MERIDIAN: A LONG CONTEXT, ROBUST INSTRUCTION FOLLOWING BENCHMARK

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce MERIDIAN, a benchmark for long-context (LC) instruction following (IF) over real-world documents. IF is the model capability that allows users to condition model output. Our goals are to enable diagnostics of model failures for this setting by exploring robustness to instruction positions in the prompt and resistance to instruction injection. We design prompts that state document-understanding tasks together with a set of typed constraints incremental to the task. We target one general knowledge domain and four domains relevant to knowledge work (finance, legal, medical, cloud service documentation) both through documents and constraints specific to the domains. We validate instruction prompts via human review. We report *Task score*, and *IF score* for the instruction prompts, and a weighted combined score. We test IF across changes in context length, instruction count, and instruction placement; we further explore single- and multi-turn settings. We evaluate current models and show that 1) Combined scores on average degrade 15.8% (from 45.4% to 38.3%) when context length increases from 8k to 128k tokens, and are sensitive to prompt position; 2) *IF score* has headroom (max of 8.5% for the most complex prompts), and is only moderately correlated with writing quality. We plan to release the benchmark and code upon acceptance.

1 INTRODUCTION

Instruction following is a Large Language Model's (LLM) ability to comply with instructions stated in the system prompt or by the user. Models acquire it through instruction tuning and alignment from human feedback (Ouyang et al., 2022; Chung et al., 2024). Existing IF and long-context benchmarks usually test instruction complexity (He et al., 2026; Pyatkin et al., 2025; Li et al., 2026), tasks over long contexts Wang et al. (2024), knowledge work (Akyürek et al., 2026; Patwardhan et al., 2026), or a model's robustness to input variation (Wu et al., 2025). To our knowledge, no evaluations test all these factors together. As enterprises increasingly leverage LLMs for analyzing documents, users rely on models to process increasing amounts of documents while following requirements stated as instructions. Models used in AI assistants need to be robust to user inputs in such settings because user–model interactions are organic and do not follow specific structures (Zhou et al., 2026).

To address this gap, we introduce MERIDIAN (**M**ulti-domain **E**valuation of **R**obust **I**nstructions across **D**ocument-length **I**nputs with **AN**notation), a framework and a benchmark for robust long-context instruction following on real-world domain documents. We contribute an evaluation dataset which we built using real-world documents and human-validated instructions that are relevant to them. MERIDIAN consists of 3131 total prompts covering token ranges from 1-8k, 8-16k, up to 128k, with the following prompt counts by domain: 699 for finance, 598 for legal, 650 for medicine, 713 for cloud architecture documentation, 471 for general (See Appendix A).

Previous work has identified that LLM users specify certain *task types* (*e.g.*, summarization, Q&A, list extraction), often accompanied by additional constraints (Yuan et al., 2022; Trippas et al., 2024). We therefore *define* a prompt as a combination of an overarching task with additional instructions that constrain it (see Figure 1 for a prompt example). To measure each component, we *define*: *Task score* which captures how well the LLM performs the overarching task, and *IF score* which evaluates compliance with all the constraining instructions. We create instructions based on a constraint taxonomy that extracts entities and topics from the documents to seed instruction templates (§3). We use human review to validate both the instruction prompts and the rubrics used to judge model gener-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

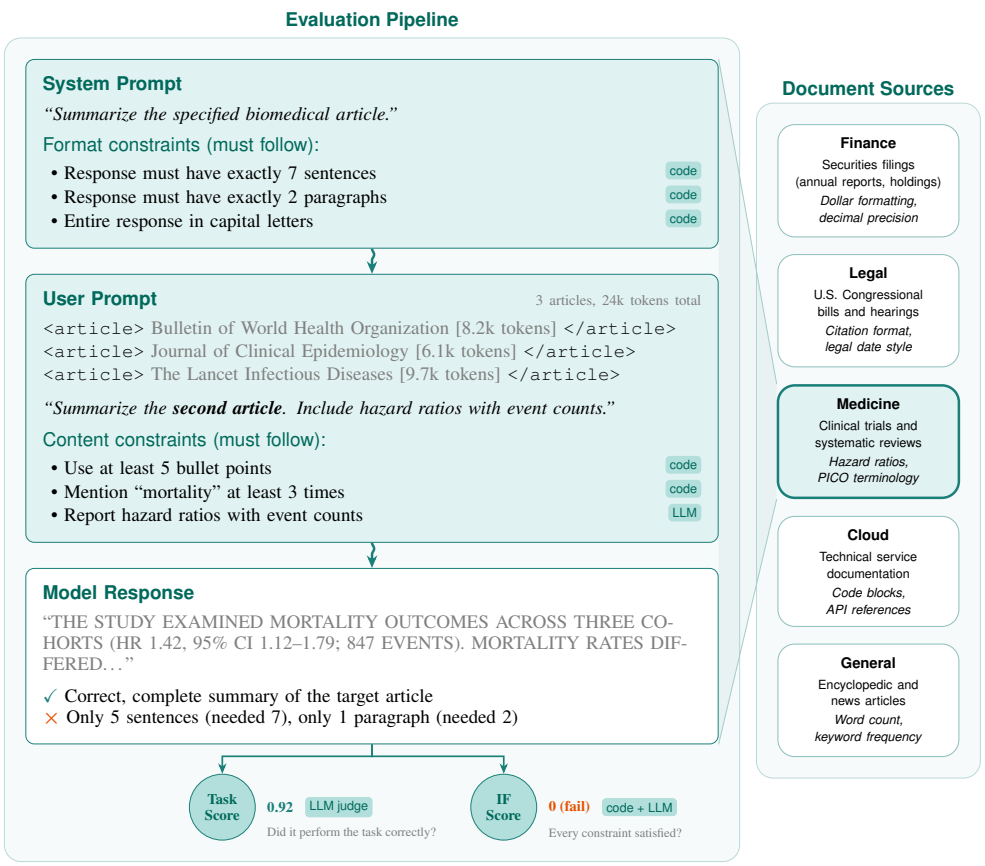

Figure 1: MERIDIAN benchmark pipeline (medicine example). The system prompt carries format constraints and the user turn carries content constraints, each with a fixed verifier (code or LLM). This example shows a summary that fails two format constraints, yielding high *Task score* but zero *IF score*.

ations, and iteratively refine the rubrics to convergence across judges (see Figure 2, §B). Finally, we benchmark current LLMs, reporting *Task* and *IF* scores and measuring robustness to positional bias and to instructions injected into documents (§3.4). Within MERIDIAN we investigate the following:

- **RQ1** *Are task performance and IF separable?* We hypothesize that they are distinct axes: a model can succeed at the task while failing the instructions, and vice versa. (§4.1)
- **RQ2** *Does instruction placement and conversational structure affect IF?* We hypothesize that scattering instructions through the context or across turns challenged attention more than clustered instructions. (§4.2)
- **RQ3** *Do models respect instruction hierarchy in long documents?* We hypothesize that as context length and instruction counts increase, models follow a substantial fraction of instructions injected in document content, when they are not asked to, in addition to the system or user instructions. (§4.3)

The rest of the paper is organized as follows. Section 2 positions MERIDIAN relative to prior work. Section 3 describes the benchmark construction, instruction taxonomy, and scoring methodology. Section 4 presents the experimental results organized around RQ1–RQ3. We hope that MERIDIAN contributes to the science of benchmarking by raising the complexity of what is measured and how failure is attributed. We plan to release the benchmark and evaluation code.

2 RELATED WORK

IF evaluation has outgrown verifiable constraints, yet rubric-based judging remains unreliable. IFEval (Zhou et al., 2023) defines verifiable output constraints (length, keywords, format)

that later suites extend (Wang et al., 2025; Xia et al., 2024; Tang et al., 2024), but these checks saturate quickly (Lambert et al., 2024; Kamath et al., 2025) and IFBench (Pyatkin et al., 2025) shows high scores reflect overfitting rather than skill. Composing constraints raises difficulty (He et al., 2024; Jiang et al., 2024; Wen et al., 2024; Zou et al., 2025; Mehta et al., 2026; Hayati et al., 2025), and because realistic constraints resist programmatic checking, evaluation increasingly relies on LLM judges that decompose instructions into atomic questions (Qin et al., 2024b; Ke et al., 2023; Zheng et al., 2023). Rubrics now serve as both evaluation targets and training rewards (He et al., 2026; Liu et al., 2026; Gunjal et al., 2025; Viswanathan et al., 2025), yet they inherit the judge's weaknesses (Wen et al., 2026). We use both code validators for constraints that admit exact checking, and LLM rubrics, with the verifier fixed per constraint at sampling time.

LC capability and professional-domain realism advance on separate tracks When testing LC tasks, benchmarks have moved from synthetic retrieval probes (Schuster et al., 2025; Hsieh et al., 2024) to multitask suites standardizing evaluation up to 100k+ tokens (Bai et al., 2024; Zhang et al., 2024; An et al., 2024; Yen et al., 2025), with a further line forcing reasoning over dispersed evidence (Wang et al., 2024; Ye et al., 2025; Wu et al., 2024). On the domain side, benchmarks encode professional expertise but usually fix a single field: legal (Guha et al., 2023; Fei et al., 2024; Shi et al., 2026), financial (Xie et al., 2026; 2024), clinical (Arora et al., 2025), and cross-domain rubrics (Akyürek et al., 2026; Song et al., 2025). GDPVal defines tasks across domains using complex human-authored prompts that do not explore LC or robustness (Patwardhan et al., 2026). The two tracks rarely meet: LIFBench (Wu et al., 2025) pairs LC with IF and reporting paraphrase stability, while MDCure (Liu et al., 2025) shows multi-document training helps. We join both tracks directly, placing instructions over real-world documents from 8k to 128k tokens across five domains with a shared constraint taxonomy and domain-specific extensions. The context length of our prompts can be arbitrarily padded with additional distractor documents for longer contexts.

