Joint Evaluation of Compliance, Planning, and Consistency under Paraphrase: A Relational-Complexity View of Frontier LLMs

Shivansh Bibra [1] **Dhruv Kumar** [2] **Murari Mandal** [3] **Yash Sinha** [4]

Abstract

We introduce a 971-prompt benchmark that jointly evaluates three properties of frontier large language models (LLMs) under controlled prompt variation: (i) instruction compliance as the number of formatting constraints scales, (ii) multi-step planning as reasoning depth grows, and (iii) within-model lexical consistency across paraphrased restatements of the same underlying question. We evaluate three frontier systems—GPT-4o, Gemini, and DeepSeek—over 2,913 real API responses on a closed factual domain. Compliance is low and appears complexity-sensitive (means 0.46–0.56; exploratory Wilcoxon $p<0.05$ in all three models); planning accuracy is non-monotonic in step count, with DeepSeek leading at every depth (0.76–0.89); and within-model paraphrase Jaccard averages only $\sim$0.27, indicating substantial surface-form variability across paraphrases. Cross-model Jaccard suggests that GPT-4o and Gemini produce more lexically similar outputs to each other (0.512) than either does to DeepSeek, which may be consistent with convergent alignment behaviour among proprietary models. We interpret these findings through the lens of Relational Complexity (RC) as a conceptual framing, drawing on the REL benchmark of Fesser et al. (2026), and report an exploratory task-vs-form pattern in which the strongest performer on task accuracy is the least lexically stable under paraphrase. The benchmark, prompts, responses, and analysis pipeline are released for reproducibility.

[1]Birla Institute of Technology and Science, Pilani. f20230513@pilani.bits-pilani.ac.in
[2]Birla Institute of Technology and Science, Pilani. dhruv.kumar5@pilani.bits-pilani.ac.in
[3]Birla Institute of Technology and Science, Pilani. f20211065@pilani.bits-pilani.ac.in
[4]Birla Institute of Technology and Science, Pilani. yash.sinha@pilani.bits-pilani.ac.in. Correspondence to: Shivansh Bibra <f20230513@pilani.bits-pilani.ac.in>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Frontier LLMs are now routinely deployed in pipelines that demand three properties simultaneously: they must obey explicit formatting rules, execute multi-step plans, and return stable answers when the same intent is rephrased. Most public benchmarks isolate one of these axes—MMLU and GSM8K probe knowledge and arithmetic reasoning, IFEval (7) probes instruction following, and prior work has extended this to instruction density (7)—but real deployment failures arise from their interaction. A user may write either "list the four Hogwarts houses" or "name every house at Hogwarts"; both express the same intent, yet a downstream parser fed the two responses can fail on one and succeed on the other. The cost of such failures is paid not by knowledge gaps but by surface instability.

A recent line of work argues that the bottleneck in LLM reasoning is not data, prompt length, or entity count, but rather the number of independent entities that must be jointly bound to satisfy a single relation. Fesser et al. (2026) formalise this quantity as Relational Complexity (RC) and show, on the REL benchmark of algebraic, biological, and chemical relational tasks, that frontier model accuracy can drop substantially as RC rises—even when input length and entity count are held fixed (3). Their conclusion is sharp: more compute, more prompting, and even tool use do not necessarily fix high-RC failure. The phenomenon is structural.

This paper asks a complementary question: might relational complexity serve as a useful conceptual lens for LLM degradation in natural-language tasks where the relation is encoded in instructions rather than in mathematical or biochemical structure? We suggest that an instruction stacking k formatting rules may implicitly define an RC-k problem on the surface form, because every rule must be jointly satisfied to produce a compliant response. Likewise, paraphrase consistency tests whether the model has internalised a single relational structure or a family of surface-coupled approximations. We therefore design a benchmark that varies these

three RC-related axes over a single closed factual domain (the four Hogwarts houses), eliminating knowledge-retrieval as a confound and isolating relational reasoning at the language level.

Our 971-prompt benchmark spans three categories—instructability (340 prompts, 1–4 jointly-bound formatting constraints), planning (331 prompts, 2–7 sequential steps), and consistency (300 prompts, 30 paraphrase groups of 10 prompts each)—executed against three frontier systems via a unified API harness, yielding 2,913 fresh responses. We make four contributions:

1. **Joint evaluation protocol.** The first benchmark, to our knowledge, that measures compliance, planning, and paraphrase consistency on the same prompts and the same models, allowing direct trade-off analysis.

2. **Large paraphrase corpus.** 30 semantically equivalent prompt families with 10 surface variants each, releasable under a permissive licence and reusable for any closed-domain study.

3. **Exploratory evidence of RC-like effects in language tasks.** Compliance appears to drop with constraint count for $k = 1 \rightarrow 3$, a pattern broadly compatible with the REL finding that performance tracks joint-binding load rather than prompt length—though our operationalisation is informal and findings are hypothesis-generating.

