

# Did You Forget What I Asked?

## Prospective Memory Failures in Large Language Models

Avni Mittal

Microsoft

avnimittal@microsoft.com

### Abstract

Large language models often fail to satisfy formatting instructions when they must simultaneously perform demanding tasks. We study this behavior through a *prospective memory*-inspired lens from cognitive psychology, using a controlled paradigm that combines verifiable formatting constraints with benchmark tasks of increasing complexity. Across three model families and over 8,000 prompts, compliance drops by 2–21% under concurrent task load. Vulnerability is highly type-dependent: terminal constraints (requiring action at the response boundary) degrade most, with drops up to 50%, while avoidance constraints remain comparatively robust. A salience-enhanced format (explicit instruction framing plus a trailing reminder) recovers much of the lost compliance, restoring performance to 90–100% in many settings. Interference is bidirectional: formatting constraints can also reduce task accuracy, with one model’s GSM8K accuracy dropping from 93% to 27%. In additional stacking experiments, joint compliance declines sharply as constraints accumulate. All results use deterministic programmatic checkers, with no LLM-as-judge component, on publicly available datasets.

## 1 Introduction

Large language models are increasingly used in settings where users provide behavioral or formatting constraints alongside substantive requests. A typical prompt might ask for an explanation while also requiring a specific output form (e.g., all capitals, a required ending phrase, or a fixed number of bullets). Users expect the model to preserve such constraints throughout generation. In practice, models often produce strong content while violating the requested format.

We study this pattern as a *functional analog* of prospective memory (PM): remembering to execute a deferred intention at the appropriate future point (Einstein et al., 2005). In humans, PM failures

increase under cognitively demanding intervening tasks; this is often discussed as a cognitive-load effect. Our paper asks whether an analogous behavioral pattern appears in LLMs when instruction maintenance and task solving must be carried out together.

To test this, we compose verifiable IFEval-style constraints with benchmark tasks spanning low to high operational load (TriviaQA, MMLU, GSM8K, CNN/DailyMail). We evaluate both instruction compliance and task correctness with deterministic code-based checkers, avoiding LLM-as-judge components. Importantly, our load ordering is operational rather than a definitive cognitive scale, so we interpret load effects behaviorally.

Across three models from distinct architecture families and over 8,000 evaluated prompts, we find:

1. **Systematic forgetting under concurrent load.** Compliance drops 2–21% when models must solve an additional task, with vulnerability varying sharply by instruction type (terminal constraints are most affected; avoidance constraints are nearly immune).
2. **A simple mitigation via salience.** A salience-enhanced prompt format recovers most lost compliance, restoring performance to 90–100% in many settings.
3. **Dual-task tradeoffs and scaling fragility.** Interference is bidirectional (format constraints can reduce task accuracy, e.g., GSM8K 93% to 27%), and joint compliance declines as multiple constraints are stacked.

## 2 Related Work

**Instruction Following, Salience, and Robustness.** IFEval (Zhou et al., 2023) established a practical, checker-based protocol for verifiable instruction compliance. More recent benchmarks broaden

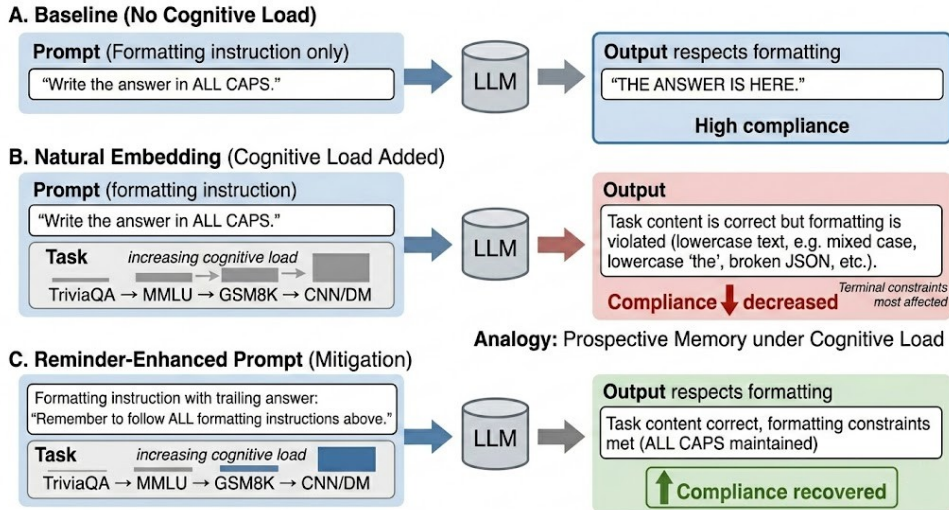


Figure 1: Overview of the experimental pipeline. A verifiable formatting constraint from IFEval is composed with a benchmark task of varying difficulty (TriviaQA, MMLU, GSM8K, or CNN/DailyMail) using either the natural embedding template or the salience-enhanced reminder template. The model’s response is then evaluated along two independent axes: (1) deterministic IFEval compliance checking (strict and loose) and (2) task-specific accuracy verification. This dual-evaluation design enables simultaneous measurement of prospective memory failure (compliance drop) and dual-task interference (accuracy drop).

this picture: FollowBench shows that performance degrades as constraints are incrementally added (Jiang et al., 2024), while AGENTIF evaluates long, highly constrained agentic prompts (Qi et al., 2025). In parallel, work on instruction prioritization and robustness shows that failures can arise from conflict resolution or adversarial pressure rather than pure forgetting (Wallace et al., 2024; Zhang et al., 2025; Xiang et al., 2024). We focus on a complementary failure mode: non-adversarial forgetting under concurrent task load.

**Prospective Memory and Dual-Task Interference.** Our framing follows prospective-memory theory. Beyond the multiprocess account (Einstein et al., 2005; McDaniel and Einstein, 2000), PAM-style evidence shows measurable ongoing-task costs when deferred intentions must be maintained (Smith, 2003). Additional studies show that interference scales with intention characteristics and task complexity (Marsh et al., 2003). This aligns with our finding that instruction type and concurrent-load level modulate both compliance and task accuracy. Methodologically, this perspective is consistent with the broader line of treating LLMs as subjects in cognitive experiments (Binz and Schulz, 2023; Jones and Steinhardt, 2022).