Robustness and instruction hierarchy are tested in isolation from context length The instruction hierarchy specifies when system prompts outrank user prompts (Wallace et al., 2024); SysBench shows models violate system constraints across turns (Qin et al., 2024a); IHEval finds the strongest open model resolves only 48% of priority conflicts at fixed length (Zhang et al., 2025). Models also struggle to separate system or user instructions from input that resembles them (Yi et al., 2023), and accuracy depends on where content sits in the context (Liu et al., 2023). Prior work measures these concerns separately from context length: robustness is tested on short context while LC benchmarks assume cooperative prompts. We bring them together by varying instruction placement and inserting adversarial in-document instructions as context grows, directly testing whether hierarchy adherence and positional stability hold at scale.

3 Benchmark Construction and Evaluation

We first describe our approach for building LC prompts with instructions that are targeting specific document(s). We then describe our human and automated processes for validating the prompts and evaluation methods. Figure 2 illustrates this pipeline and we explain each stage.

3.1 Problem formulation

Let $\mathcal{D}$ be a corpus of domain documents, each with token length $\ell(d)$. A benchmark instance is

$$x = \rho\big(s,\, u,\, C,\, \pi\big), \tag{1}$$

where s is a system prompt, u a user prompt, $C = (d_1, \ldots, d_m) \subset \mathcal{D}$ an ordered set of context documents targeting a context window of W tokens, π a layout fixing the relative placement of s, u, and the documents, and $\rho(\cdot)$ a deterministic template renderer. Constraints are drawn as a multi-set $I = I_s \cup I_u$ split between the system and user prompts by a respective number of target instructions in each prompt. Each instruction $\iota \in I$ carries sampled parameters θ_ι, a natural-language description, and an evaluation method $m(\iota) \in \{\textsc{code}, \textsc{llm}, \textsc{code+llm}\}$. Each $\iota \in I$ is mapped to a verifier. We define an incompatibility map to mitigate sampling instructions that are mutually exclusive. A model M produces response $r = M(x)$, and each instruction induces a verifier

$$v_\iota(r) \in \{1,\, 0,\, \bot\}, \tag{2}$$

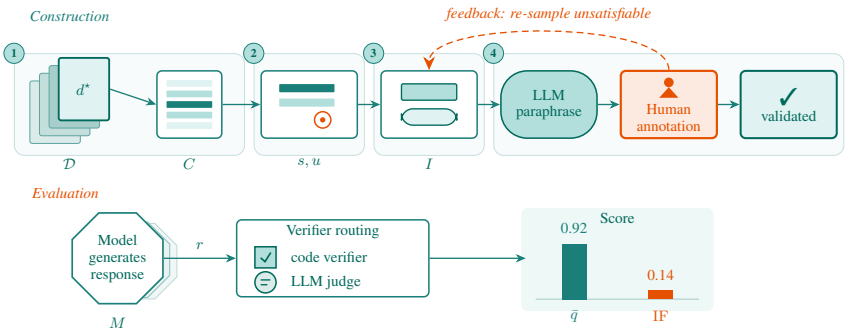

Figure 2: MERIDIAN pipeline overview. **Top:** Development stages (1. document selection from corpus to target context length → 2. the system and user prompts, s and u, define the task and constraints types → 3. constraint instantiation via parameters → 4. model paraphrase of user prompt is human-validated.). **Bottom:** Evaluation (model generation → per-constraint verification defined by constrained type). Score bars show *Task score* is high while *IF score* remains low.

where 1 and 0 denote followed and violated, and $\perp$ marks a non-applicable instruction. The benchmark reports how scores behave as we vary W, $|I|$, π.

3.2 FROM DOCUMENTS TO EVALUATION INSTANCES

Our prompt construction pipeline has four stages:

Stage 1: Document selection and context assembly. We draw documents/articles from the corpus $\mathcal{D}$ across five domains: financial filings (U.S. Securities and Exchange Commission[1]), legislative records (U.S. Congressional bills and hearings[2]), biomedical literature (PubMed Central[3]), technical documentation (AWS service docs[4]), and general text (English Wikipedia[5]) . We filter out programmatic or short documents using keywords and file size. For each instance, we sample a target document $d^\star$ and fill the ordered context $C = (d_1, \ldots, d_m)$ with additional distractor documents until the target window W is reached. The model needs to ignore distractor documents when addressing the task. We uniformly sample distractor documents from a large collection of documents of the same domain and type(s) as the target. Longer windows yield proportionally more distractors, so the model must locate the right document among many.

Stage 2: Task assignment. Each instance is assigned a task type $\tau \in \{$summarization, classification, closed Q&A, list extraction$\}$, each performed over a target article where classification assigns it to a predefined label (*e.g.*, "is the third article's stance for or against?"), closed Q&A answers a question from its content alone (*e.g.*, "what year does the third article cite?"), and list extraction returns the items in it that meet a criterion (*e.g.*, "list the people named in the third article"). The system prompt s states the task and carries system-level constraints I_s; the user prompt u identifies the target by position (*e.g.*, "summarize the third article") and carries user-level constraints I_u.

Stage 3: Constraint sampling. Each instance receives instructions sampled from a taxonomy with the following top tiers: *generic constraints* cover formatting and structure (*i.e.* response length, paragraph count, case, required or disallowed keywords, and positional requirements); *domain-specific static constraints* that are relevant to the target domain (*e.g.* "all dollar amounts must use exactly 2 decimal places", or "on first use of IS in each section, expand it as 'Introduced in Senate'"; *domain-specific templated constraints*, for which we use a model to fill in templates per domain, (*e.g.* "Trace how amendments in [bill$_a$] affect provisions referenced in [bill$_b$]." is filled with two specific bills discussed in the target Congressional transcript document).

Stage 4: Paraphrase and validation. The user prompt component u is paraphrased by an LLM to remove template artifacts. We have paraphrases validated by one expert human annotator who veri-

[1] https://www.sec.gov/search-filings; [2] https://www.congress.gov/congressional-record; [3] https://pmc.ncbi.nlm.nih.gov/; [4] https://docs.aws.amazon.com/; [5] https://en.wikipedia.org/.

fies feasibility and paraphrase faithfulness (see Appendix B). We human-review LLM judge rubrics and test their clarity and reliability using inter- and intra-judge agreement (4 modes, 3 answers each). We select for rewrite and review rubrics scoring below Krippendorff's $\alpha = 0.667$; improving α from 0.15 to 0.68 in the worst case.

3.3 Robustness conditions

Starting from the prompt elements described above (documents, system prompts, and user prompts), we derive controlled variants that change one aspect of how the same content is delivered.

Single-turn placement: All placement conditions keep the system prompt in the system role and preserve the native paraphrased user request. The FIRST configuration places the request before all articles. LAST places it after all articles and is the untransformed native layout. MIDDLE splits the articles at the halfway point and inserts the request between the two halves, inside the `<articles>` block. DISPERSED splits the request on sentence boundaries into three groups, placing the groups at the start, in the middle of the articles, and at the end (if there are less than three instruction groups, we first place at the start and then at the end).

Multi-turn delivery: The paired comparison changes the turn boundaries while preserving the text, with model replies between user messages. In this condition we leave the API system prompt empty and pass it in the user-role in the first turn. We then pass the rest of the prompt in subsequent turns in the following conditions. Each plan contains Task (T), Documents (D), three instruction utterances (I_1, I_2, I_3), and a fixed final request (G; "give me the complete final answer now"). We compare three content orders: Task-first, $T \to D \to I \to G$; Documents-first, $D \to T \to I \to G$; and Documents-last$T \to I \to D \to G$.

In-document injection: The IH-MIDDLE condition keeps the system prompt in the system role and moves one user instruction inside a non-target article while leaving the remaining instructions as before. The moved instruction should be marked as document content and ignored by the model.

3.4 Scoring

Task score measures if the model achieved the text writing task independent of the additional constraints. A model judge scores the response against a task-specific rubric with K criteria, such as preserving salient entities and relationships for the summarization task. The criteria are rated on a 1–5 scale, yielding the score q, where $\bar{q}(r) = \frac{1}{K}\sum_{k=1}^{K}(q_k - 1)/4 \in [0, 1]$. We use a binary *IF score* to measure whether all the instructions in the prompt are followed (all-or-nothing): $\mathrm{IF}(r, I) = \prod_{\iota} \mathbf{1}[v_\iota(r) = 1]$. We prefer all-or-nothing to micro-averaging, as users typically expect every instruction followed rather than only a subset. We also define a *Combined score* = $0.6*$ *Task Score* $+0.4*$ *IF score* (full formulas in Appendix C). For the instruction injection condition we add the proportion of in-document instructions that the model (incorrectly) followed: $\rho_{\mathrm{inj}}(r) = \frac{1}{|I'|}\sum_{\iota' \in I'} \mathbb{1}[v_{\iota'}(r) = 1]$. The condition-specific score accounts for both injected and legitimate *IF scores*: $S_{\mathrm{inj}} = 0.6, \mathrm{IF}_{\mathrm{legit}} + 0.4\,(1 - \rho_{\mathrm{inj}})$. We denote $\mathrm{IF}_{\mathrm{legit}}$ as the fraction of non-injected instructions followed: $\mathrm{IF}_{\mathrm{legit}}(r, I) = \frac{1}{|I|}\sum_{\iota \in I} \mathbb{1}[v_\iota(r) = 1]$.

4 Experimental Findings

We show results to answer: 1) whether *Task scores* correlate with *IF scores* (4.1), 2) whether prompt positional variations change performance (4.2), 3) whether models reject instructions inside documents (4.3). All experiments use AWS Bedrock Converse API[1] for LLM calls.

4.1 Task Success Does Not Imply Instruction Following

We selected recent models from different families, including both medium and large parameter counts. We report scores for one pass (one answer per prompt, per model, judged once, details in

[1] https://docs.aws.amazon.com/bedrock/latest/APIReference/API_runtime_Converse.html

Table 1: *Combined/Task/IF scores* (%) by context length for the USER-LAST placement, across all domain prompts. Combined $= 0.6\,\textit{Task Score} + 0.4\,\textit{IF Score}$, computed per response and then averaged; Cells are shaded by Combined within each row.

