DecomWriting-Judge: Agentic Evaluation of Long-Form Writing with Adversarial Logic Verification

Anonymous ACL submission

Abstract

Evaluating long-form generation remains a critical bottleneck, as single-pass LLM judges suffer from length bias, shallow evaluation, and the "lost-in-the-middle" phenomenon. While the Agent-as-a-Judge paradigm has shown promise in addressing these limitations in code generation scenarios, its application to long-form writing remains unexplored. To bridge this gap, we propose **DecomWriting-Judge**, an agentic framework that evaluates long-form writing through multi-stage decomposition, checklist-grounded shallow scoring, and adversarial deep logic verification. We further introduce the **Logic Reward Benchmark**, a 191-sample preference dataset for evaluating logical consistency with targeted perturbations. DecomWriting-Judge consistently outperforms all single-pass baselines on the Logic Reward Benchmark and Long-form RewardBench, with DeepSeek-V4-Flash improving from 44.50% to 65.45% accuracy when embedded in the agentic framework. The Shallow-Deep dimensional decomposition reveals that existing judges can obscure shallow-deep trade-offs and structural constraint failures. Each evaluation costs approximately $0.008, a small amount compared to expert human evaluation. Our code is available at `https://anonymous.4open.science/r/DecomWriting-Judge`.

1 Introduction

While long-context large language models (LLMs) have advanced rapidly, generating high-quality long-form content remains challenging because models must preserve coherence and satisfy multiple constraints over long contexts (Wan et al., 2025; Xiao et al., 2025). As outputs grow longer and more complex, reliable evaluation has also become a major bottleneck. The dominant LLM-as-a-Judge paradigm relies on single-pass models, which suffer from three fundamental limitations in long-form evaluation. First, they exhibit a tendency to favor longer responses (Hu et al., 2025). Second, they perform shallow evaluations that reward surface fluency over deep logic (You et al., 2026). Finally, the "lost-in-the-middle" phenomenon causes them to struggle with strict constraints over extended contexts (Liu et al., 2024a; An et al., 2024). This evaluation limitations is deeply intertwined with the challenges of generation, as both tasks share a common requirement: preserving long-range constraints and global consistency over extended contexts. Consequently, advancements in long-form generation may offer insights for evaluation.

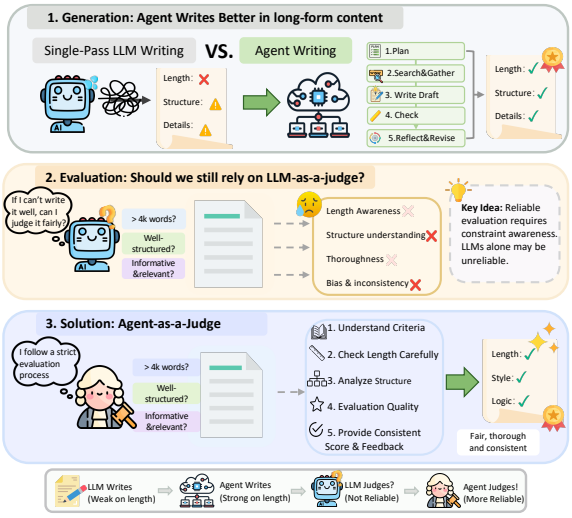

Figure 1: The performance gap between agentic and single-pass generation motivates a parallel shift from LLM judges to Agent judges.

Recently, the Agent-as-a-Judge paradigm has been applied to domains such as code generation to address these evaluation limitations through iterative planning and tool use (Zhuge et al., 2025). However, its application remains unexplored in the long-form writing domain, where complex logical dependencies and structural constraints are difficult to evaluate across long context. To investigate whether agentic structures inherently excel at han-

dling these constraints over long contexts, we first conduct a preliminary study on long-form writing generation. Our findings reveal that agentic writing systems (e.g., WriteHERE (Xiong et al., 2025) and DeepResearch) consistently outperform vanilla LLMs by over 20 points. This success demonstrates that structural decomposition and iterative reflection effectively handles long-form constraints, raising a pivotal question: *can this architectural strength transfer to evaluation?*

Motivated by this performance gap, we propose the **DecomWriting-Judge** framework, which shifts evaluation from single-pass LLM judges to agentic evaluators that decompose the task and verify results, as illustrated in Figure 3. The framework operates across four stages. Stage 1 generates query-specific checklists for style and logic. Stage 2 scores shallow dimensions of length adherence and stylistic quality via direct computation and rubric-guided assessment. Stage 3 conducts an adversarial deep logic audit through evidence-grounded verification. Stage 4 aggregates all dimension scores into an interpretable breakdown. To validate the fidelity of this adversarial logic audit, we introduce the **Logic Reward Benchmark**, comprising 191 preference pairs. We constructed this dataset by injecting five types of targeted logical perturbations into real-world passages. Extensive experiments on the Logic Reward Benchmark and Long-form RewardBench (Huang et al., 2026) demonstrate that DecomWriting-Judge consistently outperforms all single-pass LLM baselines, achieving the highest accuracy across both datasets. By controlling for the base model to isolate its inherent capabilities, we reveal the pure impact of the paradigm shift, exemplified by DeepSeek-V4-Flash improving from 44.50% to 65.45% accuracy on the Logic Reward Benchmark. When applied to real-world long-form evaluation, our framework further reveals shallow-deep trade-offs and structural constraint failures that are often obscured by single-pass evaluation. Finally, each evaluation costs only about $0.008, far below expert human evaluation. Our contributions are as follows:

- We propose **DecomWriting-Judge**, the first agent-as-a-judge framework for long-form writing, enabling multi-dimensional evaluation via multi-stage decomposition and adversarial logic verification.
- We construct the **Logic Reward Benchmark**, a preference dataset with targeted logical perturbations that serves as a controlled testbed for measuring logical consistency evaluation.
- We reveal that single-pass LLM judges obscures shallow-deep trade-offs and structural constraint failures in long-form writing.

2 Preliminary: Bridging Generation and Evaluation

Long-form generation and long-form evaluation are fundamentally intertwined, as both share a common requirement: preserving long-range constraints and global consistency over extended contexts. This underlying symmetry inspires a compelling hypothesis: if agentic architectures inherently write better than single-pass LLMs, can they similarly be leveraged to deliver stronger long-form evaluators? To investigate this possibility, we first evaluate agentic architectures on long-form writing tasks, aiming to provide insights that justify transitioning these structural strengths to evaluation.

Experimental Setup. Evaluation is conducted on two benchmarks that evaluate writing quality and length adherence:

- **WritingBench** (Wu et al., 2025d) covers 6 domains and 100 subdomains. For each query, a fine-tuned critic model scores five dynamically generated criteria on a 1–10 scale. The length-specific score *Length_C* is reported.
- **LongBench-Write** (Bai et al., 2025) features 120 diverse long-form queries with word-count targets across four length tiers (0–500, 500–2k, 2k–4k, >4k words). Its evaluation framework scores outputs on two axes: a Quality Score S_q rated via LLM-as-a-Judge on six dimensions, and a Length Score S_l:

$$S_l = \begin{cases} 100 \cdot \max(0, 1 - (l'/l - 1)/3) & \text{if } l' > l \\ 100 \cdot \max(0, 1 - (l/l' - 1)/2) & \text{if } l' \leq l \end{cases},$$

  where l' is the actual word count and l is the target. GPT-4o-mini is used as the judge for S_q, and the average score on the >4k subset is reported.