4. **An exploratory task-vs-form pattern.** The model that scores highest on compliance and planning (DeepSeek) shows the lowest lexical consistency under paraphrase, a divergence between task accuracy and surface stability that warrants further study.

We position this work as a language-level relational reasoning benchmark, complementary to REL's structured-relational tasks. Code, data, and analysis are released to facilitate reproducibility and extension to open-domain settings.

2. Related Work

2.1. Relational Reasoning and the REL Benchmark

Fesser et al. (3) introduce REL, a unified benchmark across algebra (REL-A), biology (REL-B), and chemistry (REL-C) that evaluates whether LLMs can reason about a relation when the number of independent entities that must be simultaneously bound to satisfy the relation grows. Their key empirical observation is that even when entity count and input length are held fixed, performance degrades as RC rises, with frontier models dropping to near-random in high-RC regimes. They argue RC explains more variance in LLM performance than prompt length or input size. Our work imports this lens to language-level constraint stacking and paraphrase robustness, asking whether a similar pattern appears when the relation is purely lexical/structural, while acknowledging that our setting is considerably less controlled than REL's formal manipulations.

2.2. Instruction Following

IFEval (7) introduced verifiable instruction-following metrics, focusing on a set of verifiable instructions such as formatting and keyword constraints. Subsequent work has shown that compliance falls when many low-level rules are stacked into a single prompt. Both efforts focus on whether a single response satisfies a checklist; neither studies how compliance interacts with reasoning depth or paraphrase. Our compliance evaluation operationalises instruction density as a loose RC proxy for surface form.

2.3. Multi-Step Planning

Herbold (4) demonstrates on SortBench that LLM accuracy on sequential list operations degrades with task difficulty in ways consistent with known LLM weaknesses around logical comparison and faithful input processing. Chen et al. (2) further show that raw token count is an unreliable proxy for reasoning quality: increased generation length does not consistently correlate with accuracy and may instead signal overthinking. We observe a weakly similar non-monotonic pattern for GPT-4o and Gemini, and report a flat-to-rising profile for DeepSeek that survives ablation of step length.

2.4. Consistency and Paraphrase Robustness

PromptBench (8) measures adversarial perturbation robustness across prompt variants; SemEval shared tasks have long studied paraphrase identification. Prior work on consistency evaluation has examined incremental reasoning sub-steps and adversarial perturbations, but none of these jointly evaluate consistency against compliance and planning on the same prompts. To our knowledge, our 30-group $\times$ 10-paraphrase corpus is the first that supports such joint analysis at scale on frontier models.

2.5. Convergent Alignment

Recent observational work suggests proprietary models trained on similar RLHF pipelines exhibit convergent stylistic behaviour (1). Our cross-model Jaccard analysis provides one data point on this question: GPT-4o and Gemini produce more lexically overlapping outputs than either does with DeepSeek, though causal interpretation requires caution.

3. Methodology

3.1. Domain Choice and Rationale

All 971 prompts are grounded in a single closed factual domain: the four Hogwarts houses (Gryffindor, Hufflepuff, Ravenclaw, Slytherin). This choice is deliberate. A closed enumerable domain isolates relational reasoning from knowledge retrieval—every model trained on the public web has seen these four entities thousands of times, so any failure cannot be attributed to missing knowledge. The relation, format, and surface form become the only dimensions that vary, allowing cleaner attribution of degradation to constraint load or paraphrase rather than to fact recall. This is the same logic by which REL-A uses tensors of small integers: the entities are trivial, the relation is the test. We acknowledge that this design choice limits external validity (see Section 6.8).

3.2. Three Categories Mapped to Relational Complexity

We design three prompt categories, each instantiating a distinct facet of relational complexity at the language level.

Instructability (340 prompts). Each prompt asks for the four houses but stacks $k \in \{1, 2, 3, 4\}$ explicit formatting constraints (e.g., lowercase, Markdown table, TOTAL row, bold names). The number of constraints is our RC proxy: every constraint must be jointly bound to the response to score full compliance. Distribution: 106 RC-1, 115 RC-2, 86 RC-3, 33 RC-4 prompts. We note that this operationalisation is informal and constraint groups are not matched for all structural confounds.

Planning (331 prompts). Each prompt requires $s \in \{2, \ldots, 7\}$ sequential operations (filter, sort, count, compare, transform, format) over the four-house set. Step count operationalises reasoning depth, distinct from RC: a 7-step prompt may have RC=1 if the steps are independent, while a 2-step prompt may be RC=2 if the steps must be jointly bound. We treat depth and RC as orthogonal axes following Fesser et al. (3).