Model	Overall†	8k	16k	32k	64k	128k
Kimi-K2.5	40.0/60.0/10.0	44.7/65.0/14.3	39.8/60.6/8.7	42.7/63.3/11.7	38.0/56.7/10.0	35.2/55.3/5.0
Qwen3-80B	34.3/50.0/10.9	37.0/54.1/11.4	36.1/52.9/10.9	36.1/51.5/13.1	33.2/48.1/10.9	29.6/44.4/7.5
GLM-5	40.8/57.9/15.0	45.9/64.1/18.6	43.0/61.0/15.9	42.6/59.9/16.8	38.3/54.7/13.6	35.2/51.7/10.6
GLM-5-think	43.8/64.1/13.4	44.9/64.8/15.0	40.9/62.9/8.0	45.7/66.6/14.5	43.6/61.8/16.4	43.2/64.6/11.2
GPT-OSS-120B	37.6/54.6/12.3	42.5/59.4/17.1	38.0/56.1/10.9	39.2/57.3/12.1	36.6/51.9/13.6	32.5/49.2/7.5
Opus-4.8	45.4/64.9/16.3	47.5/66.7/18.6	45.9/67.4/13.8	46.9/66.7/17.3	45.7/62.3/20.9	40.8/62.3/8.7
Opus-4.8-think	47.4/67.4/17.5	49.1/69.0/19.3	45.8/65.8/15.9	49.7/70.4/18.7	45.8/62.7/20.5	46.5/69.7/11.8
Sonnet-5	44.1/63.8/14.5	49.4/69.4/19.3	43.4/64.7/11.6	44.9/63.3/17.3	42.9/61.5/15.0	40.6/61.8/8.7
Sonnet-5-think	44.1/64.4/13.7	47.4/67.5/17.1	42.7/62.5/13.0	44.3/65.2/13.1	45.0/63.5/17.3	41.0/63.4/7.5
Overall	42.0/60.8/13.7	45.4/64.5/16.7	41.7/61.5/12.1	43.6/62.7/15.0	41.0/58.1/15.4	38.3/58.1/8.7

Appendix §F). We evaluate per-domain by context length and IF intrinsic complexity, which we define as ranges of constraint counts (EASY contains 2–3 constraints, MEDIUM 4–5, and HARD 6+). We observe the following:

IF ability degrades for longer contexts. Table 1 shows the *Combined score* for different models across context windows. We observe reductions in *Combined scores* for all models ranging between $0.017 - 0.107$ (3.8%–23.3%), with a mean of 0.071 / 15.8%. *Task* and *IF scores* follow the same trend, though not strictly monotonic for all models (full complexity in Table 9 in Appendix §E).

IF score reveals failures that Task score obscures. Figure 3 compares aggregate *Task and IF scores* for matched model, complexity, and context-length cells. The mean Pearson correlation between the two means is 0.50. The separation is clearest in the hard group: mean *Task score* remains between 0.409 and 0.630, while mean *IF scores* range only from 0 to 0.085. Thus, models can continue to generate semantically correct responses for writing tasks even when almost no response satisfies the complete instruction set. Thus *Task and IF scores* are not interchangeable measures of model performance.

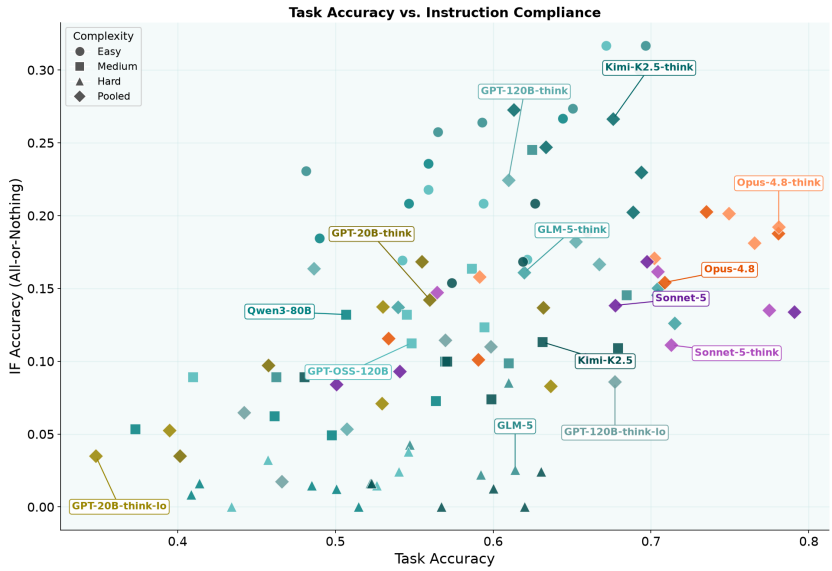

Figure 3: *Task score* vs. IF *score* for all models across context-length and complexity cells. Marker shape encodes complexity; color identifies the model. Models cluster along the x-axis at hard complexity, showing that task performance remains moderate while *IF score* is low.

4.2 POSITIONAL ROBUSTNESS TO PLACEMENT AND MULTI-TURN DELIVERY

We investigate how changing the location of user instructions in the prompt (Section 4.2.1) and delivering the prompts as multi-turn (Section 4.2.2) affects model performance.

Table 2: Placement sensitivity of *Combined*, *Task* and *IF* scores (%), baseline = USER-LAST; Mean ± std is taken over the four placements. Worst Δ and superscript is the placement with the largest drop vs baseline. See per-placement breakdown in Table 10. Significance test: unpaired two-sided z-test using std errors between best and worst scores (* $p < 0.1$, ** $p < 0.01$, *** $p < 0.001$).

	Combined			Task			IF		
Model	Base	Mean ± std	Worst Δ	Base	Mean ± std	Worst Δ	Base	Mean ± std	Worst Δ
Kimi-K2.5	40.0	39.7 ± 1.0	$-1.7^{m,*}$	60.0	58.6 ± 2.0	$-4.1^{m,***}$	10.0	11.4 ± 1.2	$+1.0^{d}$
Qwen3-80B	34.3	33.6 ± 1.6	$-3.1^{m,***}$	50.0	48.6 ± 2.4	$-4.8^{m,***}$	10.9	11.1 ± 0.6	-0.6^{m}
GLM-5	40.8	40.1 ± 1.8	$-3.4^{m,***}$	57.9	56.8 ± 2.3	$-4.6^{m,***}$	15.0	15.1 ± 1.2	-1.5^{m}
GLM-5-think	43.8	44.2 ± 0.6	-0.1^{d}	64.1	63.9 ± 0.3	-0.6^{d}	13.4	14.5 ± 1.4	$+0.6^{m}$
GPT-OSS-120B	37.6	37.1 ± 0.5	-1.2^{d}	54.6	53.5 ± 0.9	$-2.1^{m,*}$	12.3	12.5 ± 1.3	-1.4^{d}
Opus-4.8	45.4	44.5 ± 1.2	$-2.8^{m,***}$	64.9	63.3 ± 2.3	$-5.0^{m,***}$	16.3	16.2 ± 0.6	-0.9^{d}
Opus-4.8-think	**47.4**	47.4 ± 0.5	-0.5^{d}	67.4	67.4 ± 0.6	-0.7^{m}	17.5	17.4 ± 0.9	-1.4^{d}
Sonnet-5	44.1	41.4 ± 2.3	$-5.5^{m,***}$	63.8	60.2 ± 3.3	$-8.0^{m,***}$	14.5	13.2 ± 0.9	-1.8^{m}
Sonnet-5-think	44.1	42.5 ± 1.9	$-4.3^{m,***}$	64.4	61.8 ± 3.1	$-7.0^{m,***}$	13.7	13.5 ± 0.3	-0.7^{d}

4.2.1 ROBUSTNESS TO INSTRUCTION PLACEMENT

We vary the location of the user instructions in the prompt while preserving the contents in four conditions: LAST is the baseline placement, with the user prompt after the documents. FIRST places the user prompt before the document block. MIDDLE inserts the user prompt within the article list, ensuring it is after the target article so the positional reference is consistent. DISPERSED splits the user prompt at sentence boundaries into at most three phrases, preserving sentence order, and places them before, within, and after the documents. We compute the difference in *Combined score* relative to the paired LAST response to test sensitivity to layout while holding the prompt's content fixed.

All models are sensitive to instruction placement relative to the LAST baseline (Table 2). Over the four placements, 7/9 models score below their baseline, by -0.027 to -0.003. Worst-case decreases are model dependent and 5/9 differences are significant at $p \leq 0.001$. The worst placement is MIDDLE for six models and the distributed placement for the remaining three (per-placement breakdown in Table 10 in Appendix §E). Enabling thinking reduces score variance due to placement across all tested model families.

4.2.2 ROBUSTNESS TO TURN STRUCTURE

We compare single- and multi-turn TASK-FIRST scores by subtracting the former from the latter per-prompt, ignoring intermediate model answers. We observe that multi-turn scores are degraded overall (Figure 4), Table 3 reports differences by model comparing single and multi turn scores. Because the user-authored text and source prompt are fixed, we attribute the difference to delivering multi turn. We hypothesize that this translates to degraded IF in real-world multi-turn conversational interactions as also mentioned by Liu et al. (2023).

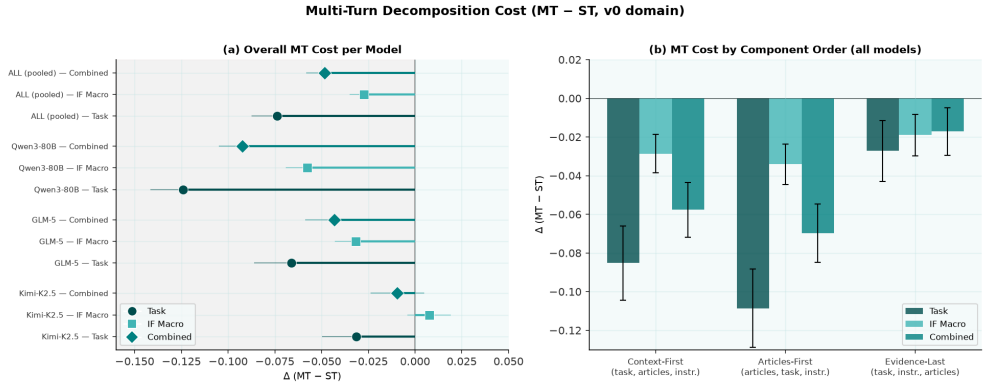

Figure 4: Paired multi-turn minus single-turn differences in *Task score*, *IF score*, and *Combined score*. Panel (a) groups differences by model; panel (b) groups them by component order.