Evaluated Generators. Generators from two paradigms are compared: vanilla single-pass LLMs and agentic writing frameworks.

- **Vanilla LLMs.** Four single-pass models are evaluated: DeepSeek-V4-flash (DeepSeek-AI, 2026), Gemini-3-Flash (Google DeepMind, 2025), and

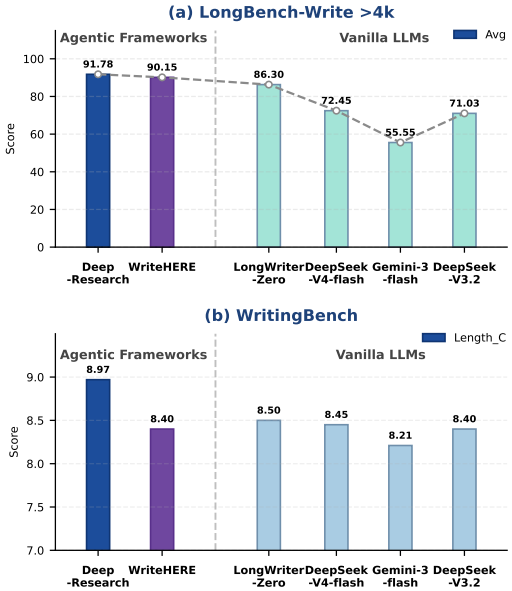

Figure 2: Agentic frameworks consistently outperform vanilla LLMs on LongBench-Write (a) and WritingBench (b).

DeepSeek-V3.2 (Liu et al., 2025a), along with LongWriter-Zero (Wu et al., 2025b), a model fine-tuned for long-form generation.

- **Agentic Frameworks.** Two agentic frameworks are evaluated, both configured with DeepSeek-V3.2 as the base LLM. WriteHERE (Xiong et al., 2025) recursively decomposes writing goals into a DAG of sub-tasks with interleaved planning and execution. DeepResearch, a custom-built agentic baseline, iteratively gathers information through search and reflection, then generates the draft under a length-scoring feedback loop, as detailed in Appendix A.

Results Analysis. The experimental results are summarized in Figure 2.

- **Agentic Systems Outperform Vanilla LLMs.** Both agentic frameworks dominate vanilla LLMs by a wide margin. On LongBench-Write, DeepResearch and WriteHERE achieve average scores of 91.78 and 90.15, while the best vanilla model trails substantially. DeepSeek-V3.2 alone scores 71.03, but reaches 91.78 when embedded in an agentic workflow, a 20-point gain driven entirely by architecture rather than model scale. LongWriter-Zero, fine-tuned for long-form generation, attains 86.30, yet this requires substantial fine-tuning compute, making it less practical than training-free agentic frameworks.

- **Decomposition and Reflection Improve Constraint Adherence.** This gap stems from a fundamental architectural distinction: single-pass LLMs process the entire task in one step, while agentic systems decompose it into finer-grained sub-tasks and iteratively reflect to catch constraint violations before they propagate.

Conclusion: From Generation to Evaluation. In summary, agentic architectures significantly outperform single-pass models in constraint adherence. This success motivates a natural extension: can agentic workflows similarly excel in evaluation, where constraint awareness is even more critical? We introduce the **DecomWriting-Judge** framework to investigate this hypothesis.

3 Method

To investigate whether agentic workflows can improve constraint-aware evaluation, this section presents DecomWriting-Judge, a framework that replaces single-pass scoring with an agentic, four-stage workflow of checklist grounding and adversarial logic verification. To evaluate its performance, we also introduce the Logic Reward Benchmark for validating logical fidelity.

3.1 DecomWriting-Judge Framework

The framework evaluates model-generated text against a query through four stages spanning three dimensions.

Stage 1: Multi-Dimensional Checklist Generation. Checklist generation anchors each dimension to query-specific criteria before scoring begins, preventing cross-dimension contamination in later stages. To this end, three agents operate in parallel to anchor each dimension to query-specific criteria before scoring begins. The Length agent extracts a target length l from the query. The Style agent instantiates four predefined style rules (writing style, rhetorical devices, query alignment, and vocabulary) into a checklist $\mathcal{C}_{\text{style}}$ with a 6-band rubric. The Logic agent instantiates five logic rules (causal, temporal, entity, state transition, and global coherence) into a checklist $\mathcal{C}_{\text{logic}}$ of verifiable check items. Full rule definitions and a complete checklist example are provided in Appendix B and C.

Stage 2: Shallow Dimension Scoring. Stage 2 handles length and style as shallow dimensions via independent scorers, since neither relies on complex reasoning.

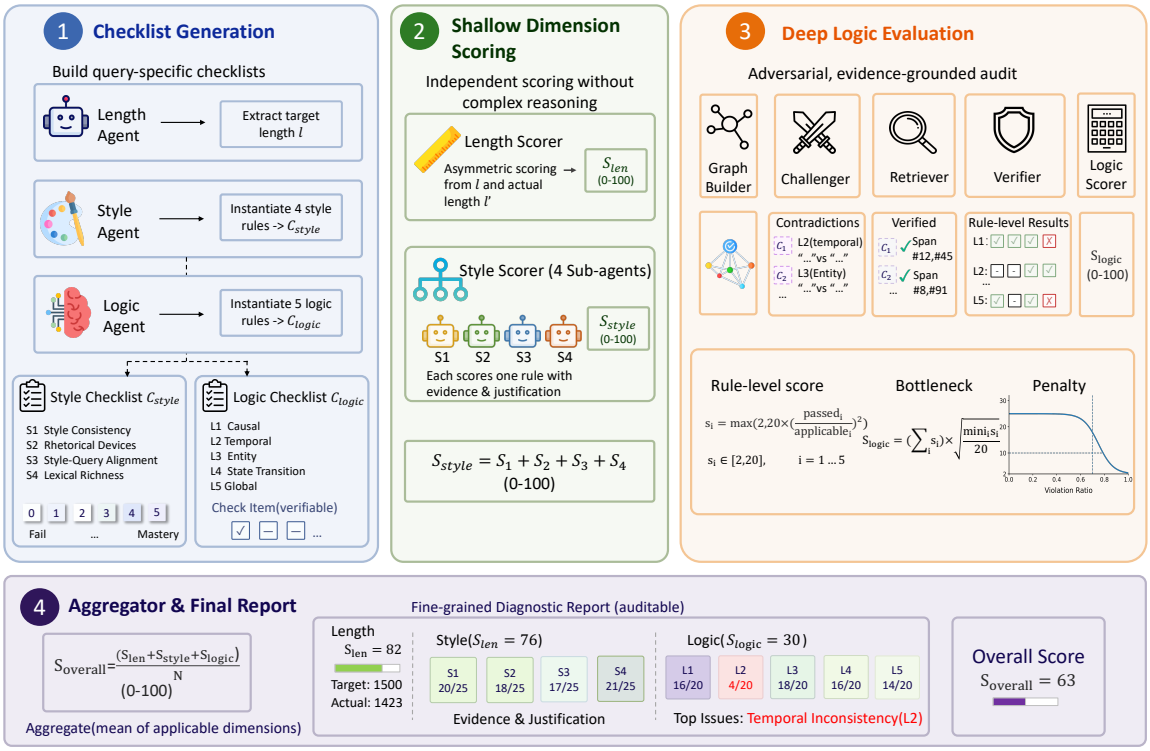

Figure 3: Overview of the DecomWriting-Judge framework.