**Long-Context and Retrieval Failures.** Prior long-context work, including Lost in the Middle

(Liu et al., 2024), documents strong position sensitivity in retrieval from long prompts. RULER further shows that long-context performance can degrade sharply on harder compositional retrieval tasks (Hsieh et al., 2024). Separately, GSM-IC demonstrates that irrelevant context can substantially harm reasoning accuracy (Shi et al., 2023). Our setting differs: the instruction is salient and fixed in position, but compliance still drops as concurrent demands rise, pointing to generation-time intention maintenance rather than pure retrieval failure.

**Prompt Salience and Evaluation Reliability.** Prompt ordering and template choices materially affect model behavior, supporting reminder-based interventions (Chen et al., 2024; Guan et al., 2025; Lyu et al., 2024). At evaluation time, LLMBAR highlights weaknesses in judge-based assessment (Zeng et al., 2024), reinforcing the value of deterministic, code-based checkers used in our experiments.

### 3 Methodology

Figure 1 provides an overview of our experimental pipeline.

#### 3.1 Problem Formulation

Given a prompt containing a formatting instruction  $I$  and a task  $T$ , the model produces response  $R$ . We measure *compliance rate*  $CR$ , defined as the frac-

tion of responses passing the deterministic IFEval checker, and define the *forgetting delta*:

$$\Delta = CR(I_{\text{alone}}) - CR(I + T) \quad (1)$$

Positive  $\Delta$  indicates the additional task caused the model to forget the instruction.

### 3.2 Composition Framework

**Instructions (IFEval).** We use 15 of IFEval’s 25 instruction types, selected for compatibility with task composition (the full list is given in Table 5, Appendix A). These span seven categories: case constraints, keyword requirements, terminal actions, structural formatting, avoidance rules, counting constraints, and length limits.

**Distraction tasks.** We pair each IFEval constraint with a benchmark task at one of four cognitive load levels: TriviaQA (factual recall, low load), MMLU (multiple-choice reasoning, medium), GSM8K (multi-step arithmetic, high), and CNN/DailyMail (long-context summarization).

### 3.3 Primary Experiment: Natural Embedding

Each IFEval prompt already contains both a task and a constraint (e.g., “Write an essay about Java. Use all capital letters.”). We append additional benchmark tasks using a neutral transition:

#### Natural Embedding Template

*{original IFEval prompt with embedded constraint}*

Additionally, please also complete the following:

*{benchmark question}*

The *baseline* is the original IFEval prompt with no added task. The only variable between the baseline and the experimental conditions is the appended workload; the instruction format and salience are identical.

### 3.4 Mitigation Experiment: Adding a Reminder

We test whether a simple prompt modification can recover compliance. We extract the constraint from IFEval metadata and present it with explicit framing and a trailing reminder sentence:

#### Reminder Template

**IMPORTANT FORMATTING INSTRUCTION:**

*{extracted constraint text}*

Now please help me with the following task:

*{benchmark question}*

**Remember to follow ALL of my formatting instructions above.**

This adds two modifications relative to the natural condition: (1) the “IMPORTANT FORMATTING INSTRUCTION:” prefix, and (2) the trailing reminder. Together, these test whether *instruction salience*, i.e., making the constraint more prominent, can mitigate forgetting.

### 3.5 Evaluation

All evaluation is fully deterministic and uses code-based checkers with no LLM-as-judge component.

**IFEval compliance.** Each of the 15 constraint types has a dedicated checker that returns binary pass/fail. We report two modes: *strict* (applied to the raw model response) and *loose* (the response passes if any of 8 normalized variants, e.g., stripping whitespace or markdown headers, satisfies the checker).

**Task accuracy.** For GSM8K, we extract the last number from the response and compare it to the gold answer. For MMLU, we extract the answer letter using a priority heuristic and compare it to the gold label. For TriviaQA, we check whether any gold alias appears as a substring of the response (case-insensitive). For CNN/DailyMail, we compute ROUGE-L F1 with Porter stemming.

### 3.6 Multi-Constraint Stacking

Our primary experiments test one formatting constraint at a time. Real-world prompts, however, often contain multiple formatting requirements. To test whether forgetting compounds under stacking, we identify 11 of the 15 IFEval types that can be safely combined (excluding types with structural conflicts; see Table 10, Appendix F) and cross four factors: number of simultaneous constraints  $N \in \{1, 2, 3, 5\}$ , cognitive load  $M \in \{0, 1, 3\}$  GSM8K problems, and trailing reminder  $R \in \{\text{off}, \text{on}\}$ . We additionally test 4 soft-tension constraint pairs to examine pair-specific interactions; the full design and prompt construction details are in Appendix F, with aggregate stacking results in Appendix G.

Condition	Natural Embedding			With Reminder		
	o4-mini	DeepSeek	Llama	o4-mini	DeepSeek	Llama
Baseline	86.1	89.3	88.9	–		
+ TriviaQA	82.2 <sub>-3.9</sub>	83.0 <sub>-6.3</sub>	87.8 <sub>-1.1</sub>	93.3	92.2	92.2
+ MMLU	82.0 <sub>-4.1</sub>	77.4 <sub>-11.8</sub>	86.2 <sub>-2.7</sub>	94.8	92.2	93.3
+ GSM8K	83.6 <sub>-2.5</sub>	78.5 <sub>-10.7</sub>	81.9 <sub>-7.0</sub>	94.8	91.5	90.0
+ 3×GSM8K	71.1 <sub>-15.0</sub>	84.4 <sub>-4.8</sub>	84.4 <sub>-4.4</sub>	95.6	100.0	100.0
+ 5×GSM8K	82.4 <sub>-3.8</sub>	68.5 <sub>-20.7</sub>	87.0 <sub>-1.9</sub>	92.6	100.0	98.1
+ CNN/DM	77.5 <sub>-8.6</sub>	78.9 <sub>-10.4</sub>	86.8 <sub>-2.1</sub>	98.4	100.0	100.0

Table 1: IFEval strict compliance (%) across all conditions. **Left:** natural embedding, where the formatting constraint is embedded in prose. Red subscripts show the forgetting delta vs. baseline. **Right:** salience-enhanced format with explicit framing and trailing reminder. The reminder recovers compliance to 90–100% across all conditions, often exceeding the no-task baseline.