Table 3: Multi-turn cost (MT−ST) on general-domain paired prompts. Negative = MT degrades. †: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$ (best vs. worst).

Model	Δ Task Score	Δ IF_{Macro}	Δ IF Score	Δ Combined Score
Kimi-K2.5	$-3.1_{\pm 1.9}{}^{***}$	$+0.8_{\pm 1.2}{}^{***}$	$+2.3_{\pm 2.1}{}^{*}$	$-0.9_{\pm 1.5}{}^{***}$
GLM-5	$-6.6_{\pm 2.0}$	$-3.2_{\pm 1.1}$	$-0.8_{\pm 2.2}$	$-4.3_{\pm 1.6}$
Qwen3-80B	$-12.4_{\pm 1.8}{}^{***}$	$-5.8_{\pm 1.2}{}^{***}$	$-4.5_{\pm 1.8}{}^{*}$	$-9.2_{\pm 1.3}{}^{***}$
All (pooled)	$-7.4_{\pm 1.4}$	$-2.7_{\pm 0.8}$	$-1.0_{\pm 1.3}$	$-4.8_{\pm 1.0}$

Table 4: Multi-turn (MT) scores by prompt order (average for general domain for all models).

Order	Δ Task Score	Δ IF Score	Δ Combined Score
Task-first	$-8.5_{\pm 1.9}$	$-2.9_{\pm 1.0}$	$-5.8_{\pm 1.5}$
Articles-first	$-10.9_{\pm 2.1}$	$-3.4_{\pm 1.1}$	$-7.0_{\pm 1.5}$
Articles-last	$-2.7_{\pm 1.6}$	$-1.9_{\pm 1.1}$	$-1.7_{\pm 1.2}$

Table 4 compares the three multi-turn orderings that permute the order in which the user delivers the task, instructions and documents. Consistent with the single-turn experiments, we observe that *Task score* and IF performance is not invariant to turn order.

4.3 Instructions in Context: Models Struggle to Preserve Hierarchy

We measure *IF scores* for instructions in the user prompt and for instructions that are between document tags (Table 5). All models follow instructions wrapped in $< article >$ tags, separating them as documents from system or user instructions. We observe that injected IF varies by data domain, with the lowest scores in cloud and legal, and the highest in finance and general (results are in Table 11 in the Appendix).

Instruction origin changes compliance: By comparing system vs user *IF scores*, we can measure if models prioritize system instructions. Table 7 (Appendix D) reports source-specific IF, and Table 8 reports each non-native placement relative to LAST. System-instruction IF exceeds user-instruction IF in all results. Relative to LAST, every other placement condition has higher system IF. This suggests that models adhere to instruction hierarchy, but the nonzero injection rates show models are susceptible to prompt injection attacks in long documents (as also mentioned in Li et al. (2024)).

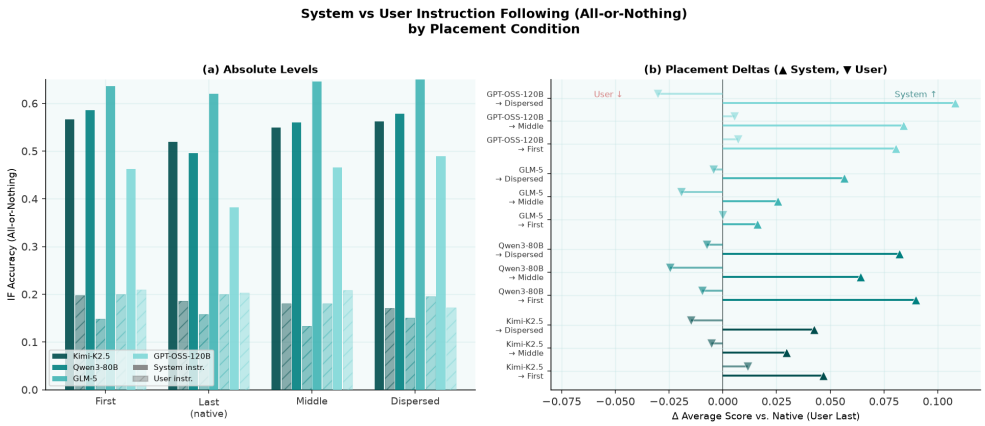

Figure 5: *IF scores* for identical constraints assigned to the system or user prompts. Panel (a) reports scores by placement; panel (b) reports paired differences by model and source.

5 Discussion: What Should IF Benchmarks Measure?

We use the findings above to motivate one case study and open questions for IF evaluations.

Table 5: Instruction-hierarchy robustness (5 domains pooled). IF-legit (↑): mean across domains ± std. dev. Injected (↓): mean across domains ± std. dev. Injected worst-case (↓): the highest injected per-domain score. Per-domain breakdown in Table 11 (Appendix).

Model	IF-legit ↑	Injected ↓	Injected worst-case ↓
Kimi-K2.5	50.7 ± 12.2	37.5 ± 12.4	56.0
Kimi-K2.5-think	65.2 ± 10.4	49.0 ± 15.9	71.0
Qwen3-80B	47.8 ± 9.1	32.4 ± 11.2	50.0
GLM-5	52.1 ± 12.4	33.0 ± 10.4	49.0
GLM-5-think	58.2 ± 11.8	37.7 ± 13.3	54.0
GPT-OSS-120B	55.9 ± 11.0	37.3 ± 11.5	53.0
Opus-4.8	57.6 ± 11.6	43.1 ± 16.4	64.0
Opus-4.8-think	57.3 ± 12.1	34.3 ± 11.9	53.0
Sonnet-5	57.3 ± 12.4	36.1 ± 12.2	53.0
Sonnet-5-think	55.4 ± 10.9	33.9 ± 8.6	48.0

Task specific dimensions: Discourse Coherence as a Case Study *Task score* measures whether a response contains the right information, while *IF score* measures whether it satisfies the stated constraints. Neither captures whether the response forms a coherent piece of writing. Discourse coherence—the property that segments of a text relate to one another through identifiable semantic and pragmatic connections—has been studied extensively in computational linguistics (Hobbs, 1979; Asher & Lascarides, 2003; Wolf & Gibson, 2005). Recent work has shown that coherence relations provide a useful framework for evaluating generated text (Alikhani et al., 2020; İnan et al., 2021), and that current LLMs encode discourse structure unevenly across layers and languages (Hedderich et al., 2021; Eichin et al., 2025). We measured discourse coherence on summarization tasks using rubrics adapted from this line of work (İnan et al., 2024; Atwell et al., 2022) (see Appendix §G for details). *Task and Discourse scores* are moderately correlated ($r = 0.65$ across models in Table 12), suggesting that writing quality is a separable dimension from *Task* and *IF scores*.

From Leaderboards to Diagnostic Profiles Throughout our experiments we observe tradeoffs in *Task and IF scores* depending on the positional condition. These differences are actionable for model development, but they complicate single model ranking. The community therefore faces a reporting question: how can benchmarks remain legible without collapsing distinct capabilities? One direction is a small set of predefined axes tied to explicit failure modes, accompanied by matched perturbation results and uncertainty. Scalar summaries can remain useful, but they should be derived views of a diagnostic profile. In this work we offer sub axes through robustness experiments.

Toward Composable Evaluation Science. Real prompts combine long contexts, many constraints, positional variation, and multi-turn delivery simultaneously, yet benchmark difficulty is typically scaled along one axis at a time. Our results show these factors interact multiplicatively and cannot be reduced to a single notion of "harder." MERIDIAN's controlled perturbations—byte-matched pairs, fixed-content placement permutations—provide attribution-preserving units of evidence, but they inherit the errors of their evaluators: deterministic checks can reject valid variants, and LLM judges exhibit the same instruction sensitivity being measured. A broader evaluation science will need coverage designs that compose difficulty while preserving attribution, together with explicit evaluator provenance and validation. We hope MERIDIAN demonstrates that separating task performance from constraint compliance, and testing each under controlled perturbation, reveals failures that no single aggregate score would surface. We release the benchmark to support this.

6 CONCLUSION

We introduce MERIDIAN to evaluate robust instruction following over long, real-world documents. The benchmark separates task performance from strict constraint satisfaction and uses controlled transformations to test robustness to instruction placement, turn structure, and instruction injection in data. Across experiments, we observe that longer contexts reduce ability of the model to complete the task correctly and follow the all the instruction constraints in the prompt. Results also show that models follow instructions injected in the documents when they shouldn't.

7 AI USE STATEMENT

In this work we have used generative AI to format Tables and Figures, to proof read for typos, and to check acronyms and terms are consistently defined and used. We have used AI generated text for early drafts of the Results section. We have fully reviewed and edited all text and figures manually for the submission.

8 ETHICS STATEMENT

Our contribution is a benchmark based on public documents, to which we add prompts that state text writing tasks. We did not manually validate each document included in the dataset, and as such we can not guarantee their suitability to all audiences if reviewed manually. We did however have all prompt data reviewed by human annotators and it did not contain purposefully offensive language.

9 REPRODUCIBILITY STATEMENT

We plan to release the code and data. In an effort to prevent contaminating future model training data with the benchmark, we will make a portion of the data available publicly and the rest we will distribute upon request.

REFERENCES

Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Marian Mazzone, David Randolph, Mohammad Mahmoudi Meymand, et al. PRBench: Large-scale expert rubrics for evaluating high-stakes professional reasoning. *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42297–42325, 2026. doi: 10.18653/v1/2026.acl-long.1958.

Malihe Alikhani, Piyush Sharma, Shengjie Li, Radu Soricut, and Matthew Stone. Cross-modal coherence modeling for caption generation. In *Proceedings of ACL*, pp. 6525–6535, 2020.

Chenxin An, Shansan Gong, Ming Zhong, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-Eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14388–14411. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.776.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, J. Q. Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. HealthBench: Evaluating large language models towards improved human health. *arXiv.org*, 2025. doi: 10.48550/arXiv.2505.08775.

Nicholas Asher and Alex Lascarides. *Logics of Conversation*. Cambridge University Press, 2003.

Katherine Atwell, Anthony Sicilia, Seong Jae Sung, and Malihe Alikhani. The change that matters in discourse parsing: Estimating the impact of domain shift on parser error. In *Findings of ACL*, 2022.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 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)*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.172.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, W. Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25:1–53, 2024. doi: 10.48550/arXiv.2210.11416.