- **Length Scorer.** The score is computed from the target length l obtained from Stage 1 and the actual length l' using the asymmetric piecewise function S_l introduced in Section 2, adopted from LongBench-Write.
- **Style Scorer.** Given $\mathcal{C}_{\text{style}}$ from Stage 1, four sub-agents operate in parallel, each scoring one of the four style rules (S1–S4) against the generated text using a 6-band rubric. Each sub-agent returns a score on a 0–25 scale with evidence. The style score is the sum of the four rule-level scores:

$$S_{\text{style}} = S_1 + S_2 + S_3 + S_4,$$

A critical constraint is *dimensional independence*: the Style Scorer evaluates stylistic quality exclusively and does not penalize defects attributable to length or logic.

Stage 3: Deep Logic Evaluation. Logical contradictions are sparse and span non-adjacent passages, making them undetectable by surface-level scoring. Stage 3 therefore conducts an adversarial, evidence-grounded audit built upon $\mathcal{C}_{\text{logic}}$ from Stage 1. The pipeline proceeds through graph construction, adversarial challenge, evidence retrieval, contradiction verification, and structured scoring.

- **Graph Builder.** A logical relationship graph is constructed to model the text's internal structure. Nodes represent entities, events, and states, while directed edges encode three relation types: *causes*, *precedes*, and *relates_to*. This graph, denoted $\mathcal{G}$, enables downstream agents to reason about cross-sentence logical dependencies beyond the original linear order.
- **Challenger and Retriever.** The Challenger agent searches the generated text for a set of contradictions $\mathcal{D}$, using $\mathcal{G}$ and $\mathcal{C}_{\text{logic}}$ to guide its analysis. Each contradiction identifies the violated logic rule, describes the inconsistency, and cites verifiable verbatim evidence from the text to prevent hallucination. The Retriever then independently validates every evidence claim by locating it in the source text, yielding the verified subset $\mathcal{D}^* \subseteq \mathcal{D}$.
- **Verifier.** For each contradiction in $\mathcal{D}^*$, the Verifier maps it to the relevant check items under the violated logic rule and judges, for each item, whether the contradiction indeed constitutes a violation of that item. Each check item is marked as passed or failed, accompanied by evidence references and a brief reasoning.
- **Logic Scorer.** The logic score is computed in two steps. First, each rule i receives a rule-level score s_i via quadratic penalization, which accelerates the penalty as more check items fail, with a floor

of 2 to prevent a single failure from zeroing out a sparsely checked rule:

$$s_i = \max\left(2,\ 20 \times \left(\frac{\text{passed}_i}{\text{applicable}_i}\right)^2\right),$$

where passed_i and applicable_i denote passed and applicable check items. Second, the sum of s_i is scaled by a bottleneck factor $\sqrt{\min_i s_i/20}$ that bounds the overall score by the worst-performing rule, so that a catastrophic failure cannot be compensated by strong performance elsewhere:

$$S_{\text{logic}} = \left(\sum_i s_i\right) \times \sqrt{\frac{\min_i s_i}{20}},$$

Stage 4: Aggregation. The final stage aggregates the overall score and the breakdown per-dimension. The overall score is the arithmetic mean of the applicable dimensions:

$$S_{\text{overall}} = \frac{S_{\text{len}} + S_{\text{style}} + S_{\text{logic}}}{N},$$

where N is the number of dimensions applicable. The output further includes per-rule sub-scores for length, style and logic, enabling fine-grained analysis of generator behavior.

3.2 Logic Reward Benchmark

Task and Data Format. To evaluate logical discrimination, we construct a preference dataset where each sample is a pair consisting of a coherent original passage (positive instance) and a variant with targeted logical perturbations (negative instance). By holding all surface-level properties constant, we isolate logic as the sole evaluation dimension. An evaluation is deemed correct if the positive instance receives a higher score than its negative counterpart.

Data Sources. Source texts are drawn from four QA subsets of the LongBench benchmark (Bai et al., 2024): multifieldqa_zh, qasper_e, multifieldqa_en_e, and narrativeqa. Each entry in these subsets consists of a context passage, a question, and an answer. The original context serves directly as the positive sample.

Perturbation Construction. The negative instance is constructed through three steps, as illustrated in Figure 4. Given the answer, the evidence sentences supporting it in the context are first identified by DeepSeek-V3.2. One of the five logic perturbation types is then randomly selected and applied to these sentences, each directly corresponding to a logic rule in Stage 3, ensuring that the dataset directly tests the framework's capacity to detect the same logical flaws. Finally, each perturbed passage is jointly filtered by Qwen3.5-Plus (Qwen Team, 2026), DeepSeek-V4-Flash, and human annotators to discard cases where the perturbation is ineffective or introduces unrelated artifacts. The resulting dataset contains 191 samples. Detailed data statistics are provided in Appendix D.

4 Experiments

4.1 Experimental Setup

Benchmarks. Two benchmarks are employed. The first is Long-form RewardBench (Huang et al., 2026), a preference dataset consisting of chosen and rejected responses, from which we adopt the Writing subset. For each preference pair, the evaluator assigns scores S_{chosen} and S_{rejected} to the two responses, a judgment is considered correct when

$$S_{\text{chosen}} - S_{\text{rejected}} > \text{diff},$$

where diff is a threshold hyperparameter. The second benchmark is the Logic Reward Benchmark introduced in Section 3.2, which follows the same protocol but additionally requires the evaluator to locate the specific perturbed passage for each sample, enabling fine-grained precision measurement. Implementation details are in Appendix E.

Baselines. Three groups of evaluators compared:

- **Agentic Systems.** We deploy the proposed framework with varying base LLMs, including Doubao-Seed-1.8 (Seed, 2026), DeepSeek-V4-Flash, and Qwen3.5-35B-A3B (Qwen Team, 2026).
- **Proprietary LLMs.** These include Doubao-Seed-1.8, Gemini-3-Flash, GPT-5-mini (Singh et al., 2025), and Qwen3.5-Plus, all used as single-pass scorers.
- **Open-Source LLMs.** This category comprises DeepSeek-V4-Flash, DeepSeek-R1 (Guo et al., 2025), DeepSeek-V3.2, Qwen3-235B-A22B-Instruct (Yang et al., 2025), and Qwen3.5-35B-A3B, all deployed as single-pass scorers.

All LLMs baselines are required to produce the same multi-dimensional score in a single pass.

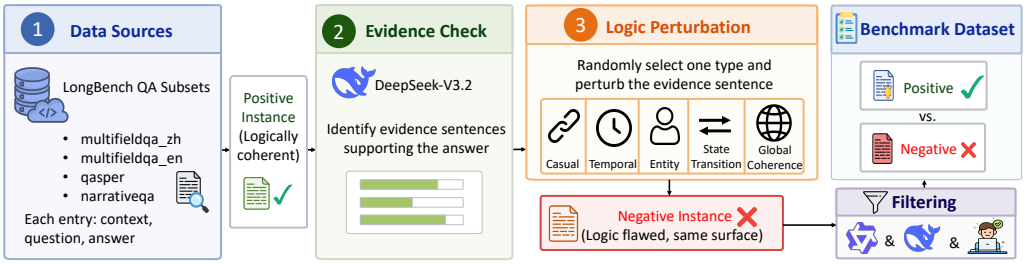

Figure 4: Construction pipeline of the Logic Reward Benchmark dataset.