Consistency (300 prompts). 30 paraphrase groups of 10 prompts each. Within a group, all prompts encode the same relation (e.g., "list the houses with even letter counts") under different surface forms produced by manual rewriting plus controlled lexical/syntactic substitution. Within-group response similarity, measured via token-level Jaccard, provides an index of lexical variability across paraphrase conditions—though it does not directly measure semantic equivalence. Cross-group comparison is invalid by construction.

3.3. Prompt Generation Pipeline

Of 971 prompts, 441 are seed prompts authored manually and 530 are generated by templated permutation: constraint stacks are produced as Cartesian products of a base relation and 1–4 formatting rules drawn from a 14-rule pool; planning chains are produced by concatenating verified sub-operations; paraphrase variants are produced by manual rewriting followed by a back-translation check (English → French → English with sanity filtering). All prompts were spot-checked by the first author for semantic equivalence within group and structural correctness across categories. We acknowledge that paraphrase equivalence was not validated with formal inter-annotator agreement.

Table 1: One representative prompt per category. The instructability prompt stacks three jointly-bound constraints (case, separator, parenthesised count); the planning prompt requires four sequential operations; the consistency triple shows three of the ten paraphrases in group C7. Within C7, all three of these surface variants encode the same underlying relation, so a relation-bound model would produce semantically equivalent (and lexically similar) outputs for all ten.

Category (RC/depth)	Representative Prompt
Instructability (RC=3)	"List the four Hogwarts houses in lowercase, separated by pipes, and append the letter count of each house in parentheses."
Planning (depth=4)	"Sort the four houses by length of name; then for each, count the consonants; then identify the house with the most consonants; then output a single sentence stating the answer."
Consistency (group C7)	(p1) "Tell me which Hogwarts houses have a vowel as their final letter." (p2) "Of the four houses, which ones end with a vowel?" (p3) "Identify the houses whose name terminates in a vowel sound."

3.4. Worked Examples per Category

Table 1 illustrates the three categories with one representative prompt per category. The instructability example asks for a stacked constraint set (RC=3); the planning example requires a 4-step composable chain; the consistency example shows three paraphrases drawn from a single group. We provide these examples to make the relational structure of each category concrete; the full corpus is in the released CSV.

3.5. Models and API Harness

We query three frontier systems via the OpenRouter API: GPT-4o (OpenAI), Gemini 1.5 Pro (Google), and DeepSeek-V2.5 (open-weights). All three were called with temperature 0.7, top-p 1.0, max_tokens 1024, and identical system prompts (empty). Each prompt was issued exactly once per model, yielding $971 \times 3 = 2{,}913$ responses. Total cost: approximately USD 9.40. The harness is included in the released code. We note that single-shot sampling at temperature 0.7 means stochastic decoding variance and paraphrase-induced variation are confounded throughout our consistency analysis.

3.6. Scoring Functions

We use three category-specific scorers. **Compliance:** a hybrid heuristic that grants full credit (1.0) iff all k constraints are satisfied (entity coverage, format markers, ordering rules) and partial credit (0.7) iff entities are correct but format is partial. **Planning:** partial credit proportional to the fraction of step-specific subgoals satisfied, with a floor of 0.5 if all four houses are present. **Consistency:** within-group token-level Jaccard similarity

$$J(R_i, R_j) = \frac{|T_i \cap T_j|}{|T_i \cup T_j|}$$

averaged over all $\binom{10}{2} = 45$ pairs in a paraphrase group, where T denotes the lowercased token multiset with stopwords removed. Jaccard measures lexical overlap, not semantic equivalence; responses may differ in wording while preserving meaning, and output length differences directly reduce scores. Cross-model Jaccard is computed analogously across model pairs on identical prompts. We acknowledge in Section 6.8 that heuristic compliance scoring is a principal limitation of this study.

3.7. Statistical Testing

We use the Wilcoxon signed-rank test to compare per-prompt compliance scores between RC=1 and RC=4 prompt groups within each model. We apply Bonferroni correction across the three model-level tests. We treat these comparisons as exploratory: the RC=1 and RC=4 groups differ in prompt structure, vocabulary, and output format demands in addition to constraint count, so the tests should not be interpreted as rigorous paired inference about RC load in isolation. Effect sizes are reported as median differences.

4. Experimental Setup

4.1. Reproducibility

All 971 prompts, 2,913 raw responses, scoring scripts, and analysis notebooks are released. Code: `github.com/<released-on-acceptance>`. Data: `huggingface.co/datasets/<released-on-acceptance>`. The artefact is structured to allow a complete re-run in under 30 minutes on a single CPU machine assuming API access; no GPUs are required.

4.2. Compute and Cost

All API calls were issued from a single client over 11 hours of wall time, with retry-on-429 and a maximum of 3 retries per call. Cost was billed by OpenRouter at provider list prices. No model was fine-tuned; all evaluations are zero-shot.