Florian Eichin, Frances Yung, Michael A Hedderich, and Vera Demberg. Probing LLMs for multilingual discourse generalization through a unified label set. In *Proceedings of ACL*, 2025.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. LawBench: Benchmarking legal knowledge of large language models. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7933–7962, 2024. doi: 10.18653/v1/2024.emnlp-main.452.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin M. K. Peters, Brandon Waldon, D. Rockmore, et al. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Social Science Research Network*. Elsevier BV, 2023. doi: 10.2139/ssrn.4583531.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean M. Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv.org*, 2025. doi: 10.48550/arXiv.2507.17746.

Shirley Anugrah Hayati, Taehee Jung, Tristan Bodding-Long, Sudipta Kar, Abhinav Sethy, Joo-Kyung Kim, and Dongyeop Kang. Chain-of-instructions: Compositional instruction tuning on large language models. volume 39, pp. 24005–24013. Association for the Advancement of Artificial Intelligence (AAAI), 2025. doi: 10.1609/aaai.v39i22.34574.

Qi He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Lida Chen, Xintao Wang, Yuncheng Huang, et al. Can large language models understand real-world complex instructions? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18188–18196. Association for the Advancement of Artificial Intelligence (AAAI), 2024. doi: 10.1609/aaai.v38i16.29777.

Yun He, Wenzhe Li, Hejia Zhang, Songlin Li, Karishma Mandyam, Sopan Khosla, Yuanhao Xiong, Nanshu Wang, Xiaoliang Peng, Beibin Li, et al. AdvancedIF: Rubric-based benchmarking and reinforcement learning for advancing llm instruction following. *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18003–18022, 2026. doi: 10.18653/v1/2026.acl-long.820.

Michael A Hedderich, Nader Youssef, Lukas Lange, and Dietrich Klakow. Probing multilingual language models for discourse. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP)*, 2021.

Jerry R Hobbs. Coherence and coreference. *Cognitive Science*, 3(1):67–90, 1979.

Cheng-Yu Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *arXiv.org*, 2024. doi: 10.48550/arXiv.2404.06654.

Mert İnan, Piyush Sharma, Baber Khalid, Radu Soricut, Matthew Stone, and Malihe Alikhani. COSMic: A coherence-aware generation metric for image descriptions. In *Findings of EMNLP*, pp. 3391–3404, 2021.

Mert İnan, Katherine Atwell, Anthony B Sicilia, and Malihe Alikhani. Combining discourse coherence with large language models for more inclusive, equitable, and robust task-oriented dialogue. In *Proceedings of LREC-COLING*, 2024.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4667–4688. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.257.

Gemma Team Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, Louis Rouillard, et al. Gemma 3 technical report. *arXiv.org*, 2025. doi: 10.48550/arXiv.2503.19786.

Pei Ke, Fei Huang, Fei Mi, Yasheng Wang, Qun Liu, Xiaoyan Zhu, and Minlie Huang. DecompEval: Evaluating generated texts as unsupervised decomposed question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9676–9691. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.539.

Nathan Lambert, Jacob Daniel Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv.org*, 2024. doi: 10.48550/arXiv.2411.15124.

Keyu Li, Junhao Shi, Yang Xiao, Mohan Jiang, Jie Sun, Yunze Wu, Shijie Xia, Xiaojie Cai, Tianze Xu, Weiye Si, et al. AgencyBench: Benchmarking the frontiers of autonomous agents in 1M-token real-world contexts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7422–7440. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.337. URL https://arxiv.org/abs/2601.11044. arXiv:2601.11044.

Zekun Li, Baolin Peng, Pengcheng He, and Xifeng Yan. Evaluating the instruction-following robustness of large language models to prompt injection. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 557–568, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.33. URL https://aclanthology.org/2024.emnlp-main.33/.

Gabrielle Kaili-May Liu, Bowen Shi, Avi Caciularu, Idan Szpektor, and Arman Cohan. MDCure: A scalable pipeline for multi-document instruction-following. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29258–29296. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1418.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2023. doi: 10.1162/tacl_a_00638.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. OpenRubrics: Towards scalable synthetic rubric generation for reward modeling and LLM alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17417–17437. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.791.

Sushant Mehta, Liudas Panavas, Suhaas Garre, and Edwin Chen. ComplexConstraints and beyond: Expert rubrics for RLVR. In *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM)*, pp. 668–677. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.gem-main.61. arXiv:2606.09118.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, S. Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Neural Information Processing Systems*, volume 35, pp. 27730–27744. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-2011.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simon Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, et al. Gdpval: Evaluating ai model performance on real-world economically valuable tasks. In *International Conference on Learning Representations*, volume 2026, pp. 24005–24040, 2026.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems 38*, pp. 54870–54896. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-1645. URL https://arxiv.org/abs/2507.02833. arXiv:2507.02833.

Yanzhao Qin, Tao Zhang, Yanjun Shen, Wenjing Luo, Haoze Sun, Yan Zhang, Yujing Qiao, Weipeng Chen, Zenan Zhou, Wentao Zhang, et al. SysBench: Can large language models follow system messages? *arXiv.org*, 2024a. doi: 10.48550/arXiv.2408.10943.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InfoBench: Evaluating instruction following ability in

large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13025–13048. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.findings-acl.772.

Thomas Schuster, Marian Lambert, Nico Döring, and Julius Trögele. Needle-in-the-haystack testing llms with a complex reasoning task. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`, 2025.

Yu Shi, Huanghai Liu, Yiran Hu, Gaojie Song, Xinran Xu, Yubo Ma, Tianyi Tang, Li Zhang, Qingjing Chen, Di Feng, et al. PLawBench: A rubric-based benchmark for evaluating llms in real-world legal practice. *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10067–10116, 2026. doi: 10.18653/v1/2026.acl-long.458.

Tingyu Song, Guo Gan, Mingsheng Shang, and Yilun Zhao. IFIR: A comprehensive benchmark for evaluating instruction-following in expert-domain information retrieval. 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)*, pp. 10186–10204. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.511.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. Struc-Bench: Are large language models good at generating complex structured tabular data? In *North American Chapter of the Association for Computational Linguistics*, pp. 12–34. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-short.2.

Johanne R Trippas, Sara Fahad Dawood Al Lawati, Joel Mackenzie, and Luke Gallagher. What do users really ask large language models? an initial log analysis of google bard interactions in the wild. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2703–2707, 2024.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models. In *Advances in Neural Information Processing Systems 38*, pp. 127195–127221. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-3824. arXiv:2507.18624.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv.org*, 2024. doi: 10.48550/arXiv.2404.13208.

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, et al. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5627–5646. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.322.

Zhaoyang Wang, Jinqi Jiang, Huichi Zhou, Wenhao Zheng, Xuchao Zhang, Chetan Bansal, and Huaxiu Yao. Verifiable format control for large language model generations. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3499–3513. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.194.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, et al. Benchmarking complex instruction-following with multiple constraints composition. In *Advances in Neural Information Processing Systems 37*, pp. 137610–137645. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-4371.

Bosi Wen, Yilin Niu, Cunxiang Wang, Xiaoying Ling, Ying Zhang, Pei Ke, Hongning Wang, and Minlie Huang. IF-RewardBench: Benchmarking judge models for instruction-following evaluation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23816–23843. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1092.

Florian Wolf and Edward Gibson. Representing discourse coherence: A corpus-based study. *Computational linguistics*, 31(2):249–287, 2005.

Xiaodong Wu, Minhao Wang, Yichen Liu, Xiaoming Shi, He Yan, Xiangju Lu, Junmin Zhu, and Wei Zhang. LIFBench: Evaluating the instruction following performance and stability of large language models in long-context scenarios. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16445–16468. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.803. arXiv:2411.07037.

Yuhao Wu, Ming Shan Hee, Zhiqiang Hu, and Roy Ka-Wei Lee. LongGenBench: Benchmarking long-form generation in long context LLMs. *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2409.02076.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. FoFo: A benchmark to evaluate llms' format-following capability. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 680–699, 2024. doi: 10.18653/v1/2024.acl-long.40.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, et al. FinBen: A holistic financial benchmark for large language models. In *Neural Information Processing Systems*, pp. 95716–95743. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-3033. arXiv:2402.12659.

Zhuohan Xie, Dhruv Sahnan, Debopriyo Banerjee, Georgi N. Georgiev, Rushil Thareja, Hachem Madmoun, Jinyan Su, Aaryamonvikram Singh, Yuxia Wang, Rui Xing, et al. FinChain: A symbolic benchmark for verifiable chain-of-thought financial reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14529–14553. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.662. arXiv:2506.02515.

Xi Ye, Fangcong Yin, Yinghui He, Joie Zhang, Howard Yen, Tianyu Gao, Greg Durrett, and Danqi Chen. LongProc: Benchmarking long-context language models on long procedural generation. In *arXiv.org*, 2025. doi: 10.48550/arXiv.2501.05414. arXiv:2501.05414.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context language models effectively and thoroughly. In *arXiv.org*, 2025. doi: 10.48550/arXiv.2410.02694. arXiv:2410.02694.

Jingwei Yi, Yueqi Xie, Bin Zhu, Keegan Hines, Emre Kıcıman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and defending against indirect prompt injection attacks on large language models. *Knowledge Discovery and Data Mining*, pp. 1809–1820, 2023. doi: 10.1145/3690624.3709179.

Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, pp. 841–852, 2022.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, et al. ∞bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.814.

Zhihan Zhang, Shiyang Li, Zixuan Zhang, Xin Liu, Haoming Jiang, Xianfeng Tang, Yifan Gao, Zheng Li, Haodong Wang, Zhaoxuan Tan, et al. IHEval: Evaluating language models on following the instruction hierarchy. 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)*, pp. 8374–8398. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.425.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, E. Xing, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Neural Information Processing Systems*, volume 36, pp. 46595–46623. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-2020.

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.org*, 2023. doi: 10.48550/arXiv.2311.07911.

Jianan Zhou, Fleur Corbett, Joori Byun, Talya Porat, and Nejra van Zalk. A systematic review and meta-analysis of psychological and behavioural responses in human-agent vs. human-human interactions. *Communications Psychology*, 2026.