Models	LFG-Reward	Logic Reward Benchmark		
	Acc.	Acc.	Δ	Prec.
Agentic Systems				
Ours (Doubao-seed-1.8)	63.21	**69.63**	9.25	**46.86**
Ours (DeepSeek-V4-Flash)	**65.34**	65.45	**9.34**	46.20
Ours (Qwen3.5-35B-A3B)	54.49	64.40	8.10	33.68
Proprietary LLMs				
Doubao-Seed-1.8	42.07	33.51	3.48	40.73
Gemini-3-Flash	57.93	52.36	6.37	31.76
GPT-5-mini	50.61	61.26	7.25	17.50
Qwen3.5-Plus	57.01	44.50	7.65	26.39
Open-Source LLMs				
DeepSeek-V4-Flash	55.18	44.50	6.33	40.44
DeepSeek-R1	27.44	41.88	2.43	21.44
DeepSeek-V3.2	31.71	23.56	1.37	26.22
Qwen3-235B-A22B	43.90	45.55	5.21	29.33
Qwen3.5-35B-A3B	43.29	43.98	5.09	24.26

Table 1: Performance comparison on the Long-form RewardBench and Logic Reward benchmarks. **Δ** denotes the score gap between the average chosen and rejected scores. The **best** results are highlighted in each column.

Metrics. Two metrics are reported. Accuracy (Acc.) is the fraction of samples where the chosen response receives a higher score than the rejected response. On the Logic Reward Benchmark, precision (Prec.) is also reported, defined as

$$\text{Prec.} = \frac{\text{correctly located passages}}{\text{total passages flagged by the evaluator}},$$

i.e., the proportion of flagged passages that correspond to genuine perturbations.

Implementation Details. The threshold diff is set to 5 for Long-form RewardBench and 3 for the Logic Reward Benchmark. All LLMs calls use a maximum of 32,768 tokens with temperature 0.0.

4.2 Main Results

Table 1 shows that the proposed agentic framework consistently outperforms single-pass LLMs across both benchmarks.

Superiority of Agentic Systems. Agentic variants occupy the top tier on both benchmarks, surpassing all proprietary and open-source LLMs by a substantial margin. On the Logic Reward Benchmark and LFG-Reward, the three agentic systems achieve the highest Acc., Prec..

Architecture Over Scale. Comparing the same base model across paradigms isolates the architectural contribution. DeepSeek-V4-Flash improves from 44.50% to 65.45% Acc. when embedded in the agentic framework, a gain driven entirely by design rather than model scale, thereby validating our hypothesis in Section 2.

Discrepancy Between Acc. and Prec. Several LLMs exhibit a notable gap between preference accuracy and localization precision. GPT-5-mini achieves a reasonable overall score of 61.26 on the Logic Reward Benchmark yet records the lowest Prec. of 17.50, while agentic systems rank highest on both metrics, suggesting that structured multi-stage evaluation enables genuine logical analysis rather than reliance on superficial cues.

4.3 Ablation Study

An ablation study on the Logic Reward Benchmark assesses two key architectural components, specifically the checklist generation module and the graph builder. As shown in Table 2, removing checklist generation causes a decline in both Acc. and Prec., and removing the graph builder leads to a drop in Acc. with a notable impact on the chosen-rejected gap. This confirms both are essential: the graph builder captures non-linear, cross-sentence logical dependencies, whereas the checklist mitigates cross-dimensional evaluation noise.

Model Variant	Acc.	Prec.
Full System (Doubao-seed-1.8)	**69.63**	**46.86**
w/o Checklist Generation	65.97	43.80
w/o Graph Builder	64.92	46.25

Table 2: Ablation study on the Logic Reward Benchmark. **best** results are highlighted.

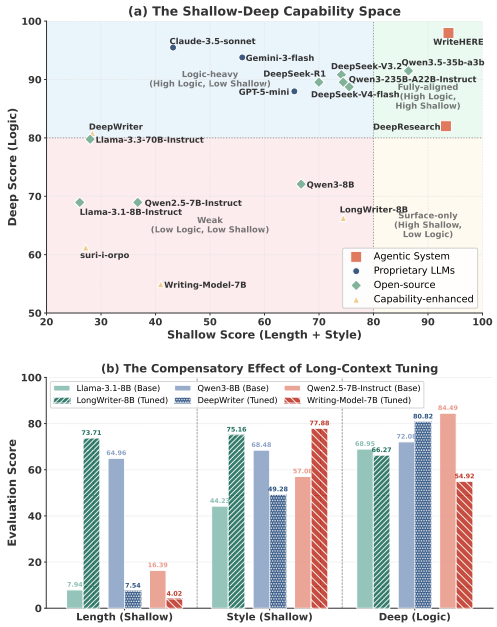

Figure 5: (a) Shallow-Deep capability space; (b) Compensatory effect in long-context tuning.

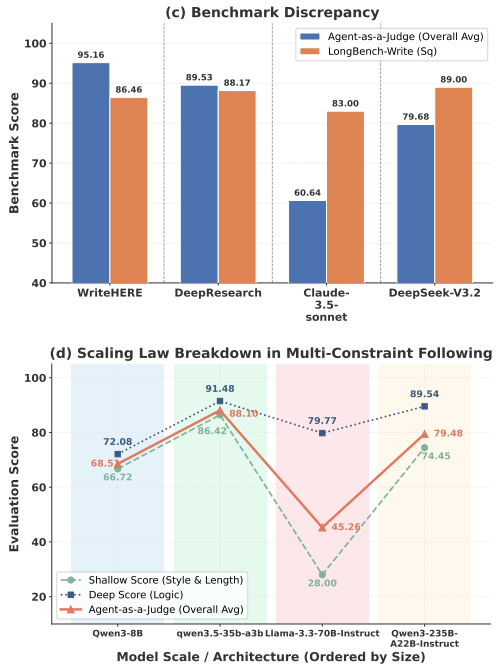

Figure 6: (c) Benchmark discrepancy for agentic systems; (d) Instruction tuning quality vs. parameter scale.

4.4 Long-Form Writing Evaluation

In this section, the DecomWriting-Judge framework is applied to evaluate model outputs on the >4k-word subset of LongBench-Write, producing Shallow, Deep, and overall scores alongside the original S_q baseline. Models span agentic writing systems, proprietary LLMs, open-source LLMs, and capability-enhanced variants. The specific models evaluated and the full results are reported in Appendix F. Figure 5 and 6 summarize the results, from which four findings emerge.

The Shallow-Deep Capability Space As shown in Figure 5(a), the Shallow-Deep decomposition maps models onto a two-dimensional plane. Agentic systems occupy the top-right quadrant with balanced performance across both dimensions, while most proprietary models cluster in the Logic-heavy region, where strong Deep scores compensate for weaker Shallow adherence. The majority of capability-enhanced models remain confined to the Weak quadrant. By disentangling these dimensions, our mapping reveals granular capability profiles and hidden trade-offs that are entirely masked when evaluation is collapsed into a single scalar S_q.