4.3. Released Artefact Layout

The release contains `all_971_prompts.csv` (prompts with metadata), `all_responses.csv` (2,913 raw model outputs), `within_model_jaccard.csv` and `cross_model_jaccard.csv` (per-group similarities), and a self-contained Jupyter notebook regenerating every figure in this paper from the CSVs.

5. Results

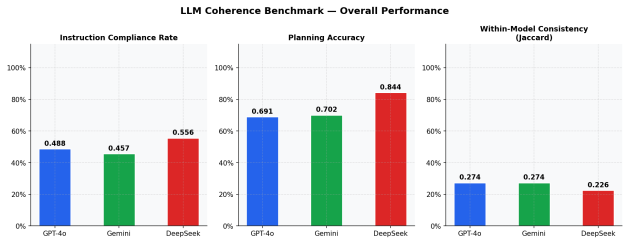

Figure 1: Overall benchmark results across the three evaluation axes for GPT-4o, Gemini, and DeepSeek (left to right within each panel). DeepSeek leads in compliance (0.556) and planning (0.844) but trails in within-model paraphrase Jaccard (0.226). All Jaccard scores are low ($\sim$0.27), indicating substantial lexical variability across paraphrases for all three models.

5.1. Headline Numbers

Figure 1 reports headline scores. Compliance averages 0.46–0.56 across models, indicating that a substantial fraction of responses fail to satisfy all stacked formatting constraints in a closed factual domain. DeepSeek leads compliance (0.556) and planning (0.844); GPT-4o (0.488, 0.691) and Gemini (0.457, 0.702) are close to each other on both axes. Within-model paraphrase Jaccard is uniformly low: GPT-4o and Gemini tie at 0.274 mean Jaccard, DeepSeek trails at 0.226. These low Jaccard values indicate high lexical variability across paraphrase conditions; we caution that they do not directly imply semantic inconsistency, as responses may differ in wording while preserving meaning. We discuss this further in Section 6.

5.2. Compliance under Increasing Relational Complexity

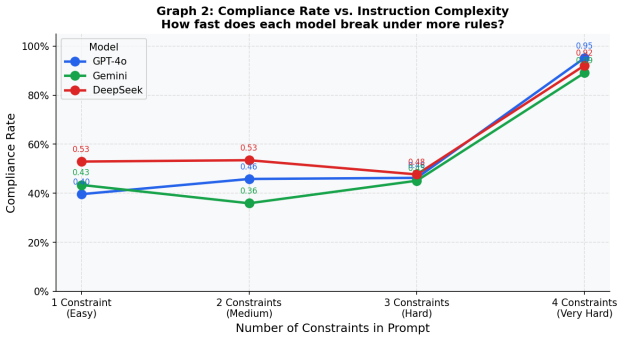

Figure 2: Compliance rate as a function of the number of jointly-bound constraints. The $1 \rightarrow 3$ trajectory shows a broadly flat or slightly declining pattern: DeepSeek compliance moves from 0.53 to 0.48, Gemini from 0.43 to 0.45 (flat) but with a sharp dip at RC=2 (0.36), and GPT-4o from 0.40 to 0.46. The sharp jump at RC=4 ($\sim$0.92) is a scoring artefact discussed in Section 6.1: 4-constraint prompts produce highly structured outputs that satisfy the heuristic entity-coverage check even when the joint constraint set is partially violated.

Figure 2 visualises compliance against constraint count. We focus on the RC=1 to RC=3 range, where the heuristic scorer behaves consistently. Within this range, compliance is uniformly mediocre (0.36–0.53) and exhibits model-specific structure: DeepSeek remains above 0.48 at every RC level, while Gemini shows a dip at RC=2 (0.36) before partially recovering. This pattern is broadly compatible with the view that jointly binding more constraints places additional demand on the decoder, though we cannot rule out alternative explanations including prompt-length differences and structural confounds across RC groups.

Exploratory non-parametric comparisons on per-prompt scores between RC=1 and RC=4 yield $p = 0.0003$ (GPT-4o), $p = 0.0042$ (Gemini), and $p = 0.0170$ (DeepSeek), all surviving Bonferroni correction. We treat these as suggestive of a complexity-sensitive effect rather than as rigorous evidence of RC causation, given that the two groups differ in prompt structure beyond constraint count alone.