Tao Zou, Xinghua Zhang, Haiyang Yu, Minzheng Wang, Fei Huang, and Yongbin Li. Eifbench: Extremely complex instruction following benchmark for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20941–20964, 2025.

A Dataset statistics

We detail the composition of our benchmarks in Table 6.

	General	Legal	Medicine	Finance	AWS docs	All
Context length (tokens)						
$\leq$8k	94	103	25	102	164	488
8–16k	83	67	91	82	119	442
16–32k	99	154	159	91	159	662
32–64k	94	133	178	95	140	640
64–128k	101	141	197	100	131	670
$>$128k	0	0	0	229	0	229
Instructions per prompt						
2	0	54	63	123	69	309
3	2	109	128	125	165	529
4	67	42	55	99	75	338
5	104	87	94	108	115	508
6	170	120	105	69	117	581
7	91	63	50	63	56	323
8	35	63	74	41	59	272
9	1	47	64	44	46	202
10	1	13	17	27	11	69
Total	471	598	650	699	713	3131

Table 6: Composition of the benchmark corpus: prompt counts per domain by context length and by the total number of instructions in the system and user prompts. Context length is the token count in the system and user prompts.

B Annotation Protocol and Data Quality Details

Automated pre-filtering Synthesized prompts can fail in three ways: inaccurate paraphrase, infeasible instruction, or unintended constraint conflict. An LLM pipeline screens for all three before human review: (i) paraphrase fidelity checking with concept-drift rejection and meta-instruction leakage detection; (ii) constraint compatibility checking against both instruction descriptions and verifier source code. Two lexical gates (a leakage regex and a length-ratio gate rejecting ratios outside $[0.75, 4]$) run as additional filters.

Human annotation protocol Every surviving prompt is reviewed by a trained annotator who answers a fixed schedule: (i) are instructions feasible; (ii) does the paraphrase preserve meaning; (iii)

does it leak meta-instructions; (iv) do instructions conflict (and of which type); (v) which specific instructions are problematic. Negative judgments carry explanations. A QA pass cross-references discards against submissions to catch inconsistencies, with 10% spot-checks.

Triage and routing Annotations drive a deterministic split: accepted prompts enter the IF set (instructions jointly satisfiable), system–user conflicts enter the IH set (human-verified hierarchy violations), and paraphrase-inaccurate or user–user conflicts are discarded. A human override sheet adjudicates edge cases.

Feedback loop Discard patterns (by instruction template, paraphrase length ratio, and domain) are fed back as tighter pre-filtering rules and template revisions. The initial finance domain had a 53% discard rate; this dropped to 20% for medicine and 24% for AWS as fixes accumulated, and within AWS fell from 20% to 16% across three annotation batches.

Scoring reliability Verification and rubric scoring share a single code path (inline and post-hoc scores are identical by construction). Unparseable judge verdicts count as failures; $\perp$ verdicts are excluded from denominators.

Verification Reliability Instruction-following evaluation has started to incorporate model judges with rubrics as task complexity has exceeded the capabilities of code validation (He et al., 2026). This creates a quality-assurance problem at two levels: whether each verifier implements its instruction correctly, and whether the collection of verifiers measures the intended construct without systematic blind spots. One-off agreement estimates are insufficient when the rubric, verifier, and challenge set evolve together and are unreliable. Our approach is based on inter- and intra- judge agreement over multiple answers. We plan to extend this approach to include human-labels on rubrics, prioritizing annotations that are not converging statistically.

Secondary scoring formulas Robustness degradation: $\Delta_{\text{rob}} = S_{\text{BASELINE}} - \min_{\pi \in \Pi} S_\pi$, where a robust model has $\Delta_{\text{rob}} \approx 0$.

Hierarchy delta: $\delta_{\text{IH}} = \text{IF}_\mu^{\text{IH}}(r, I_s) - \text{IF}_\mu^{\text{IF}}(r, I_s)$, where negative values indicate the model abandons system instructions under conflict.

C Scoring Formulas

Task score. LLM judge scores K criteria on a 1–5 scale:

$$\bar{q}(r) = \frac{1}{K} \sum_{k=1}^{K} \frac{q_k - 1}{4} \in [0, 1]. \tag{3}$$

Injection rate. Fraction of adversarial instructions obeyed (lower is better):

$$\rho_{\text{inj}}(r) = \frac{1}{|I'|} \sum_{\iota' \in I'} \mathbb{1}[v_{\iota'}(r) = 1]. \tag{4}$$

Problem formulation. A benchmark instance is $x = \rho(s, u, C, \pi)$ where s is a system prompt, u a user request, $C = (d_1, \ldots, d_m)$ an ordered document set filling a target window W, π a layout, and ρ a template renderer. Each instruction ι induces $v_\iota(r) \in \{1, 0, \perp\}$.

D Instruction Source Analysis

Tables 7 and 8 report per-source *IF scores* for the four original open-weight models. These results demonstrate the system–user compliance gap discussed in Section 4.3.

E Detailed Result Tables

The main text reports simplified summaries. The full breakdowns are below.

Table 7: *IF average score* by instruction source (system prompt vs. user message) and placement condition. †: significantly different from second-ranked model at $p < 0.05$.

Model	Placement	System IF	User IF
Kimi-K2.5	First	$0.709_{\pm 0.012}$	$0.493_{\pm 0.011}$
	Last	$0.672_{\pm 0.013}$	$0.482_{\pm 0.011}$
	Middle	$0.691_{\pm 0.013}$	$0.462_{\pm 0.011}$
	Dispersed	$0.705_{\pm 0.013}$	$0.464_{\pm 0.011}$
Qwen3-80B	First	$0.738_{\pm 0.012}$	$0.429_{\pm 0.011}^{***}$
	Last	$0.658_{\pm 0.013}$	$0.451_{\pm 0.011}^{***}$
	Middle	$0.710_{\pm 0.012}$	$0.412_{\pm 0.011}^{***}$
	Dispersed	$0.727_{\pm 0.012}$	$0.412_{\pm 0.011}^{***}$
GLM-5	First	$0.771_{\pm 0.012}^{***}$	$0.498_{\pm 0.011}$
	Last	$0.757_{\pm 0.012}^{***}$	$0.497_{\pm 0.011}$
	Middle	$0.778_{\pm 0.011}^{***}$	$0.454_{\pm 0.012}$
	Dispersed	$0.794_{\pm 0.011}^{***}$	$0.470_{\pm 0.012}^{***}$
GPT-OSS-120B	First	$0.629_{\pm 0.012}^{***}$	$0.517_{\pm 0.010}^{***}$
	Last	$0.542_{\pm 0.012}^{***}$	$0.516_{\pm 0.010}^{***}$
	Middle	$0.618_{\pm 0.012}^{***}$	$0.513_{\pm 0.010}^{***}$
	Dispersed	$0.644_{\pm 0.012}^{***}$	$0.463_{\pm 0.010}$

Table 8: Placement deltas in system and user *IF score* vs. native (user-last). Non-native placements generally boost system compliance while degrading user compliance.

Model	Placement	Δ Sys	Δ User
Kimi-K2.5	First	$+0.037$	$+0.011$
	Middle	$+0.019$	-0.019
	Dispersed	$+0.034$	-0.018
Qwen3-80B	First	$+0.080$	-0.022
	Middle	$+0.052$	-0.039
	Dispersed	$+0.070$	-0.039
GLM-5	First	$+0.014$	$+0.001$
	Middle	$+0.021$	-0.043
	Dispersed	$+0.037$	-0.027
GPT-OSS-120B	First	$+0.087$	$+0.001$
	Middle	$+0.077$	-0.003
	Dispersed	$+0.102$	-0.053

Context Length × Complexity Table 9 reports Combined Score for every (model, complexity, context bucket) cell.

Placement Robustness Table 10 reports per-placement scores for all models.

Instruction Hierarchy by Domain Table 11 reports per-domain IF-legit and Injected rates.

F Experimental Protocol Details

Context-length and complexity grid. The standard single-turn analysis uses Kimi-K2.5, Qwen3-80B, GLM-5, and GPT-OSS-120B. These four independently developed model families form the largest complete-case panel with outputs for the same prompts across every context bucket and complexity stratum; restricting the comparison to complete cells avoids changing model composition across the grid. Each prompt contains one system message with task framing and system constraints, followed by one user message with the ordered documents, paraphrased request, and user constraints. A target document identified by ordinal position is surrounded by same-domain

Table 9: Full *Combined Score* by instruction complexity and context length (primary domains, native single-turn layout).

Model	Complexity	8k	16k	32k	64k	128k
Kimi-K2.5	easy	$0.536_{\pm 0.032}$	$0.459_{\pm 0.032}$	$0.471_{\pm 0.024}$	$0.439_{\pm 0.026}$	$0.406_{\pm 0.032}$
	medium	$0.451_{\pm 0.024}$	$0.424_{\pm 0.031}$	$0.389_{\pm 0.026}$	$0.382_{\pm 0.024}$	$0.324_{\pm 0.030}$
	hard	$0.388_{\pm 0.018}$	$0.365_{\pm 0.021}$	$0.372_{\pm 0.016}$	$0.320_{\pm 0.017}$	$0.340_{\pm 0.018}$
Qwen3-80B	easy	$0.493_{\pm 0.030}$	$0.411_{\pm 0.032}$	$0.430_{\pm 0.024}$	$0.387_{\pm 0.025}$	$0.368_{\pm 0.032}$
	medium	$0.367_{\pm 0.022}$	$0.357_{\pm 0.029}$	$0.318_{\pm 0.021}$	$0.302_{\pm 0.022}$	$0.245_{\pm 0.025}$
	hard	$0.305_{\pm 0.016}$	$0.309_{\pm 0.019}$	$0.297_{\pm 0.013}$	$0.255_{\pm 0.015}$	$0.249_{\pm 0.015}$
GLM-5	easy	$0.545_{\pm 0.029}$	$0.461_{\pm 0.033}$	$0.500_{\pm 0.025}$	$0.442_{\pm 0.028}$	$0.381_{\pm 0.034}$
	medium	$0.469_{\pm 0.025}$	$0.473_{\pm 0.030}$	$0.405_{\pm 0.025}$	$0.382_{\pm 0.027}$	$0.313_{\pm 0.029}$
	hard	$0.400_{\pm 0.019}$	$0.378_{\pm 0.019}$	$0.364_{\pm 0.016}$	$0.320_{\pm 0.017}$	$0.345_{\pm 0.019}$
GPT-OSS-120B	easy	$0.530_{\pm 0.032}$	$0.440_{\pm 0.032}$	$0.441_{\pm 0.023}$	$0.422_{\pm 0.029}$	$0.393_{\pm 0.031}$
	medium	$0.417_{\pm 0.028}$	$0.380_{\pm 0.035}$	$0.406_{\pm 0.026}$	$0.374_{\pm 0.024}$	$0.282_{\pm 0.028}$
	hard	$0.334_{\pm 0.018}$	$0.343_{\pm 0.021}$	$0.322_{\pm 0.015}$	$0.287_{\pm 0.017}$	$0.261_{\pm 0.016}$

Table 10: Per-placement scores (%) and paired deltas versus USER-LAST for every reported model; *IF Score* is all-or-nothing: a prompt scores 1 only when every verifiable instruction in it was followed. Scores pool 4 domains (legal, medicine, finance, AWS). Δ columns are within-prompt paired differences on the prompts that model answered in all four placements; stars mark a two-sided paired test: $^{*}p \leq 0.05$, $^{**}p \leq 0.01$, $^{***}p \leq 0.001$.