A response passes *joint compliance* if it satisfies all  $N$  constraint checkers simultaneously. We report both strict and loose variants.

## 4 Experiments

### 4.1 Datasets and Models

We compose prompts from five publicly available datasets (Table 7, Appendix A): IFEval (Zhou et al., 2023) provides 90 stratified formatting constraints (6 per type  $\times$  15 types, fixed across runs); TriviaQA (Joshi et al., 2017) supplies low-load factual recall; MMLU (Hendrycks et al., 2020) supplies medium-load multiple-choice reasoning; GSM8K (Cobbe et al., 2021) supplies high-load multi-step arithmetic (tested in single, triple, and quintuple chains); and CNN/DailyMail (See et al., 2017) supplies long-context summarization. We evaluate three models spanning different architecture families: o4-mini (reasoning, Azure OpenAI), DeepSeek-V3.1 (open-weight, Azure AI), and Llama-3.3-70B-Instruct (instruction-tuned, Azure AI). All use greedy decoding and the same system prompt. Full dataset, model, and hyperparameter details are in Appendix A.

### 4.2 Experimental Conditions

Our experiments cross two factors: *distraction task* (none, TriviaQA, MMLU, GSM8K, CNN/DailyMail) and *prompt template* (natural embedding vs. reminder-enhanced). This yields the following conditions:

**Baseline (IFEval-only).** The original IFEval prompt with no added task. This measures each model’s intrinsic compliance rate and serves as the reference for computing forgetting deltas.

**Task-only (no constraint).** Benchmark tasks presented without any formatting instruction. This provides a ceiling for task accuracy, against which we measure dual-task interference.

**Natural embedding conditions.** For each of the four distraction tasks (TriviaQA, MMLU, GSM8K, CNN/DM), the benchmark question is appended to the original IFEval prompt with a neutral connector. Additionally, GSM8K is tested in chained configurations of 3 and 5 problems to intensify cognitive load, yielding six natural conditions in total.

**Reminder-enhanced conditions.** The same six distraction conditions, but using the reminder template: the constraint is extracted and highlighted with “IMPORTANT FORMATTING INSTRUCTION:” at the top, and a trailing reminder sentence is appended at the bottom.

### 4.3 Sampling, Evaluation, and Scale

We conduct 3 independent runs (seeds 42, 137, 256) with the IFEval sample held fixed and benchmark items re-sampled per run. Within each run, all three models receive identical prompts. All evaluation is fully deterministic: IFEval compliance uses 15 code-based checkers (strict and loose modes); task accuracy uses last-number extraction (GSM8K), priority letter matching (MMLU), alias substring matching (TriviaQA), and ROUGE-L F1 (CNN/DM). Mean prompt and response token counts per condition are reported in Table 9 (Appendix C). The full experiment comprises over 8,000 evaluated prompts. Detailed sampling, evaluation, and randomization procedures are in Appendix B.

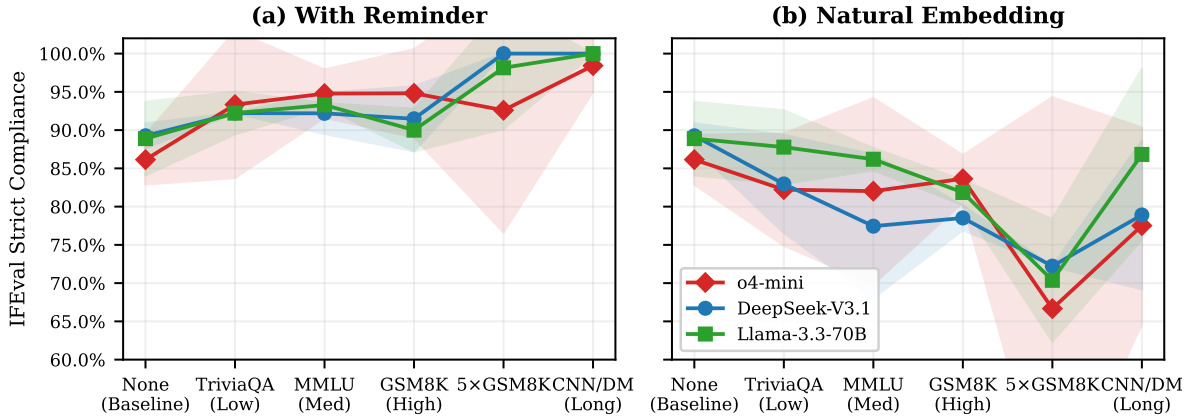


Figure 2: Main result. IFEval compliance under increasing task complexity, with shaded 95% CIs from 3 independent runs. (a) With salience-enhanced prompt: compliance stays flat at 90–100%. (b) Natural embedding: compliance drops consistently as distraction difficulty increases. Both conditions share the same baseline (no additional task).

## 5 Results and Analysis

### 5.1 Forgetting and Recovery Under Load

Table 1 presents IFEval strict compliance under both experimental conditions. The left half (natural embedding) shows that appending additional tasks causes systematic compliance degradation across all three models. The right half (salience-enhanced reminder) shows that a simple prompt modification nearly eliminates this effect. All results report strict compliance; loose evaluation shows the same qualitative pattern with slightly attenuated deltas. Red subscripts ( $_{-x}$ ) denote the forgetting delta relative to the baseline. Appendix I presents a taxonomy of five forgetting mechanisms derived from manual analysis of all task-correct failures.