The Compensatory Effect in Long-Context Tuning Long-context tuning consistently introduces trade-offs between shallow compliance and logical coherence, as illustrated in Figure 5(b). For example, Llama-3.1-8B → LongWriter-8B exemplifies this pattern, where Shallow improves from 26.09 to 74.44 while Deep degrades from 68.95 to 66.27. The scalar S_q registers only the net improvement, whereas the DecomWriting-Judge explicitly surfaces the hidden dimensional cost.

Benchmark Discrepancy As shown in Figure 6(c), comparing DecomWriting-Judge scores against S_q reveals ranking discrepancies. WriteHERE achieves the highest DecomWriting-Judge score of 95.16 yet is rated only 86.46 by S_q, below single-pass DeepSeek-V3.2 at 89.00. Conversely, Claude-3.5-Sonnet is substantially overestimated by S_q at 83.00 relative to its DecomWriting-Judge score of 60.64. Such ranking contradictions clearly indicate that traditional baselines favor surface-level writing fluency over the actual satisfaction of complex structural constraints.

Instruction Tuning Quality Over Parameter Scale As depicted in Figure 6(d), multi-dimensional evaluation reveals no positive correlation with parameter scale. Qwen3.5-35B-A3B, with 35B parameters, achieves 88.10 overall, substantially outperforming both Qwen3-235B-A22B-Instruct at 79.48, despite being 7× larger, and Llama-3.3-70B-Instruct at 45.26, despite being 2× larger. The 70B model scores only 28.00 on the Shallow dimension, indicating that parameter scale alone cannot guarantee strong multi-constraint following. Instead, structural compliance depends more on instruction tuning than model size.

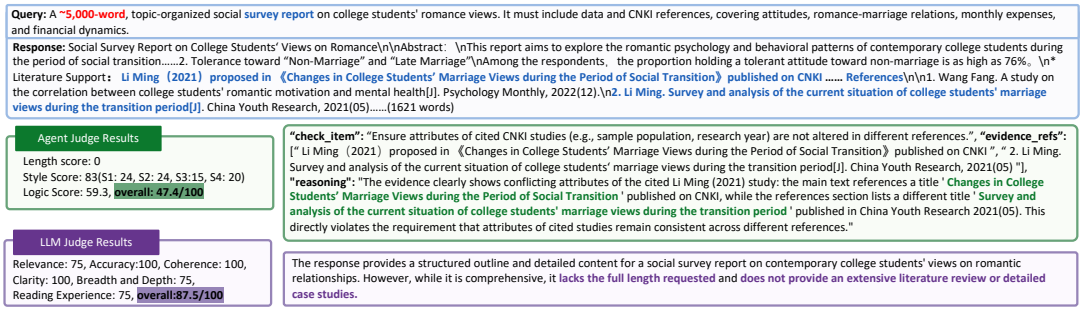

Figure 7: Case study comparing DecomWriting-Judge and LLM-as-a-Judge on a LongBench-Write output.

4.5 Case Study

To concretely illustrate the limitations of single-pass evaluation, we present a case study on a Gemini-3-Flash output evaluated against a query requesting a 5,000-word social survey report. Figure 7 illustrates this discrepancy, where an LLM Judge (GPT-4o-mini) assigns $S_q = 87.5$ to the output while DecomWriting-Judge returns 47.4, driven by a Length score of 0 (1,621 vs. 5,000 required words) and a Logic score of 59.3 that detects a citation inconsistency. This contrast demonstrates how single-pass evaluation reward surface fluency while dimensional decomposition exposes constraint failures invisible to a single score.

4.6 Cost Analysis

A single evaluation under the LLM-as-a-Judge paradigm requires one API call, costing approximately \$0.0009 and completing in under 30 seconds. The DecomWriting-Judge framework performs 10 orchestrated LLM calls across four stages, incurring a per-sample cost of roughly \$0.008 and an average wall-clock time of approximately 10 minutes. While the multi-stage process introduces additional computational overhead relative to single-pass evaluation, the per-sample cost remains minimal in absolute terms and is negligible compared to that of expert human evaluation.

5 Related Work

Long-Form Generation with LLM-as-a-Judge. Long-form evaluation predominantly relies on the LLM-as-a-Judge paradigm, using unified rubrics for holistic scoring (Bai et al., 2025; Liu et al., 2024b) or checklists for fine-grained constraint verification (Wu et al., 2025a,d,c; Xiao et al., 2025; Que et al., 2024; Ruan et al., 2025). Yet single-pass judges suffer from shallow evaluation (You et al., 2026) and the lost-in-the-middle phenomenon (Liu et al., 2024a; An et al., 2024), and tokenizer-based models cannot reliably verify length constraints due to subword tokenization, exhibiting a length bias (Hu et al., 2025) that rewards verbosity over constraint fulfillment.

Agent-as-a-Judge. To address these shortcomings, the Agent-as-a-Judge paradigm (You et al., 2026) reframes evaluation as a dynamic auditing process driven by iterative planning, tool use, and multi-agent collaboration. External tools enable strict constraint verification (Hao et al., 2025; Liu et al., 2025b; Cramer and McIntyre, 2025; Zhuge et al., 2025), while multi-agent debate reduces individual model biases (Saha et al., 2024; Qian et al., 2026; Ma et al., 2025; Choi et al., 2025). Applying agentic evaluators to long-form writing, however, remains underexplored: complex logical dependencies resist surface-level metrics (Wu et al., 2025c; Arias et al., 2025), and integrating verifiable, bias-mitigating mechanisms for detecting deep inconsistencies is still an open challenge.

6 Conclusion

This paper introduced DecomWriting-Judge, a multi-stage agentic framework that evaluates long-form writing through agentic decomposition, checklist-grounded shallow scoring, and adversarial deep logic verification, together with the Logic Reward Benchmark for logical consistency assessment. Experiments demonstrate that agentic evaluation consistently outperforms single-pass LLM judges, with the same base model improving from 44.50% to 65.45% accuracy when embedded in the agentic framework. The Shallow-Deep dimensional decomposition reveals that existing judges obscure shallow-deep trade-offs and structural constraint failures, while each evaluation costs approximately \$0.008, a negligible amount compared to expert human evaluation.

Limitations

Computational Overhead. The multi-stage design introduces additional inference cost compared to single-pass LLM evaluation. Each sample requires approximately 10 orchestrated LLM calls across four stages, incurring roughly $0.008 in API costs and 10 minutes of wall-clock time, whereas a single-pass judge completes in under 30 seconds at negligible cost. While this overhead remains small in absolute terms, it may become a bottleneck for large-scale evaluation pipelines. Future work could explore model distillation to reduce per-stage inference cost, or implement early-exit mechanisms that skip deep logic auditing when shallow dimension scores already indicate severe quality failures.

Checklist Dependency. The quality of evaluation hinges on the query-specific checklists generated in Stage 1. If the checklist generator misinterprets the query or omits relevant criteria, the downstream scores for both style and logic are correspondingly affected. This cascading dependency means that the framework inherits the limitations of its initial generation step. Future work could mitigate this through self-consistency sampling across multiple checklist drafts, or by incorporating a human-in-the-loop verification step for checklist quality before proceeding to scoring.