		Combined		Task Score		IF Score		
Model	Placement	Score	Δ	Score	Δ	Score	Δ	n
Kimi-K2.5	first	40.0	0.0	58.3	-1.8^{*}	12.7	$+2.7^{**}$	873
	last	40.0	–	60.0	–	10.0	–	873
	middle	38.3	-1.7^{*}	55.9	-4.1^{***}	11.9	$+1.9^{*}$	873
	dispersed	40.5	$+0.5$	60.3	$+0.2$	11.0	$+1.0$	873
Qwen3-80B	first	33.8	-0.5	48.6	-1.4	11.6	$+0.7$	873
	last	34.3	–	50.0	–	10.9	–	873
	middle	31.3	-3.1^{***}	45.2	-4.8^{***}	10.3	-0.6	873
	dispersed	34.9	$+0.5$	50.5	$+0.5$	11.5	$+0.6$	873
GLM-5	first	40.9	$+0.1$	57.7	-0.2	15.6	$+0.6$	873
	last	40.8	–	57.9	–	15.0	–	873
	middle	37.4	-3.4^{***}	53.3	-4.6^{***}	13.5	-1.5	873
	dispersed	41.4	$+0.6$	58.1	$+0.2$	16.3	$+1.3$	873
GLM-5-think	first	45.1	$+1.2^{*}$	64.1	-0.1	16.6	$+3.2^{**}$	873
	last	43.8	–	64.1	–	13.4	–	873
	middle	43.9	$+0.1$	63.9	-0.3	14.0	$+0.6$	873
	dispersed	43.8	-0.1	63.5	-0.6	14.1	$+0.7$	873
GPT-OSS-120B	first	37.2	-0.4	53.4	-1.2	13.1	$+0.8$	873
	last	37.6	–	54.6	–	12.3	–	873
	middle	37.0	-0.6	52.5	-2.1^{*}	13.9	$+1.6$	873
	dispersed	36.5	-1.2	53.5	-1.1	10.9	-1.4	873
Opus-4.8	first	44.8	-0.6	63.9	-1.0	16.3	0.0	873
	last	45.4	–	64.9	–	16.3	–	873
	middle	42.7	-2.8^{***}	59.9	-5.0^{***}	16.8	$+0.6$	873
	dispersed	44.9	-0.5	64.6	-0.2	15.3	-0.9	873
Opus-4.8-think	first	48.2	$+0.8$	68.2	$+0.9$	18.1	$+0.6$	873
	last	47.4	–	67.4	–	17.5	–	873
	middle	47.2	-0.2	66.7	-0.7	18.0	$+0.5$	873
	dispersed	46.9	-0.5	67.4	0.0	16.2	-1.4	873
Sonnet-5	first	41.4	-2.7^{***}	60.4	-3.4^{***}	12.8	-1.7	873
	last	44.1	–	63.8	–	14.5	–	873
	middle	38.6	-5.5^{***}	55.8	-8.0^{***}	12.7	-1.8	873
	dispersed	41.7	-2.4^{***}	60.9	-2.9^{**}	12.8	-1.7	873
Sonnet-5-think	first	42.8	-1.3^{*}	62.4	-2.0^{*}	13.5	-0.2	873
	last	44.1	–	64.4	–	13.7	–	873
	middle	39.8	-4.3^{***}	57.4	-7.0^{***}	13.5	-0.2	873
	dispersed	43.1	-1.0	63.1	-1.3	13.1	-0.7	873

distractors until the context reaches 8k, 16k, 32k, 64k, or 128k tokens. The model–complexity–length cells contain 53–136 prompts, with identical prompt counts across models.

Complexity is assigned before evaluation from the total number of active system and user constraints: EASY contains 2–3, MEDIUM 4–5, and HARD 6 or more. The bins measure instruction load

Table 11: Full instruction-hierarchy robustness breakdown by domain. IF-legit (↑): real instructions followed. Injected (↓): embedded instructions obeyed.

Domain	Model	IF-legit ↑	Injected ↓
General	Kimi-K2.5-think	$0.797_{\pm 0.013}$	$0.715_{\pm 0.024}$
	GLM-5-think	$0.704_{\pm 0.015}$	$0.543_{\pm 0.025}$
	GPT-OSS-120B	$0.704_{\pm 0.015}$	$0.482_{\pm 0.025}$
	Opus-4.8	$0.668_{\pm 0.015}$	$0.602_{\pm 0.025}$
	Opus-4.8-think	$0.674_{\pm 0.015}$	$0.384_{\pm 0.025}$
	Sonnet-5	$0.681_{\pm 0.014}$	$0.467_{\pm 0.025}$
	Sonnet-5-think	$0.652_{\pm 0.014}$	$0.353_{\pm 0.024}$
Legal	Kimi-K2.5	$0.438_{\pm 0.019}$	$0.348_{\pm 0.028}$
	Kimi-K2.5-think	$0.562_{\pm 0.016}$	$0.374_{\pm 0.024}$
	Qwen3-80B	$0.422_{\pm 0.015}$	$0.290_{\pm 0.021}$
	GLM-5	$0.482_{\pm 0.015}$	$0.322_{\pm 0.022}$
	GLM-5-think	$0.507_{\pm 0.016}$	$0.314_{\pm 0.023}$
	GPT-OSS-120B	$0.472_{\pm 0.015}$	$0.311_{\pm 0.022}$
	Opus-4.8	$0.471_{\pm 0.016}$	$0.311_{\pm 0.022}$
	Opus-4.8-think	$0.501_{\pm 0.017}$	$0.303_{\pm 0.023}$
	Sonnet-5	$0.492_{\pm 0.016}$	$0.281_{\pm 0.021}$
	Sonnet-5-think	$0.499_{\pm 0.016}$	$0.294_{\pm 0.022}$
Medicine	Kimi-K2.5	$0.491_{\pm 0.016}$	$0.388_{\pm 0.024}$
	Kimi-K2.5-think	$0.596_{\pm 0.016}$	$0.381_{\pm 0.025}$
	Qwen3-80B	$0.463_{\pm 0.016}$	$0.324_{\pm 0.024}$
	GLM-5	$0.484_{\pm 0.016}$	$0.319_{\pm 0.023}$
	GLM-5-think	$0.523_{\pm 0.018}$	$0.321_{\pm 0.024}$
	GPT-OSS-120B	$0.519_{\pm 0.017}$	$0.323_{\pm 0.024}$
	Opus-4.8	$0.544_{\pm 0.016}$	$0.387_{\pm 0.024}$
	Opus-4.8-think	$0.529_{\pm 0.018}$	$0.327_{\pm 0.025}$
	Sonnet-5	$0.528_{\pm 0.020}$	$0.324_{\pm 0.028}$
	Sonnet-5-think	$0.524_{\pm 0.016}$	$0.353_{\pm 0.024}$
Finance	Kimi-K2.5	$0.709_{\pm 0.018}$	$0.555_{\pm 0.029}$
	Qwen3-80B	$0.631_{\pm 0.019}$	$0.496_{\pm 0.031}$
	GLM-5	$0.726_{\pm 0.018}$	$0.486_{\pm 0.030}$
	GLM-5-think	$0.740_{\pm 0.019}$	$0.516_{\pm 0.032}$
	GPT-OSS-120B	$0.673_{\pm 0.025}$	$0.529_{\pm 0.038}$
	Opus-4.8	$0.749_{\pm 0.017}$	$0.639_{\pm 0.028}$
	Opus-4.8-think	$0.749_{\pm 0.021}$	$0.534_{\pm 0.036}$
	Sonnet-5	$0.750_{\pm 0.017}$	$0.535_{\pm 0.029}$
	Sonnet-5-think	$0.699_{\pm 0.019}$	$0.478_{\pm 0.030}$
Cloud	Kimi-K2.5	$0.392_{\pm 0.026}$	$0.208_{\pm 0.029}$
	Qwen3-80B	$0.397_{\pm 0.019}$	$0.184_{\pm 0.022}$
	GLM-5	$0.391_{\pm 0.025}$	$0.193_{\pm 0.029}$
	GLM-5-think	$0.437_{\pm 0.026}$	$0.193_{\pm 0.031}$
	GPT-OSS-120B	$0.426_{\pm 0.018}$	$0.221_{\pm 0.022}$
	Opus-4.8	$0.447_{\pm 0.019}$	$0.217_{\pm 0.023}$
	Opus-4.8-think	$0.415_{\pm 0.022}$	$0.166_{\pm 0.024}$
	Sonnet-5	$0.414_{\pm 0.024}$	$0.201_{\pm 0.029}$
	Sonnet-5-think	$0.398_{\pm 0.019}$	$0.217_{\pm 0.024}$

rather than retrospective model performance. A shared Python runner serializes the same prompt for each model and invokes model-specific endpoints through a shared registry layered over a common Bedrock utility module. The benchmark text is not rewritten by model. Responses follow one scoring path: task-specific rubric criteria are scored from 1–5 and normalized to $[0, 1]$; fixed code, LLM (we use Sonnet 4.5 throughout for judging), or hybrid verifiers score each applicable instruction; strict IF is 1 only if every verifier passes; and Combined is computed per response before aggregation. The main tables report means with standard errors.