**Forgetting.** We assess statistical significance using McNemar’s test with continuity correction and Cohen’s  $h$  effect sizes; full results including bootstrap 95% CIs and  $p$ -values for all 18 model–condition pairs are in Table 8 (Appendix B.1). As Table 1 shows, forgetting is broad-based: compliance drops under nearly every model–task combination. DeepSeek-V3.1 is the most affected (avg.  $\Delta = 10.8\%$ ), o4-mini shows moderate forgetting (avg.  $\Delta = 6.3\%$ ), and Llama-3.3-70B is the most robust (avg.  $\Delta = 3.2\%$ ). Figure 2(b) visualizes the compliance curves with 95% CIs, and Figure 3 shows the per-condition deltas with error bars.

**Recovery.** The right half of Table 1 shows that appending a single reminder sentence (“Remember to follow ALL of my formatting instructions above”) recovers compliance from the 77–87% range to 90–100% across all models and tasks. The recovery

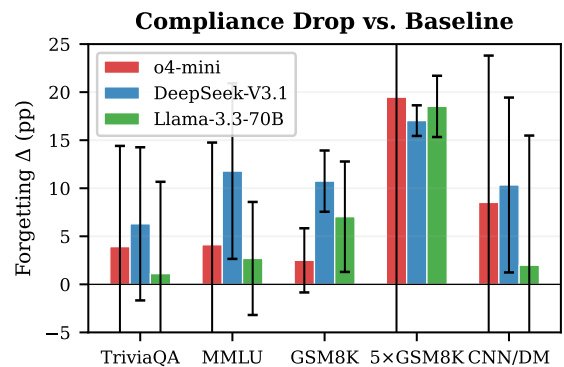


Figure 3: Forgetting deltas by distraction type and model, with 95% CI error bars from 3 runs. Positive values indicate compliance dropped vs. baseline. DeepSeek shows the largest forgetting; Llama is most robust.

holds even under heavy chaining: at  $3\times$ GSM8K, DeepSeek and Llama reach 100% (vs. 84.4% natural); at  $5\times$ GSM8K, DeepSeek maintains 100% and Llama 98.1%. The average compliance gain is 15.1% for o4-mini, 17.5% for DeepSeek, and 9.9% for Llama (Figure 4). The gain is largest for CNN/DM (+13 to +21%), where the instruction is most temporally distant from the response end. Notably, the reminder prompts *exceed* the no-task baseline in most cases, suggesting that standard instruction formats may systematically underestimate model capability.

### 5.2 Terminal Constraints Are Most Vulnerable

Table 2 decomposes forgetting by instruction type. The pattern is striking: *terminal constraints*, those requiring action at the end of the

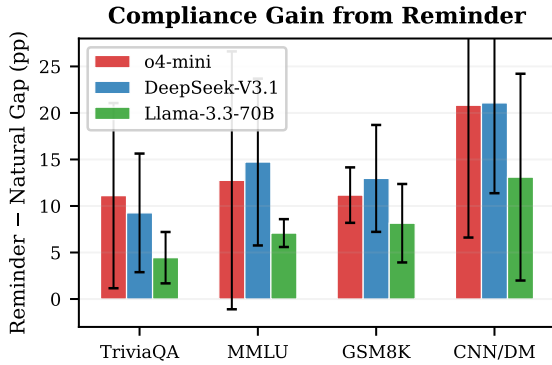


Figure 4: Compliance gain from the salience-enhanced prompt format, by distraction type, with 95% CI error bars. The effect is largest for the long-context condition (CNN/DailyMail (CNN/DM): +13 to +21%).

Instruction Type	o4	DS	Llama
<i>Most vulnerable (baseline <math>\rightarrow</math> +GSM8K, %):</i>			
end_checker	-17	-50	-17
json_format	+6	-17	-50
number_bullet_lists	-12	-44	-11
english_capital	-17	-17	-22
<i>Most robust:</i>			
no_comma	0	-11	0
forbidden_words	0	0	+6
number_words	0	0	-6

Table 2: Per-type forgetting delta (baseline minus natural + GSM8K, in %). We use the single GSM8K condition (rather than averaging across all tasks) because it represents the highest single-task cognitive load while keeping prompt structure uniform. Models shown are o4-mini, DeepSeek-V3.1, and Llama-3.3-70B-Instruct. Note: with only 6 items per type, individual deltas should be interpreted as indicative; the cross-type pattern is the robust finding.

response, suffer the largest drops. `end_checker` drops 50% for DeepSeek; `json_format` drops 50% for Llama; `number_bullet_lists` drops 44% for DeepSeek. Meanwhile, avoidance constraints (`no_comma`, `forbidden_words`) are nearly immune to forgetting, holding at or near 100%.

This mirrors predictions from human prospective memory research. Constraints requiring *continuous monitoring* (“never use a comma”) are maintained because every generated token provides an enforcement opportunity. Constraints requiring a *deferred action* (“end with this exact phrase”) are forgotten because the relevant moment is temporally distant from the instruction. By the time the model reaches the end of its response, it has generated hundreds of tokens of task content, and the deferred instruction has lost effective salience.

Condition	o4-mini	DeepSeek	Llama
<i>Task-only baseline (no formatting constraint):</i>			
Avg. across tasks	93.3	87.8	94.4
<i>With formatting constraint (reminder):</i>			
+ TriviaQA	88.9	91.5	90.7
+ MMLU	76.5	68.4	71.3
+ GSM8K $\times$ 1	85.9	84.4	83.3
+ GSM8K $\times$ 3	26.7	76.7	64.4
+ GSM8K $\times$ 5	53.7	77.8	61.1
+ CNN/DM <sup>†</sup>	18.5	20.2	19.4

Table 3: Task accuracy (%) with and without a formatting constraint. The task-only baseline averages accuracy across all single-task conditions (TriviaQA, MMLU, GSM8K $\times$ 1) without any formatting constraint. Adding IFEval instructions degrades benchmark performance, especially for o4-mini on chained GSM8K (93%  $\rightarrow$  27%). <sup>†</sup>CNN/DM reports ROUGE-L F1 ( $\times$ 100) rather than binary accuracy; no task-only baseline is available for this condition.

This vulnerability pattern suggests a clear mitigation strategy: if forgetting occurs because the constraint loses salience during generation, then *increasing salience* at the prompt boundary should recover compliance.