Scope of Logical Perturbations. The Logic Reward Benchmark covers five perturbation types: causal, temporal, entity, state transition, and global coherence. While these capture common categories of logical inconsistency, they are not exhaustive. Other forms of reasoning error, such as mathematical contradictions, false presuppositions, or domain-specific factual inaccuracies, fall outside the benchmark's current scope. Extending the perturbation taxonomy and expanding beyond QA-sourced passages to more diverse genres would strengthen the benchmark's coverage of real-world logical errors.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2024. Make your llm fully utilize the context. *Advances in Neural Information Processing Systems*, 37:62160–62188.

Anthropic. 2024. Introducing claude 3.5 sonnet. https://www.anthropic.com/research/claude-3-5-sonnet.

Esteban Garces Arias, Hannah Blocher, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2025. Towards better open-ended text generation: A multicriteria evaluation framework. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pages 631–654.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longwriter: Unleashing 10,000+ word generation from long context llms. In *The Thirteenth International Conference on Learning Representations*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. 2025. When identity skews debate: Anonymization for bias-reduced multi-agent reasoning. *arXiv preprint arXiv:2510.07517*.

Marcos Cramer and Lucian McIntyre. 2025. Verifying llm-generated code in the context of software verification with ada/spark. *arXiv preprint arXiv:2502.07728*.

DeepSeek-AI. 2026. DeepSeek-V4 technical report. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf.

Google DeepMind. 2025. Gemini 3 model card and developer documentation. https://ai.google.dev/gemini.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Yilun Hao, Yongchao Chen, Yang Zhang, and Chuchu Fan. 2025. Large language models can solve real-world planning rigorously with formal verification tools. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language*

Technologies (Volume 1: Long Papers), pages 3434–3483.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. 2025. Explaining length bias in LLM-based preference evaluations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6763–6794, Suzhou, China. Association for Computational Linguistics.

Hui Huang, Yancheng He, Wei Liu, Muyun Yang, Jiaheng Liu, Kehai Chen, Bing Xu, Conghui Zhu, Hailong Cao, and Tiejun Zhao. 2026. Long-form rewardbench: Evaluating reward models for long-form generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31149–31157.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Chengwu Liu, Ye Yuan, Yichun Yin, Yan Xu, Xin Xu, Zaoyu Chen, Yasheng Wang, Lifeng Shang, Qun Liu, and Ming Zhang. 2025b. Safe: Enhancing mathematical reasoning in large language models via retrospective step-aware formal verification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12171–12186.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Andrew Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, and 1 others. 2024b. Alignbench: Benchmarking chinese alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11621–11640.

Chiyu Ma, Enpei Zhang, Yilun Zhao, Wenjun Liu, Yaning Jia, Peijun Qing, Lin Shi, Arman Cohan, Yujun Yan, and Soroush Vosoughi. 2025. Judging with many minds: Do more perspectives mean less prejudice? on bias amplifications and resistance in multi-agent based llm-as-judge. *arXiv preprint arXiv:2505.19477*.

Chau Minh Pham, Simeng Sun, and Mohit Iyyer. 2024. Suri: Multi-constraint instruction following in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1722–1753.

Yiyue Qian, Shinan Zhang, Yun Zhou, Haibo Ding, Diego Socolinsky, and Yi Zhang. 2026. Collabeval: Enhancing llm-as-a-judge via multi-agent collaboration. *arXiv preprint arXiv:2603.00993*.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, and 1 others. 2024. Hellobench: Evaluating long text generation capabilities of large language models. *arXiv preprint arXiv:2409.16191*.

Qwen Team. 2026. Qwen 3.5 technical report. https://qwen.ai/blog?id=qwen3.5.

Jie Ruan, Inderjeet Nair, Shuyang Cao, Amy Liu, Sheza Munir, Micah Pollens-Dempsey, Tiffany Chiang, Lucy Kates, Nicholas David, Sihan Chen, and 1 others. 2025. Expertlongbench: Benchmarking language models on expert-level long-form generation tasks with structured checklists. *arXiv preprint arXiv:2506.01241*.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2024. Branch-solve-merge improves large language model evaluation and generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8352–8370.

Bytedance Seed. 2026. Seed1. 8 model card: Towards generalized real-world agency. *arXiv preprint arXiv:2603.20633*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Kaiyang Wan, Honglin Mu, Rui Hao, Haoran Luo, Tianle Gu, and Xiuying Chen. 2025. A cognitive writing perspective for constrained long-form text generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9832–9844.

Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, and 1 others. 2025. Reverse-engineered reasoning for open-ended generation. *arXiv preprint arXiv:2509.06160*.

Siwei Wu, Yizhi Li, Xingwei Qu, Rishi Ravikumar, Yucheng Li, Tyler Loakman, Shanghaoran Quan, Xiaoyong Wei, Riza Batista-Navarro, and Chenghua Lin. 2025a. Longeval: A comprehensive analysis of long-text generation through a plan-based paradigm. *arXiv preprint arXiv:2502.19103*.

Yuhao Wu, Yushi Bai, Zhiqiang Hu, Roy Ka-Wei Lee, and Juanzi Li. 2025b. Longwriter-zero: Mastering ultra-long text generation via reinforcement learning. *arXiv preprint arXiv:2506.18841*.

Yuhao Wu, Ming Shan Hee, Zhiqiang Hu, and Roy Ka-Wei Lee. 2025c. Longgenbench: Benchmarking long-form generation in long context LLMs. In *The Thirteenth International Conference on Learning Representations*.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Wang Zijia, Ji Zhang, Mengyue Wu, Qin Jin, and 1 others. 2025d. Writingbench: A comprehensive benchmark for generative writing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zikai Xiao, Fei Huang, Jianhong Tu, Jianhui Wei, Wen Ma, Yuxuan Zhou, Jian Wu, Bowen Yu, Zuozhu Liu, and Junyang Lin. 2025. LongWeave: A long-form generation benchmark bridging real-world relevance and verifiability. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10386–10417, Suzhou, China. Association for Computational Linguistics.

Ruibin Xiong, Yimeng Chen, Dmitrii Khizbullin, Mingchen Zhuge, and Jürgen Schmidhuber. 2025. Beyond outlining: Heterogeneous recursive planning for adaptive long-form writing with language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24689–24725. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Runyang You, Hongru Cai, Caiqi Zhang, Qiancheng Xu, Meng Liu, Tiezheng Yu, Yongqi Li, and Wenjie Li. 2026. Agent-as-a-judge. *arXiv preprint arXiv:2601.05111*.

Mingchen Zhuge, Changsheng Zhao, Dylan R Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, and 1 others. 2025. Agent-as-a-judge: Evaluate agents with agents. In *International Conference on Machine Learning*, pages 80569–80611. PMLR.

A DeepResearch Pipeline

Algorithm 1 outlines the DeepResearch agent used in the preliminary study. The agent operates in two phases: iterative research via search-query generation and reflection, followed by length-controlled writing with a feedback loop that revises the draft until a length score threshold is met.