Placement rendering. The placement experiment evaluates 1,156 source prompts for each of the same four models under four layouts. LAST leaves the complete request after the documents. FIRST moves it before the document block. MIDDLE splits the ordered articles at the midpoint and inserts the complete request between the halves. DISPERSED splits the request at sentence boundaries into at most three contiguous groups, preserves sentence order, and places the groups before, within, and after the documents. The deterministic renderer retains the system message, article order and bytes, target ordinal, request bytes, and instruction privilege; each variant records its source-prompt identifier. One response is generated per model, prompt, and placement, and each delta is computed against that prompt's LAST response.

Multi-turn conversion. The multi-turn study uses Kimi-K2.5, GLM-5, and Qwen3-80B, the models with complete paired runs for every component order, on general-domain prompts. Each model–order cell contains approximately 465–471 pairs. The pooled analysis contains 4,225 pairs for *Task and Combined score* and 4,239 for *micro-IF score* and strict *IF score*; the difference comes from responses for which the task evaluator returned no usable score. Both members of a pair leave the API system field empty, converting the original system content into the user-role Taskcomponent. They contain identical user-authored Task (T), Documents (D), three instruction utterances (I_1, I_2, I_3), and final request (G) in the same order. Multi-turn sends the components separately and retains each intermediate model reply in the history; single-turn concatenates the user-authored bytes into one message. Only the response to G is scored. Records are matched by source prompt, model, and order, and tables report the mean paired multi-turn minus single-turn difference with its standard error.

Distance control. The turn-distance ablation uses the same prompts across $K \in \{0, 1, 2, 4, 8, 16\}$ content-free acknowledgment exchanges. In the distance arm, documents, task, and instructions occur in the first user message, followed by the K exchanges and a fixed generation request that refers to the earlier requirements. In the position-control arm, documents occur first, the same exchanges intervene, and task and instructions are placed in the final user message adjacent to generation. The padding adds no task-relevant evidence or constraints, and only the final response is scored.

G DISCOURSE COHERENCE CASE STUDY

Motivation and scope. *Task score* and strict *IF score* do not directly measure whether a long-form answer is organized and readable. We therefore evaluate discourse as a diagnostic case study rather than a universal benchmark axis. The analysis is post-hoc. It scores existing responses and joins the resulting discourse judgments to previously computed *Task score*s without regenerating answers or rerunning the *Task score* and *IF score* evaluators.

Evaluation population. We restrict the study to eligible single-turn summarization answers and pool five domains (general, legal, medicine, finance, and AWS) and five prompt settings (user-first, user-last, user-middle, dispersed, and in-document injection). Empty answers, records missing required task fields, and prompts whose active constraints require numbered or bulleted lists, JSON, or one bullet per entity are ineligible because those constraints prescribe fragmented output. The resulting candidate pool contains 13,258 answers. A valid discourse judgment was recovered for 12,827 answers. For reporting, we retain model variants with at least 100 valid judgments. This gives 12,657 scored answers from 11 variants: GPT-OSS-20B-think-lo, GPT-OSS-120B, Qwen3-80B, GLM-5, GLM-5-think, Kimi-K2.5, Kimi-K2.5-think, Opus-4.8, Opus-4.8-think, Sonnet-5, and Sonnet-5-think. The per-model discourse sample sizes range from 147 to 1,416. Three additional variants with 35–90 valid judgments fall below the reporting threshold. Counts and prompt coverage are therefore not identical across models.

Discourse rubric. A judge assigns an independent 1–5 Likert score to each of five dimensions. *Topic continuity* measures given–new information flow and whether topic shifts are motivated. *Referential clarity* measures whether entities, pronouns, and other references are unambiguous. *Logical progression* measures whether claims are ordered and build on prior content. *Rhetorical cohesion* measures the clarity of discourse relations and connective use. *Structural integration* measures whether the answer forms integrated prose rather than a disconnected checklist. These dimensions target organization and readability, not factual correctness or constraint satisfaction.

Table 12: Discourse coherence by model (single-turn, all domains pooled). Coherence is the mean of five sub-dimensions. Teal : best; pink : worst. Significance (best vs. worst): $^{*}p<0.05$, $^{**}p<0.01$, $^{***}p<0.001$.

Model	Coherence	Topic	Referential	Logical	Rhetorical	Structural	Task
GPT-OSS-20B-think-lo	$0.487_{\pm 0.017}^{***}$	$0.543_{\pm 0.022}^{***}$	$0.594_{\pm 0.014}$	$0.578_{\pm 0.019}^{***}$	$0.398_{\pm 0.017}$	$0.325_{\pm 0.032}^{***}$	$0.648_{\pm 0.026}$
Sonnet-5	$0.487_{\pm 0.005}$	$0.508_{\pm 0.007}$	$0.640_{\pm 0.004}$	$0.552_{\pm 0.007}$	$0.461_{\pm 0.005}^{***}$	$0.276_{\pm 0.009}$	$0.584_{\pm 0.011}$
Sonnet-5-think	$0.484_{\pm 0.006}$	$0.503_{\pm 0.007}$	$0.636_{\pm 0.004}$	$0.548_{\pm 0.007}$	$0.452_{\pm 0.005}$	$0.280_{\pm 0.009}$	$0.642_{\pm 0.011}$
Opus-4.8-think	$0.482_{\pm 0.005}$	$0.507_{\pm 0.007}$	$0.641_{\pm 0.004}^{***}$	$0.551_{\pm 0.006}$	$0.434_{\pm 0.005}$	$0.278_{\pm 0.009}$	$0.749_{\pm 0.010}^{***}$
Opus-4.8	$0.473_{\pm 0.005}$	$0.495_{\pm 0.006}$	$0.638_{\pm 0.004}$	$0.539_{\pm 0.006}$	$0.434_{\pm 0.005}$	$0.258_{\pm 0.009}$	$0.627_{\pm 0.011}$
Kimi-K2.5-think	$0.466_{\pm 0.006}$	$0.495_{\pm 0.008}$	$0.612_{\pm 0.005}$	$0.532_{\pm 0.008}$	$0.399_{\pm 0.006}$	$0.290_{\pm 0.011}$	$0.725_{\pm 0.010}$
GLM-5	$0.439_{\pm 0.006}$	$0.456_{\pm 0.007}$	$0.605_{\pm 0.004}$	$0.492_{\pm 0.007}$	$0.388_{\pm 0.005}$	$0.254_{\pm 0.010}$	$0.564_{\pm 0.011}$
GLM-5-think	$0.430_{\pm 0.005}$	$0.448_{\pm 0.007}$	$0.593_{\pm 0.004}$	$0.496_{\pm 0.006}$	$0.374_{\pm 0.005}$	$0.240_{\pm 0.009}$	$0.695_{\pm 0.010}$
Kimi-K2.5	$0.417_{\pm 0.006}$	$0.433_{\pm 0.007}$	$0.599_{\pm 0.005}$	$0.465_{\pm 0.008}$	$0.377_{\pm 0.006}$	$0.212_{\pm 0.009}$	$0.603_{\pm 0.012}$
Qwen3-80B	$0.391_{\pm 0.006}$	$0.399_{\pm 0.008}$	$0.552_{\pm 0.005}$	$0.424_{\pm 0.008}$	$0.369_{\pm 0.007}$	$0.212_{\pm 0.010}$	$0.478_{\pm 0.011}^{***}$
GPT-OSS-120B	$0.366_{\pm 0.006}^{***}$	$0.368_{\pm 0.007}^{***}$	$0.549_{\pm 0.005}^{***}$	$0.411_{\pm 0.007}^{***}$	$0.325_{\pm 0.005}^{***}$	$0.175_{\pm 0.008}^{***}$	$0.523_{\pm 0.010}$

Judge protocol and score construction. The displayed results use one Sonnet-4.5 judge in answer-only mode. In addition to the rubric and output instructions, the judge receives the candidate answer but not the source documents, task description, or active-constraint summary. It must return JSON containing all five dimension keys. Values are rounded to integers and clipped to $[1, 5]$. Parsing is attempted up to three times, and an answer is omitted when no attempt yields a complete valid judgment. For dimension d, a raw score s_d is normalized as

$$D_d = \frac{s_d - 1}{4} \in [0, 1]. \tag{5}$$

Overall coherence is the equally weighted mean $C = \frac{1}{5}\sum_{d=1}^{5} D_d$. Thus each dimension contributes 0.2 and no weight is fitted from the observed results.

Aggregation and statistical reporting. Table 12 reports response-level means with all available domains and settings pooled. Each $\pm$ value is the standard error $s/\sqrt{n}$ computed from the scored responses for that model and metric. *Task scores* are available for 145–1,102 responses per reported model, compared with 147–1,416 for discourse, so the *Task score* and *Discourse Score* means use their respective non-missing cases. The model-level association reported in Section 5 is the unweighted Pearson correlation across the 11 pairs of model means ($r = 0.646$, rounded to 0.65). Within each table column, significance markers compare the highest and lowest displayed means with an unpaired two-sided z-test,

$$z = \frac{|\bar{x}_{\text{best}} - \bar{x}_{\text{worst}}|}{\sqrt{\text{SE}_{\text{best}}^2 + \text{SE}_{\text{worst}}^2}}. \tag{6}$$

The reported p-values are not adjusted for multiple comparisons.

Interpretation and limitations. The answer-only design isolates surface discourse quality but cannot test factual grounding or determine whether a structural choice was required by the source or instructions. A single judge also provides no estimate of inter-judge reliability. Unequal model coverage, uneven domain and setting frequencies, and failed judgments can change the pooled composition. Response-level standard errors treat rows as independent even when conditions may share a source prompt, while the unpaired tests do not exploit prompt matching. Finally, the correlation has only 11 model-level observations. We therefore use the study to show that discourse can change model rankings, not to establish an independent latent capability or a general-purpose leaderboard.