### 5.3 Dual-Task Interference

The forgetting effect is not one-directional. Adding formatting instructions also degrades performance on the benchmark task itself, revealing symmetric dual-task interference.

Table 3 shows task accuracy under the reminder-enhanced condition compared against the task-only baseline (no formatting constraint). MMLU accuracy drops 17–23% across all models. The most dramatic interference appears in o4-mini’s performance on chained GSM8K: accuracy on triplets drops from 93% (task-only) to just 27% when a formatting constraint is simultaneously active. Llama and DeepSeek show smaller but consistent drops on chained math (61–78% vs. the task-only ceiling).

The o4-mini result is particularly striking. As a reasoning model, o4-mini appears to allocate its reasoning budget toward format compliance at the expense of mathematical accuracy, a tradeoff not observed as severely in the standard instruction-tuned models.

**Non-monotonic scaling.** An unexpected pattern emerges in the chained GSM8K conditions (Figure 7): compliance does not decrease monotonically with chain length. For o4-mini, compliance on GSM8K $\times$ 3 (71.1%) is lower than on

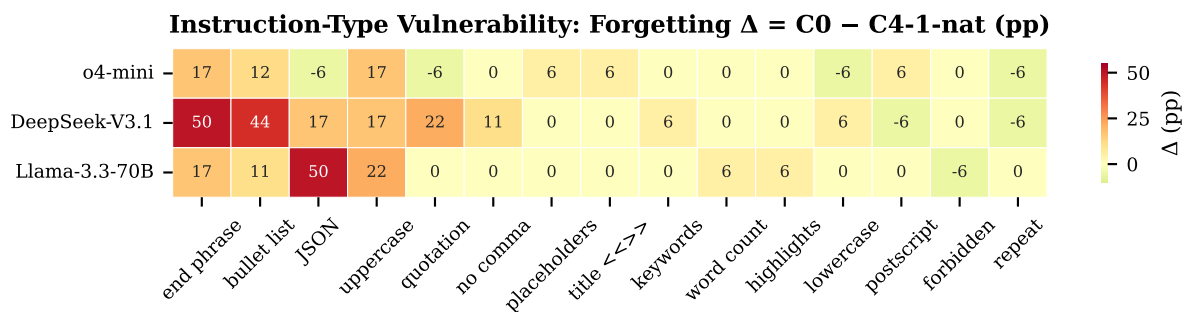


Figure 5: Instruction-type vulnerability heatmap. Each cell shows the forgetting delta (baseline minus natural + GSM8K, in %). Sorted by average vulnerability across models. Terminal and structural constraints cluster at the top (most forgotten); avoidance constraints cluster at the bottom (most robust).

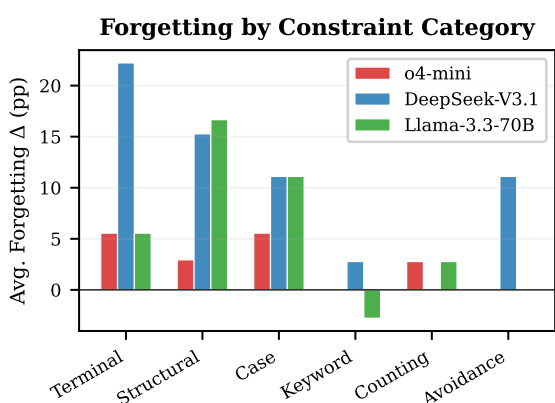


Figure 6: Average forgetting delta by constraint category. Terminal and structural constraints show the largest drops across all models.

GSM8K $\times$ 5 (82.4%); for DeepSeek, the pattern reverses (84.4% on  $\times$ 3 vs. 68.5% on  $\times$ 5).

#### 5.4 Multi-Constraint Stacking Amplifies Forgetting

Our primary experiments test one constraint at a time. When multiple formatting constraints are stacked in a single prompt, the forgetting effect becomes much worse. Table 4 shows joint compliance as a function of both constraint count ( $N$ ) and cognitive load ( $M$ ), averaged over the reminder condition ( $R$ ). The full prompt construction is described in Appendix F; the complete  $N \times M \times R$  grid, reminder effects, per-type vulnerability, and soft-tension analyses are in Appendices G and H.

The table reveals two clear patterns. First, compliance decays steeply with  $N$ : o4-mini drops from near-100% at  $N=1$  to 32–55% at  $N=5$  depending on load, and DeepSeek drops to 55–65%. Llama is most resilient, maintaining 75–95% at  $N=5$ .

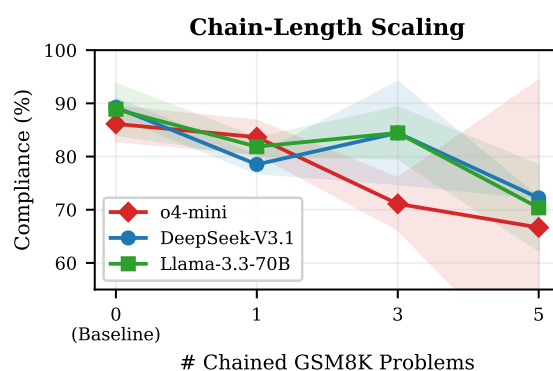


Figure 7: Compliance vs. number of chained GSM8K problems. The non-monotonic pattern is model-dependent: o4-mini dips at 3, DeepSeek at 5.

**Cognitive load compounds stacking.** Reading left to right within each model panel, adding GSM8K problems alongside stacked constraints consistently degrades compliance. For o4-mini at  $N=5$ , compliance drops from 55.0% ( $M=0$ ) to 32.5% ( $M=3$ ). The same pattern holds across models: DeepSeek drops from 65.0% to 65.0% at  $N=5$  (flat, but falls from 85.0% to 70.0% at  $N=2$ ), and Llama drops from 95.0% to 75.0% at  $N=5$ . Stacking and cognitive load act as additive sources of forgetting.