Algorithm 1 DeepResearch Pipeline

Require: User prompt $\mathcal{P}$, length target l
Ensure: Final written draft $\mathcal{D}$

Phase 1: Iterative Research
$c_{\text{loop}} \leftarrow 0$
repeat
  Generate k search queries from $\mathcal{P}$
  for each query **in parallel do**
    Perform web search and synthesize findings
  end for
  Reflect on gathered information and identify knowledge gaps
  $c_{\text{loop}} \leftarrow c_{\text{loop}} + 1$
until no gaps remain **or** $c_{\text{loop}} \geq$ *max_reflect_loops*

Phase 2: Length-Controlled Writing
$c_{\text{rewrite}} \leftarrow 0$
Generate initial draft $\mathcal{D}$ conditioned on length target l
Calculate length score $\mathcal{S}$ by comparing draft length l' against l
while $\mathcal{S} < \tau$ **and** $c_{\text{rewrite}} <$ *max_rewrite_loops* **do**
  Generate section-level revision guidance
  Revise draft $\mathcal{D}$ based on guidance
  Recalculate draft length l' and score $\mathcal{S}$
  $c_{\text{rewrite}} \leftarrow c_{\text{rewrite}} + 1$
end while
return $\mathcal{D}$

B Evaluation Rules

Table 3 lists the abstract evaluation rules that serve as the foundation for query-specific checklist generation in Stage 1. The rules are organized into two dimensions: style (four rules) and logic (five rules).

C Checklist Generation Example

Table 4 illustrates a complete Style checklist generated by the Style Agent for a query requesting an academic research paper on autonomous vehicle navigation. Each rule is instantiated with query-specific check items and a 6-band scoring rubric.

Table 5 shows the corresponding Logic checklist for the same query. Unlike the Style checklist, Logic rules are verified adversarially in Stage 3 rather than scored via rubric, so only the rule focus and check items are generated.

Rule	Description
	Style
Writing style	Register unity, free of immersion-breaking expressions.
Rhetoric & technique	Quality of rhetorical devices, sentence rhythm, and imagery originality.
Query alignment	Requested style reflected in generated text.
Vocabulary	Avoidance of word repetition and semantic mismatches.
	Logic
Causal	Every outcome must have a plausible cause.
Temporal	Event ordering must be temporally coherent.
Entity	Entity attributes must not change without explanation.
State transitions	State transitions must be continuous and logically justified.
Global coherence	Full text stays on-topic with convergent structure.

Table 3: Abstract evaluation rules for the style and logic dimensions.

D Dataset Statistics

The Logic Reward Benchmark contains 191 samples drawn from four QA subsets of LongBench (Bai et al., 2024). Table 6 reports the dataset–language cross distribution. The dataset is predominantly in English (64.4%), with the remaining samples in Chinese (35.6%). Context lengths range from 688 to 23,433 words, with an average of 4,763 words and a median of 4,213 words. The majority of passages fall within 2,000 to 10,000 words, as shown in Figure 8. Figure 9 presents the distribution of the five perturbation types and their relative depth position within the context. State and global coherence perturbations are the most frequent (26.2% and 25.1%, respectively), followed by causal and entity perturbations (18.8% each), and temporal perturbations (11.0%). Each sample contains exactly one perturbed passage. The perturbation depth ratio ranges from 0 to 0.99 with an average of 0.32.

E Precision Metric for Logic Reward Benchmark

The localization precision (Prec.) reported in Table 1 measures the evaluator's ability to correctly locate the perturbed passage within each sample of the Logic Reward Benchmark. Since each sample

Rule	Check Items	Scoring Bands (0–25)
S1: Writing style	(1) No casual language or slang in any section. (2) Objective, analytical tone across all sections. (3) No abrupt shifts from formal to informal language.	0–2: Frequent casual language; 3–7: Multiple tone shifts; 8–12: Inconsistent professionalism; 13–17: Rare isolated issues; 18–22: Uniform tone, negligible shifts; 23–25: Perfectly consistent academic register.
S2: Rhetoric & technique	(1) Parallel structure in optimization steps. (2) Varied sentence structure to avoid monotony. (3) Analogies clarify complex concepts without distraction.	0–2: Monotonous, confusing; 3–7: Repetitive, forced devices; 8–12: Some variation, inconsistent clarity; 13–17: Moderate variation, minor gaps; 18–22: Effective variation and parallelism; 23–25: Expertly varied, precise parallelism.
S3: Query alignment	(1) Standard research paper structure. (2) Formal academic tone with precise technical language. (3) Sufficient detail on optimization and efficiency reformulations.	0–2: No research structure; 3–7: Missing key sections, inconsistent tone; 8–12: Basic structure, superficial detail; 13–17: All components covered with sufficient detail; 18–22: Rigorous tone, detailed explanations; 23–25: Comprehensive, perfectly aligned.
S4: Vocabulary	(1) No over-repetition of key terms. (2) Technical jargon used accurately. (3) Synonyms enhance readability without sacrificing precision.	0–2: Frequent repetition, incorrect jargon; 3–7: Significant over-repetition, incorrect jargon; 8–12: Some repetition, occasional errors; 13–17: Minimal repetition, mostly accurate; 18–22: Precise vocabulary, negligible issues; 23–25: Expertly varied, flawless jargon use.

Table 4: Example Style checklist generated for an academic research query. Check items and bands are abbreviated; the full rubric contains detailed criteria for each band.

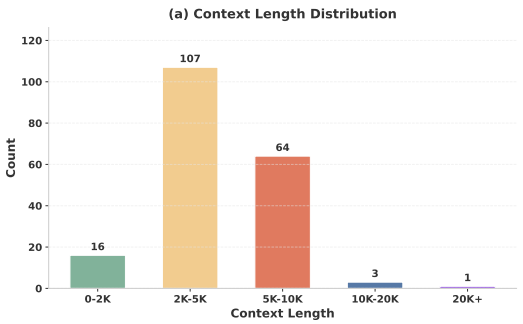

Figure 8: Context length distribution of the Logic Reward Benchmark.

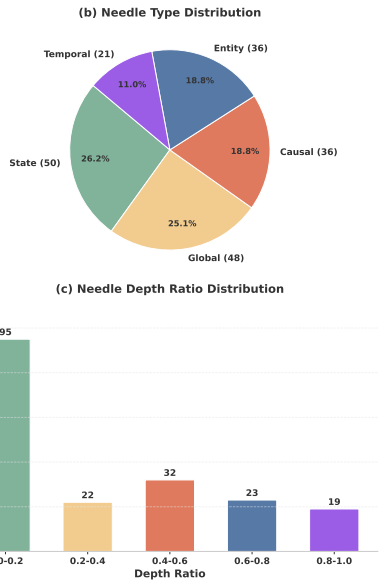

Figure 9: Perturbation type and depth ratio distribution.

contains exactly one known perturbed passage (the "needle"), precision quantifies whether the evaluator pinpoints the injected logical error rather than flagging unperturbed regions.

During the adversarial logic audit (Stage 3), the Challenger agent examines the text and proposes candidate contradictions, which are then validated by the Retriever and verified by the Verifier, yielding the verified contradiction set $\mathcal{D}^*$. Each contradiction in $\mathcal{D}^*$ is associated with one or more sen-

Rule	Focus	Check Items
C1: Causal	Every stated benefit has a plausible causal link to the system design or optimization.	(1) Each claimed benefit tied to a specific system component. (2) Efficiency improvements linked to explicit reformulations. (3) Comfort outcomes connected to model constraints or objectives.
T1: Temporal	Temporal coherence in the receding horizon cycle and hierarchical operation sequence.	(1) Time window updates follow a logically consistent flow. (2) Hierarchical decision layers operate in coherent temporal order. (3) Dynamic state updates align with real-world timeline constraints.
E1: Entity	Consistent attributes for key entities across all sections.	(1) Key entity definitions unchanged throughout. (2) Hierarchical system structure described uniformly. (3) Performance metrics use consistent measurement methodologies.
S1: State	Continuous, justified state transitions in vehicle navigation and optimization.	(1) Vehicle state transitions justified by system outputs and conditions. (2) Horizon optimization state updates align with dynamic constraints. (3) Transitions between decision layers supported by valid dependencies.
G1: Global	Convergent structure supporting the core thesis of system effectiveness.	(1) All sections relate to the hierarchical system design. (2) Logical flow from problem to design to optimization to validation. (3) All claims converge to support the system's core capability.