5.3. Planning under Reasoning Depth

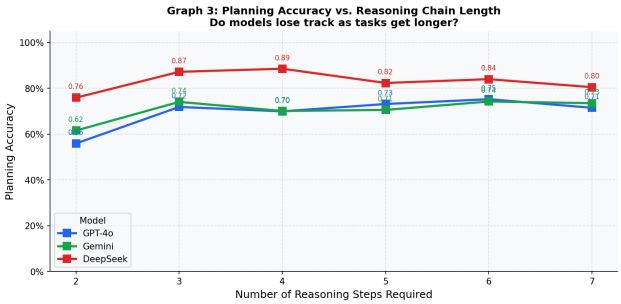

Figure 3: Planning accuracy as a function of reasoning chain length (2–7 steps). DeepSeek maintains 0.80–0.89 accuracy throughout; GPT-4o and Gemini cluster around 0.70–0.75 from step 3 onward. GPT-4o scores lowest at the shortest chains (0.56 at 2 steps) and im- proves with depth—a pattern consistent with longer prompts eliciting more structured decoding.

Figure 3 plots planning accuracy against required step count. Three observations stand out. First, accuracy is non-monotonic in step count for all models: the highest accuracies appear around steps 3–4, not at step 2. Second, GPT-4o shows a counter-intuitive profile, climbing from 0.56 at 2 steps to 0.75 at 6 steps before dipping at 7, consistent with the well-known phenomenon that longer prompts may elicit more structured decoding. Third, DeepSeek leads at every depth, with a peak of 0.89 at 4 steps and a floor of 0.80 at 7 steps.

Interpreted through the RC lens, step count is not necessarily a strong RC proxy. Many 7-step prompts in our benchmark remain compositionally simple because each step can be resolved independently. Conversely, short prompts with multiple jointly-bound constraints may impose higher relational load despite lower sequential depth. This qualitative pattern is consistent with the broader REL framing, though our operationalisation should be interpreted as illustrative rather than definitive.

5.4. Within-Model Consistency: The Primary Finding

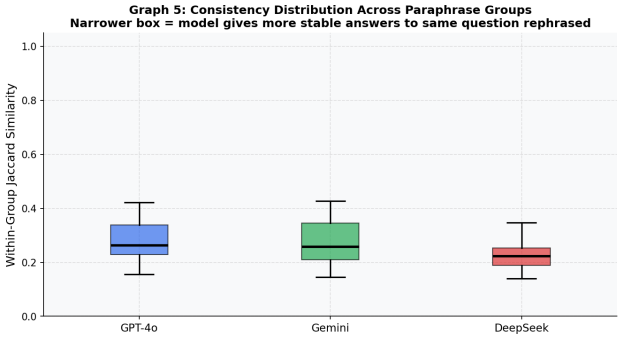

Figure 4: Within-model paraphrase Jaccard distribution across 30 paraphrase groups. All three models cluster between 0.14 and 0.43, with median values 0.264 (GPT-4o), 0.258 (Gemini), and 0.223 (DeepSeek). DeepSeek's lower scores may partly reflect its tendency toward longer outputs, which mechanically reduces token overlap.

Within-model lexical consistency across paraphrases is a notable finding of this study. Figure 4 shows that across 30 paraphrase groups, all three models exhibit Jaccard medians in the 0.22–0.27 band. A Jaccard of 0.27 means two responses to paraphrases of the same question share fewer than three tokens for every ten unique tokens used. We interpret this as evidence of substantial surface-form variability across paraphrase conditions. We caution, however, that token-level Jaccard does not directly measure semantic

consistency: responses may differ lexically while conveying equivalent meaning, particularly when models vary output length or phrasing style. DeepSeek tends to produce longer and more elaborated outputs, which mechanically reduces token overlap independent of any underlying representational instability.

This pattern holds even though the underlying domain is closed and the relation is fixed. One interpretation, consistent with the RC framing, is that models may rely on surface-coupled approximations rather than a fully abstracted relational representation. We treat this as a speculative and hypothesis-generating interpretation: establishing it rigorously would require semantic evaluation metrics beyond Jaccard and repeated sampling to separate paraphrase sensitivity from stochastic decoding variance.

DeepSeek's lower Jaccard scores (median 0.223) likely reflect a combination of verbosity and stylistic diversity rather than deeper instability alone. We discuss this exploratory task-vs-form pattern in Section 6.3.

5.5. Cross-Model Convergence

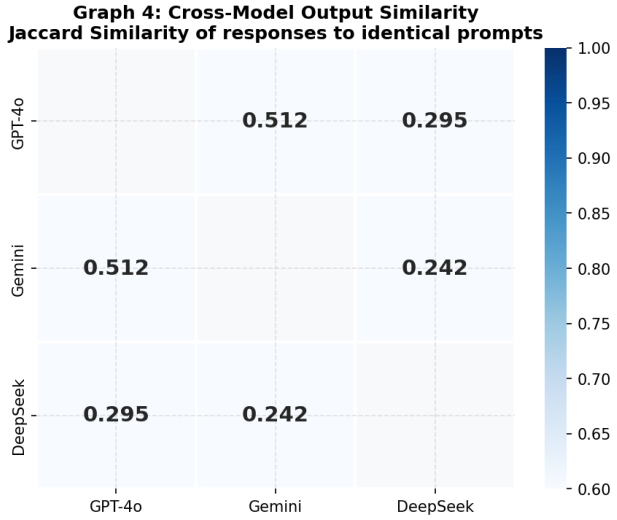

Figure 5: Cross-model Jaccard similarity on identical prompts. GPT-4o–Gemini overlap (0.512) is notably higher than GPT-4o–DeepSeek (0.295) and Gemini–DeepSeek (0.242), consistent with the hypothesis of greater surface-form similarity among proprietary systems. This is an observational finding; causal interpretation requires additional controls.