**The reminder becomes unreliable.** In the single-constraint experiments, the trailing reminder consistently recovers 10–17% of lost compliance (Table 1). Under stacking, the picture changes. The matched reminder-on minus reminder-off differences are small on average: +5.8% for o4-mini, +2.5% for Llama, and +1.7% for DeepSeek (Table 12). More importantly, the effect is inconsistent: it ranges from +25% (o4-mini,  $N=3$ ,  $M=1$ ) to -20% (o4-mini,  $N=2$ ,  $M=3$ ). In several high-load

N	o4-mini			DeepSeek			Llama		
	M=0	M=1	M=3	M=0	M=1	M=3	M=0	M=1	M=3
1	100.0	100.0	97.1	100.0	95.0	85.0	100.0	100.0	100.0
2	94.9	97.5	85.0	85.0	70.0	70.0	100.0	100.0	85.0
3	97.4	82.5	61.9	100.0	80.0	70.0	95.0	100.0	85.0
5	55.0	55.0	32.5	65.0	55.0	65.0	95.0	80.0	75.0

Table 4: Joint compliance (%) under multi-constraint stacking, as a function of constraint count  $N$  and cognitive load  $M$  (number of concurrent GSM8K problems), averaged over reminder condition  $R$ . Reading left to right shows the effect of adding load; reading top to bottom shows the effect of adding constraints. Both factors compound: o4-mini at  $N=5$ ,  $M=3$  retains only 32.5%.

cells, adding the reminder actually *hurts* compliance. A single generic reminder cannot maintain salience for multiple constraints the way it does for one.

## 6 Discussion

**Why do LLMs forget?** The formatting instruction remains fully visible in the context window throughout generation, so the forgetting we observe is not a retrieval failure in the traditional sense. We hypothesize it reflects *representational competition*: when generating tokens for complex tasks, the model’s hidden states become dominated by task-relevant representations, reducing the effective influence of the formatting constraint on token selection. This is analogous to the “monitoring failure” pathway in human PM, where attentional resources devoted to the ongoing task crowd out the monitoring process that would detect the PM cue. The reminder sentence re-elevates the constraint’s attention weight at the critical final position, just before generation begins.

### Connection to human prospective memory.

Our results replicate three classic human PM findings: (1) the *cognitive load effect*, where PM drops with demanding ongoing tasks; (2) the *cue salience effect*, where prominent cues improve PM (our reminder); and (3) the *temporal distance effect*, where deferred actions are more forgotten than continuous ones (terminal vs. pervasive constraints). The reminder works because it operates as a *retrieval cue* placed at the highest-salience position in the prompt: immediately before generation, where recent tokens naturally receive high attention weight.

**Salience as a design principle.** The reminder not only recovers lost compliance but often *exceeds* the no-task baseline (Table 1), suggesting that standard instruction formats systematically underestimate

model capability. Forgetting profiles also differ across architectures in ways that do not reduce to a simple load ordering. These patterns indicate that salience-enhancing interventions should be treated as a general prompt design principle rather than a remedial patch.

### Practical implications.

For practitioners deploying LLMs: (1) always add a reminder sentence when combining formatting instructions with complex tasks; (2) be especially vigilant with terminal constraints; (3) test compliance under realistic cognitive load, not in isolation; (4) expect that adding formatting requirements will degrade task accuracy and budget for this tradeoff; (5) when stacking multiple formatting constraints, do not rely on a single generic reminder, since its effect becomes inconsistent under stacking; consider per-constraint reminders or breaking complex format requirements into separate prompts.

## 7 Conclusion

We introduced a composition-based paradigm inspired by prospective memory research to study instruction-following failures in LLMs. Across three models and over 8,000 prompts, we find that formatting compliance drops 2–21% under concurrent task load, with terminal constraints most vulnerable; a simple salience-enhanced prompt recovers compliance to 90–100%; and when multiple constraints are stacked, joint compliance falls below 50% for one model and the reminder becomes unreliable. These findings suggest that instruction-tuning should reward compliance under load and that practitioners should treat terminal constraints as high-risk, apply reminders by default, and use structured approaches for multi-constraint scenarios.

## 8 Limitations

Our study has several limitations. We evaluate only three models from two cloud providers, limiting generalization to other architectures and scales. The IFEval sample provides only 6 items per instruction type, constraining per-type statistical power. The reminder condition modifies multiple prompt components simultaneously (constraint extraction, emphasis prefix, and trailing reminder), and we do not ablate their individual contributions. We test only single-turn scenarios; multi-turn settings may exhibit stronger forgetting. The cognitive load ordering is based on task complexity rather than empirically measured computational cost, and the non-monotonic patterns we observe suggest this ordering does not fully capture what drives forgetting. Finally, the stacking experiment uses a single run with 10 prompts per cell, limiting its statistical power.

### Ethical considerations

This work uses only publicly available datasets (IFEval, TriviaQA, MMLU, GSM8K, CNN/DailyMail) accessed through their standard distributions. No human subjects were involved. All experiments use greedy decoding (temperature 0) with fixed random seeds across three runs; note that exact bitwise reproducibility through cloud API endpoints is not guaranteed due to floating-point non-determinism, though generation-level outputs are expected to be stable. Our evaluation pipeline relies exclusively on code-based checkers with no LLM-as-judge component. Code and data will be released upon publication.

### References

Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120.

Xinyun Chen, Ryan A Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise order matters in reasoning with large language models. *arXiv preprint arXiv:2402.08939*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gilles O Einstein, Mark A McDaniel, Ruthann Thomas, Sara Mayfield, Hilary Shank, Nova Morrisette, and Jennifer Breneiser. 2005. Multiple processes in prospective memory retrieval: factors determining monitoring versus spontaneous retrieval. *Journal of Experimental Psychology: General*, 134(3):327.

Bryan Guan, Tanya Roosta, Peyman Passban, and Mehdi Rezagholizadeh. 2025. The order effect: investigating prompt sensitivity to input order in llms. *arXiv preprint arXiv:2502.04134*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Follow-bench: 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)*, pages 4667–4688.

Erik Jones and Jacob Steinhardt. 2022. Capturing failures of large language models via human cognitive biases. *Advances in Neural Information Processing Systems*, 35:11785–11799.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

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

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. 2024. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *Advances in Neural Information Processing Systems*, 37:118603–118631.