Table 5: Example Logic checklist generated for the same query as Table 4.

Dataset	Language	Count
multifieldqa_zh	zh	68
qasper_e	en	64
multifieldqa_en_e	en	56
narrativeqa	en	3
Total		**191**

Table 6: Dataset–language cross distribution of the Logic Reward Benchmark.

tences as evidence. For each evidence sentence, we determine whether it correctly identifies the needle as follows. Let e be the embedding of the evidence sentence and g be the embedding of the ground-truth perturbed sentence, both encoded by BGE-M3 (Chen et al., 2024). The evidence sentence is considered a correct localization if the cosine similarity between the two embeddings exceeds a threshold of 0.85:

$$\text{correct}(f) = \begin{cases} 1 & \text{if } \cos(e_f, g) > 0.85 \\ 0 & \text{otherwise} \end{cases},$$

where $\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$.

For a given sample, let $\mathcal{F}$ be the set of sentences flagged by the evaluator. The sample-level precision is then:

$$\text{Prec.}_{\text{sample}} = \frac{\sum_{f \in \mathcal{F}} \text{correct}(f)}{|\mathcal{F}|},$$

with $\text{Prec.}_{\text{sample}} = 0$ if $\mathcal{F}$ is empty. The aggregate Precision reported in Table 1 is the macro-average over all 191 samples:

$$\text{Prec.} = \frac{1}{N} \sum_{i=1}^{N} \text{Prec.}_{\text{sample}}^{(i)},$$

Models	DecomWriting-Judge			Avg.	By Depth		S_q
	Length	Style	Logic		Shallow	Deep	
Agentic Systems							
DeepReSearch	**95.48**	91.12	82.00	89.53	93.30	82.00	88.17
WriteHERE	94.07	**93.45**	**97.97**	**95.16**	**93.76**	**97.97**	86.46
Proprietary LLMs							
Claude-3.5-Sonnet	8.82	77.62	95.47	60.64	43.22	95.47	83.00
Gemini-3-Flash	22.26	89.58	93.78	68.54	55.92	93.78	87.33
GPT-5-mini	42.30	88.66	87.97	72.98	65.48	87.97	86.17
Open-Source LLMs							
DeepSeek-V4-Flash	60.76	90.31	88.74	79.94	75.54	88.74	86.67
DeepSeek-R1	47.71	92.24	89.54	76.50	69.98	89.54	85.83
DeepSeek-V3.2	55.44	92.78	90.82	79.68	74.11	90.82	**89.00**
Qwen3-235B-A22B-Instruct	57.62	91.28	89.54	79.48	74.45	89.54	87.17
Llama-3.3-70B-Instruct	1.12	54.88	79.77	45.26	28.00	79.77	69.83
Qwen3.5-35B-A3B	80.47	92.36	91.48	88.10	86.42	91.48	88.33
Qwen3-8B	64.96	68.48	72.08	68.51	66.72	72.08	75.50
Llama-3.1-8B-Instruct	7.94	44.23	68.95	40.37	26.09	68.95	57.00
Qwen2.5-7B-Instruct	16.39	57.08	84.49	52.98	36.74	84.49	71.83
Capability-Enhanced LLMs							
LongWriter-8B	73.71	75.16	66.27	71.71	74.44	66.27	76.33
Writing-Model-7B	4.02	77.88	54.92	45.61	40.95	54.92	75.67
DeepWriter	7.54	49.28	80.82	45.88	28.41	80.82	66.83
Suri-I-ORPO	24.75	29.62	61.21	38.53	27.19	61.21	33.17

Table 7: Performance of various models evaluated by the proposed DecomWriting-Judge framework on the LongBench-Write benchmark (>4k subset). **Shallow** aggregates Length and Style scores; **Deep** corresponds to the Logic score; S_q is the original LongBench-Write quality score judged by GPT-4o-mini. The **best** results in each column are highlighted.

F Evaluated Models and Full Results

F.1 Evaluated Models

The following models are evaluated in the long-form writing evaluation (Section 4.4). Models span four categories, reflecting diverse generation paradigms and scales.

- **Agentic Systems.** These frameworks decompose the writing task into structured sub-tasks through planning, retrieval, and iterative refinement, as introduced in Section 2. This category includes DeepResearch and WriteHERE, the two agentic systems evaluated in our preliminary study.
- **Proprietary LLMs.** These are commercial, API-accessible models that generate long-form text via single-pass inference. This category includes Claude-3.5-Sonnet (Anthropic, 2024), Gemini-3-Flash, and GPT-5-mini, representing the frontier of proprietary long-form generation capabilities.
- **Open-Source LLMs.** These publicly available models cover a wide range of parameter scales from 7B to 235B. This category includes DeepSeek-V4-Flash, DeepSeek-R1, DeepSeek-V3.2, Qwen3-235B-A22B-Instruct, Llama-3.3-70B-Instruct (Grattafiori et al., 2024), Qwen3.5-35B-A3B, Qwen3-8B, Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Qwen2.5-7B-Instruct (Yang et al., 2024).
- **Capability-Enhanced LLMs.** These models are explicitly fine-tuned or optimized for long-form writing tasks. This category includes LongWriter-8B (Bai et al., 2025), trained for 10,000+ word generation, Writing-Model-7B (Wu et al., 2025d), a writing-specific fine-tuned variant based on Qwen2.5-7B-Instruct, DeepWriter (Wang et al., 2025), a writing-specialized model(based on Qwen3-8B) using reverse-engineered reasoning, and Suri-I-ORPO (Pham et al., 2024), trained with instructional odds ratio preference optimization.

F.2 Evaluation Results

Table 7 reports the full evaluation results, consistent with the four findings discussed in Section 4.4. Agentic systems occupy the top tier with balanced Shallow and Deep scores, with WriteHERE achieving the highest overall average of 95.16. Proprietary LLMs cluster in the Logic-heavy region, where strong Deep scores (e.g., Claude-3.5-Sonnet at 95.47) are offset by critically low Length scores (8.82), reflecting a systematic failure in length constraint adherence that S_q largely masks. Claude-3.5-Sonnet scores 83.00 under S_q yet only 60.64

average score under DecomWriting-Judge. Among open-source LLMs, Qwen3.5-35B-A3B achieves 88.10 overall, substantially outperforming both Qwen3-235B-A22B-Instruct (79.48, 7× larger) and Llama-3.3-70B-Instruct (45.26, 2× larger), confirming that instruction tuning quality governs multi-constraint compliance more than parameter scale. Capability-enhanced models remain confined to the Weak quadrant, revealing a persistent trade-off between the two dimensions: Optimizations enforce a strict trade-off: improving Shallow compliance at the expense of Deep logic (e.g., LongWriter-8B vs. Llama-3.1-8B-Instruct), or conversely, sacrificing Shallow performance to preserve Deep reasoning (e.g., DeepWriter vs. Qwen-3B).