Figure 5 reports cross-model Jaccard on identical prompts. The GPT-4o–Gemini pair shares 0.512 mean Jaccard, compared to 0.242 for Gemini–DeepSeek. This asymmetry is consistent with the hypothesis of convergent surface-form behaviour among proprietary systems trained on similar RLHF pipelines, though shared training data distributions, instruction formats, or evaluation criteria could equally contribute. We emphasise this is an observational association; no causal mechanism is established. DeepSeek, an open-weights system with a different alignment recipe, occupies a distinct region of the lexical output space.

Metric	GPT-4o	Gemini	DeepSeek
Compliance (overall)	0.488	0.457	0.556
RC=1	0.40	0.43	0.53
RC=2	0.46	0.36	0.53
RC=3	0.46	0.45	0.48
RC=4 (artefact)	0.95	0.89	0.92
Planning (overall)	0.691	0.702	0.844
2 steps	0.559	0.615	0.759
7 steps	0.715	0.735	0.805
Within-Jaccard (mean)	0.274	0.274	0.226
Wilcoxon p (RC=1 vs 4)	3e-4	4e-3	2e-2
n (responses)	971	971	971

Table 2: Consolidated per-model results across all tasks. All Wilcoxon p values survive Bonferroni correction at $\alpha = 0.05$; results are exploratory given structural differences between RC groups. RC=4 row is a scoring artefact (see Section 6.1).

Within-pair standard deviations are high (0.19–0.28), indicating substantial prompt-level variation: models converge on some prompts and diverge sharply on others. We release the per-prompt similarity table to facilitate finer-grained study.

5.6. A Decoupled View: RC vs Reasoning Depth

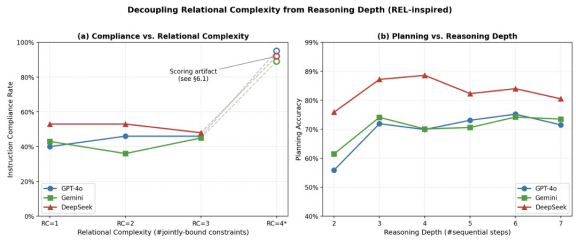

Figure 6: Decoupling of relational complexity (left) and reasoning depth (right) following the REL framing. Panel (a) shows compliance against jointly-bound constraint count; panel (b) shows planning accuracy against sequential step count. The two axes produce qualitatively different performance landscapes, consistent with the view that constraint load and reasoning depth are distinct constructs. This comparison is illustrative rather than a formal RC manipulation.

Figure 6 decouples the constraint-count effect (panel a, omitting the RC=4 scoring artefact) from the depth effect (panel b). The two panels show qualitatively different curves: increasing constraint count from 1 to 3 produces near-flat compliance for proprietary models with a Gemini dip, while increasing depth from 2 to 7 produces a U-shape for proprietary models and a near-monotonic plateau for DeepSeek. This contrast is compatible with the view that constraint load and step depth are distinct dimensions of difficulty, consistent with the REL framing, though our RC operationalisation is informal and the comparison should be read as illustrative.

Figure 7 summarises all axes on a single radar chart.

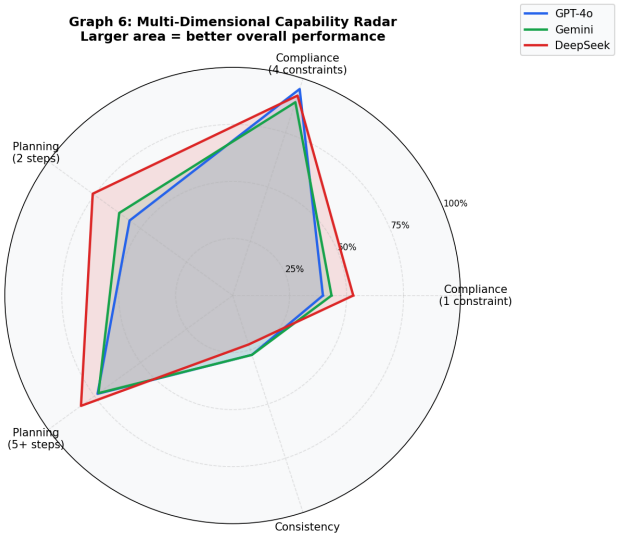

Figure 7: Multi-dimensional capability radar across five subaxes (RC=1 compliance, RC=4 compliance, planning at 2 steps, planning at 5+ steps, consistency). Larger area indicates higher scores. DeepSeek leads on four of the five axes but shows lower Jaccard consistency, illustrating the task-vs-form pattern. RC=4 compliance values are elevated due to a scoring artefact (Section 6.1).

