.9	0.5

Table 7: Template-level error taxonomy over the full 640 source runs per template. Categories are mutually exclusive. OOG denotes story-induced expressions outside the target grammar, most often division or other operation intrusion. Values are percentages.

Human label	n	Human positive	Extractor positive	Agreement
Auditability	95	73.7	47.4	67.4
Story match	49	30.6	26.5	95.9
Declared/story disagreement	35	77.1	82.9	94.3

Table 8: Human/extractor agreement on the stratified 100-row annotation sample. Agreement is computed only on rows with completed human labels for the corresponding diagnostic. Positive rates are not population prevalence estimates because the sample is stratified toward boundary and failure cases.