Richard L Marsh, Jason L Hicks, Gabriel I Cook, Jeffrey S Hansen, and Andrew L Pallos. 2003. Interference to ongoing activities covaries with the characteristics of an event-based intention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(5):861.

Mark A McDaniel and Gilles O Einstein. 2000. Strategic and automatic processes in prospective memory

retrieval: A multiprocess framework. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 14(7):S127–S144.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Amy Xin, Youfeng Liu, Bin Xu, Lei Hou, and Juanzi Li. 2025. Agentif: Benchmarking instruction following of large language models in agentic scenarios. *arXiv preprint arXiv:2505.16944*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Rebekah E Smith. 2003. The cost of remembering to remember in event-based prospective memory: investigating the capacity demands of delayed intention performance. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(3):347.

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

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. Certifiably robust rag against retrieval corruption. *arXiv preprint arXiv:2405.15556*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *International Conference on Learning Representations*, volume 2024, pages 40193–40219.

Zhihan Zhang, Shiyang Li, Zixuan Zhang, Xin Liu, Haoming Jiang, Xianfeng Tang, Yifan Gao, Zheng Li, Haodong Wang, Zhaoxuan Tan, and 1 others. 2025. 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)*, pages 8374–8398.

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

## A Experimental Details

### A.1 Instruction Types

Table 5 lists the 15 IFEval instruction types used in our experiments, grouped by category.

Category	Type	Count?
Case	english_capital	No
	english_lowercase	No
Keyword	existence	No
	forbidden_words	No
Terminal	postscript	No
	end_checker	No
Structural	json_format	No
	title	No
	quotation	No
Avoidance	no_comma	No
Counting	num_placeholders	Yes
	num_bullet_lists	Yes
	num_highlighted	Yes
Length	number_words	Yes
Sequential	repeat_prompt	No

Table 5: 15 IFEval instruction types used, grouped by category. “Count?” indicates whether the checker requires a numeric parameter.

### A.2 Models

Table 6 lists the three models used in our experiments. We deliberately select models from distinct architecture families to test whether prospective memory failures generalize beyond a single training paradigm. o4-mini is a reasoning-specialized model accessed through Azure OpenAI; DeepSeek-V3.1 is an open-weight model served via Azure AI Inference; and Llama-3.3-70B-Instruct is a 70-billion-parameter instruction-tuned model, also served through Azure AI. All three use the same system prompt and generation hyperparameters (§4.2).

Model	Provider	Type	Params
o4-mini	Azure OpenAI	Reasoning	N/A
DeepSeek-V3.1	Azure AI	Open-weight	N/A
Llama-3.3-70B	Azure AI	Instruct	70B

Table 6: Models evaluated. All accessed via Azure-hosted API endpoints with greedy decoding (temperature=0.0, max\_tokens=2048).

### A.3 Datasets

Table 7 summarizes the five datasets used to construct our evaluation prompts. The IFEval sam-

ple provides the formatting constraints and is held fixed across all runs (seed 42) to ensure the same 90 items are tested in every condition. The four benchmark datasets serve as distraction tasks at increasing levels of cognitive demand: TriviaQA (single-step factual recall), MMLU (four-option reasoning), GSM8K (multi-step arithmetic), and CNN/DailyMail (long-context summarization). Benchmark items are re-sampled per run using distinct seeds (42, 137, 256) to provide variance estimates. GSM8K draws 270 items per run to support single, triplet, and quintuplet chain conditions. CNN/DailyMail articles are filtered to 600–1,000 tokens (measured with the `cl100k_base` tokenizer) to ensure a consistent long-context condition.

Dataset	Split	Pool	Sampled	Load
IFEval	train	541	90	–
TriviaQA	validation	11,313	90/run	Low
MMLU	test	14,042	90/run	Medium
GSM8K	test	1,319	270/run	High
CNN/DM	test	–	45/run	Long

Table 7: Datasets and sampling. **Pool**: items in the source split. **Sampled**: items drawn per run. **Load**: cognitive load level assigned in our design. CNN/DM pool size varies after length filtering.

## B Sampling, Evaluation, and Randomization Protocols

### B.1 Statistical Details

We assess statistical significance using McNemar’s test with continuity correction on paired binary outcomes (same IFEval items, baseline vs. condition) and report Cohen’s  $h$  effect sizes ( $h = 0.2$ : small,  $0.5$ : medium,  $0.8$ : large). Bootstrap 95% confidence intervals are computed by resampling the 270 pooled item-level outcomes (90 items  $\times$  3 runs) with 10,000 bootstrap iterations. With 18 model–condition comparisons, individual  $p$ -values are not corrected for multiple testing; readers should interpret marginal results ( $p \approx .05$ ) with appropriate caution.

Table 8 reports the full statistical results for the natural-embedding experiment.

### Per-model analysis.

- **DeepSeek-V3.1** shows the strongest forgetting (avg.  $\Delta = 10.8\%$ ). Its compliance drops significantly for all four single-task conditions ( $p < .001$  for TriviaQA, MMLU, and GSM8K;  $p = .006$  for CNN/DM). The largest

single drop is on  $5 \times$  GSM8K ( $\Delta = 20.7\%$ ,  $h = 0.52$ , a medium effect).

- **o4-mini** shows moderate forgetting (avg.  $\Delta = 6.3\%$ ). Single-task effects are marginal ( $p = .052$ – $.063$ ), but the drop reaches significance for chained GSM8K  $\times 3$  ( $\Delta = 15.0\%$ ,  $h = 0.37$ ,  $p = .002$ ) and CNN/DM ( $p = .010$ ).
- **Llama-3.3-70B** is the most robust (avg.  $\Delta = 3.2\%$ ), with significant degradation only on  $+GSM8K$  ( $\Delta = 7.0\%$ ,  $h = 0.20$ ,  $p < .001$ ) and  $+3 \times GSM8K$  ( $p = .013$ ). Most other conditions show negligible effect sizes.

### B.2 Sampling and Randomization

We conduct 3 independent runs using seeds 42, 137, and 256. The IFEval sample (90 items, stratified as 6 per type  $\times$  15 types) is drawn once with seed 42 and held fixed across all runs. The four benchmark datasets are re-sampled per run using distinct seeds so that cross-run variance reflects genuine sampling variability rather than item-specific effects.