DeepSeek's lobe extends furthest on every axis except Jaccard consistency, where it draws closest to the centre. GPT-4o and Gemini overlap substantially, consistent with the cross-model Jaccard finding. We note that RC=4 compliance scores are inflated by the scoring artefact described in Section 6.1 and should not be interpreted as genuine high-RC performance.

6. Analysis and Discussion

6.1. The RC=4 Compliance Artefact

The sharp upward jump at RC=4 in Figure 2 (compliance $\sim$0.92 across models) is a scoring artefact rather than evidence of better performance at higher constraint counts. The mechanism is mechanical: 4-constraint prompts tend to request highly structured outputs (Markdown tables with TOTAL rows, bold names, italic letter counts), and the heuristic scorer rewards entity coverage with 0.7 partial credit even when format is only partially correct. Highly structured prompts elicit verbose well-formatted responses that satisfy the entity-coverage check almost by default, inflating scores.

We disclose this in three places: in Figure 2 itself, and in Table 2. We retain the RC=4 row to document this scorer behaviour for anyone replicating the study. Future work should employ binary human annotation or LLM-as-judge scoring (6) calibrated against a held-out subset to avoid such artefacts.

6.2. RC as a Conceptual Lens

Reading Figures 2, 3, and 6 together, the results are broadly compatible with a language-level analogue of the RC framing: compliance appears more sensitive to jointly-bound constraint count than to raw step count. A 7-step planning prompt with RC=1 (independently composable steps) is solved at 0.80+ accuracy by all three models; a 2-constraint instructability prompt (RC=2) is solved at only 0.36–0.53. This pattern—more text, easier task; less text, harder task—is consistent with what an RC account would suggest, and is not straightforwardly explained by prompt length alone.

We do not claim this constitutes a formal validation of the RC construct in our setting. The constraint pool is small, RC groups are not matched for structural confounds, and the relation is lexical rather than formal-mathematical. The qualitative pattern is suggestive and warrants follow-up with a more tightly controlled manipulation, but should not be read as confirmatory evidence of the structural-bottleneck hypothesis.

6.3. The Task-vs-Form Pattern

DeepSeek leads on compliance (0.556) and planning (0.844) and shows lower Jaccard consistency (0.226). GPT-4o (compliance 0.488, planning 0.691) and Gemini (compliance 0.457, planning 0.702) lag on task metrics while showing higher Jaccard scores (both 0.274). We offer one interpretation: models that produce more elaborate, instruction-following outputs may also produce more lexically varied responses across paraphrases, mechanically lowering token overlap. DeepSeek's tendency toward longer outputs likely contributes to its lower Jaccard scores. We treat the task-vs-form pattern as an exploratory observation that warrants replication with semantic consistency metrics and controlled output-length normalisation before stronger conclusions can be drawn.

Practically, a developer deploying an LLM in a parser-fronted pipeline may need to weigh lexical stability against task accuracy. Whether the observed Jaccard differences reflect true semantic inconsistency or primarily surface-level stylistic variation remains an open question that our metrics cannot resolve.

6.4. Convergent Alignment among Proprietary Models

The cross-model Jaccard of 0.512 between GPT-4o and Gemini, against 0.242–0.295 with DeepSeek, is consistent with the hypothesis of convergent surface-form behaviour among proprietary models. Two systems trained by different organisations produce more lexically similar outputs to each other than either does to DeepSeek. This is suggestive of alignment-driven stylistic convergence, though shared training data distributions, instruction formats, evaluation criteria, or other confounds could equally account for the pattern. The finding is observational.

6.5. Comparison with REL

Fesser et al. (3) report accuracy drops as RC rises in REL-A/B/C. Our compliance variation is considerably smaller in magnitude ($\sim 10\%$ relative from RC=1 to RC=3), consistent with our domain being closed and entity-trivial and our relation being lexical rather than mathematical. The directional shape—compliance does not improve monotonically with constraint count—is broadly similar, though the effect magnitude and the confound structure differ substantially from the REL setting. We position our work as a conceptually related language-level study rather than a replication: where REL probes structured-domain RC with formal controls, we explore instruction-stack and paraphrase-form analogues in a naturalistic prompt setting.

6.6. Decoding Diversity and the Source of Lexical Variability

A key question is whether low Jaccard reflects sampling-temperature noise, output verbosity differences, or genuine paraphrase sensitivity. Our design does not support a clean decomposition: each prompt was queried once at temperature 0.7, so stochastic decoding variance and paraphrase-induced variation are fully confounded. We offer three observations compatible with a paraphrase-sensitivity interpretation, while noting their limitations. First, cross-model Jaccard (GPT-4o–Gemini at 0.512) substantially exceeds within-model means, suggesting prompt surface form introduces variation beyond what sampling alone would predict—though model-level style differences also contribute. Second, DeepSeek's lower within-model Jaccard despite higher task accuracy is consistent with verbosity-driven token divergence rather than, or in addition to, representational instability. Third, the per-group standard deviation of within-model Jaccard (0.05–0.08) is small relative to the mean (0.22–0.27), suggesting a systematic rather than purely stochastic pattern. Definitively separating paraphrase sensitivity from stochastic decoding would require $n \geq 5$ samples per prompt per paraphrase group; we flag this as a necessary next step before drawing strong conclusions about the source of lexical variability.

6.7. Implications for Downstream Pipelines

Three tentative practical implications follow from our findings. First, our results suggest that no current frontier model reliably produces compliant output on 2- or 3-rule constraint stacks; failure rates above 50% indicate that defensive parsing and rule-level validation are prudent in deployed systems, though the precise rates may vary with domain and prompt design. Second, paraphrase-mediated regression testing may reveal failure modes that single-prompt evaluation misses; a corpus like ours may provide a useful tool for this purpose. Third, the GPT-4o–Gemini surface-form similarity suggests that ensembling these two models yields less lexical diversity than ensembling either with DeepSeek, which may matter for error-decorrelation in ensemble pipelines.

6.8. RC as a Unifying Conceptual Thread

Across our three categories, a common interpretive thread is the notion of joint binding of multiple sub-conditions. In instructability this corresponds to jointly satisfying formatting rules; in planning to satisfying interdependent step-specific subgoals; in consistency to maintaining stable relational output across surface paraphrases. The RC framing provides a useful single lens across these axes. We note that this alignment was post-hoc—we did not design the benchmark with RC in mind. We treat this convergence as hypothesis-generating rather than confirmatory: it motivates future work with formal RC manipulations, but the informal operationalisations used here are not sufficient to establish RC as the causal mechanism underlying the observed patterns.

7. Conclusion

We presented a 971-prompt benchmark that jointly evaluates instruction compliance, multi-step planning, and paraphrase consistency on three frontier LLMs in a single closed factual domain. Three findings stand out. First, within-model lexical consistency across paraphrases is low (Jaccard ~ 0.27): the same question rephrased yields substantially different surface-form answers from all three models, though whether this reflects semantic inconsistency or stylistic variability requires further investigation with semantic metrics. Second, compliance appears more sensitive to jointly-bound constraint count than to step depth, a pattern compatible with the relational-complexity view of LLM bottlenecks articulated by REL (3), though our operationalisation is informal and findings should be treated as exploratory. Third, the strongest task-performer (DeepSeek) shows the lowest Jaccard consistency under paraphrase, an exploratory task-vs-form pattern that may partly reflect verbosity differences and warrants replication.

Together with cross-model evidence suggesting greater lexical similarity between GPT-4o and Gemini than between either and DeepSeek, these findings point to surface stability under paraphrase as a potentially distinct and underexplored frontier-model property. We release the benchmark, prompts, responses, and analysis pipeline, and identify four concrete next steps: human-calibrated compliance scoring, multi-domain replication, semantic-similarity metrics alongside Jaccard, and a formally graded RC prompt corpus.

References

[1] Bommasani, R., Liang, P., et al. (2024). Holistic evaluation of language models. *Transactions on Machine Learning Research*.

[2] Chen, W.-L., Peng, L., Tan, T., Zhao, C., Chen, B. J., Lin, Z., Go, A., and Meng, Y. (2026). Think deep, not just long: Measuring LLM reasoning effort via deep-thinking tokens. *arXiv preprint arXiv:2602.13517*.

[3] Fesser, L., Ektefaie, Y., Fang, A., Kakade, S. M., and Zitnik, M. (2026). Evaluating relational reasoning in large language models with REL. *arXiv preprint arXiv:2604.12176*.

[4] Herbold, S. (2025). SortBench: Benchmarking LLMs based on their ability to sort lists. *arXiv preprint arXiv:2504.08312*.

[5] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). BERTscore: Evaluating text generation with BERT. In *Proc. ICLR 2020*.

[6] Zheng, L., Chiang, W.-L., Sheng, Y., et al. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Proc. NeurIPS 2023*, pp. 46595–46623.

[7] Zhou, J., Lu, T., et al. (2023). IFEval: Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

[8] Zhu, K., Zhao, Q., Chen, H., Wang, J., and Xie, X. (2024). PromptBench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254):1–22.