Within each run, all three models receive the same set of prompts, enabling direct paired comparisons. Prompt ordering is deterministic within a run and consistent across models. GSM8K draws 270 items per run to support single, triplet, and quintuplet chain conditions. CNN/DailyMail articles are filtered to 600–1,000 tokens (measured with `cl100k_base`) to ensure a consistent long-context condition.

### B.3 Evaluation Protocols

All evaluation is fully deterministic and requires no LLM-as-judge.

**IFEval compliance.** Each of the 15 constraint types has a dedicated code-based checker returning binary pass/fail. We report both *strict* compliance (applied to the raw model response) and *loose* compliance (applied to 8 normalized response variants: stripping leading/trailing whitespace, removing markdown headers, extracting content between quotation marks, etc.). A response passes loose compliance if *any* variant passes.

**Task accuracy.** GSM8K: we extract the last number from the response via regex and compare to the gold answer. MMLU: we extract the answer letter using a priority heuristic (“The answer is (X)”  $>$  first capital letter A–D  $>$  regex fallback) and compare to the gold label. TriviaQA: we check whether any gold alias appears as a substring of the

Model	Condition	Compliance [95% CI]	$\Delta$	Cohen’s $h$	$p$
o4-mini	Baseline	86.1 [82.0, 90.3]	–	–	–
	+ TriviaQA	82.2 [77.4, 86.7]	3.9	0.11	.063
	+ MMLU	82.0 [77.2, 86.5]	4.1	0.11	.052
	+ GSM8K	83.6 [79.2, 88.1]	2.5	0.07	.265
	+ 3×GSM8K	71.1 [61.1, 80.0]	15.0	0.37	.002**
	+ 5×GSM8K	82.4 [70.6, 92.2]	3.8	0.10	1.00
	+ CNN/DM	77.5 [70.5, 84.5]	8.6	0.23	.010**
DeepSeek	Baseline	89.3 [85.6, 93.0]	–	–	–
	+ TriviaQA	83.0 [78.5, 87.4]	6.3	0.18	<.001***
	+ MMLU	77.4 [72.2, 82.3]	11.8	0.32	<.001***
	+ GSM8K	78.5 [73.3, 83.3]	10.7	0.30	<.001***
	+ 3×GSM8K	84.4 [76.7, 91.1]	4.8	0.14	.221
	+ 5×GSM8K	68.5 [55.6, 79.6]	20.7	0.52	.013*
	+ CNN/DM	78.9 [71.9, 85.9]	10.4	0.29	.006**
Llama	Baseline	88.9 [85.2, 92.6]	–	–	–
	+ TriviaQA	87.8 [83.7, 91.5]	1.1	0.04	.546
	+ MMLU	86.2 [82.1, 90.3]	2.7	0.08	.070
	+ GSM8K	81.9 [77.0, 86.3]	7.0	0.20	<.001***
	+ 3×GSM8K	84.4 [76.7, 91.1]	4.4	0.13	.013*
	+ 5×GSM8K	87.0 [77.8, 94.4]	1.9	0.06	1.00
	+ CNN/DM	86.8 [80.6, 92.2]	2.1	0.06	.221

Table 8: Statistical summary of forgetting under natural embedding. Compliance (%) with bootstrap 95% CIs (10k resamples).  $\Delta$  = forgetting delta vs. baseline. Cohen’s  $h$ : effect size (0.2 small, 0.5 medium).  $p$ : McNemar’s test with continuity correction. Significance: \* $p$  < .05, \*\* $p$  < .01, \*\*\* $p$  < .001. Average  $\Delta$  across conditions: DeepSeek 10.8%, o4-mini 6.3%, Llama 3.2%.

response (case-insensitive). CNN/DailyMail: we compute ROUGE-L F1 using the `rouge_score` library with Porter stemming enabled; a score above 0.10 indicates task engagement.

#### B.4 Inference Configuration

All models use greedy decoding (temperature=0.0, max\_tokens=2048). For o4-mini, temperature is omitted (the API does not accept it for reasoning models) and max\_completion\_tokens replaces max\_tokens. All calls include a system prompt: “You are a helpful assistant. Follow the user’s instructions carefully and completely.” API calls use exponential backoff with up to 5 retries; responses are checkpointed after every prompt to enable recovery from transient failures.

### C Prompt and Response Length

Table 9 reports the mean input (prompt) and output (response) token counts per condition, averaged across all three models and all runs. The natural-embedding conditions have comparable prompt lengths to their reminder-enhanced counterparts, since the IFEval prompt text and benchmark content are similar in both templates. Response tokens are generally higher under natural embedding because the model also addresses the original IFEval

creative task (e.g., writing an essay), whereas the reminder template extracts only the formatting constraint.

Condition	Task	Input	Output
Baseline (C0)	–	72	538
<i>Reminder-enhanced</i>			
+ TriviaQA	Factual	97	248
+ MMLU	4-choice	166	308
+ GSM8K×1	Arithmetic	138	317
+ GSM8K×3	3×Arith.	279	644
+ GSM8K×5	5×Arith.	402	993
+ CNN/DM	Summary	849	309
<i>Natural embedding</i>			
+ TriviaQA	Factual	98	558
+ MMLU	4-choice	166	610
+ GSM8K×1	Arithmetic	139	624
+ GSM8K×3	3×Arith.	282	977
+ GSM8K×5	5×Arith.	393	1150
+ CNN/DM	Summary	854	603

Table 9: Mean input and output token counts per condition, averaged across all models and runs. Token counts are from the API response metadata.

### D Prompt Templates

This section provides the exact prompt templates used in each experimental condition. All prompts are preceded by the system message: “You are a helpful assistant. Follow all user

instructions carefully.”

## D.1 Baseline (C0): IFEval Only

### C0: Baseline

*{original IFEval prompt, used verbatim}*

### Example:

### C0 Example: english\_capital + essay task

Write an article about how intra-team conflict affected sports teams. Write in a crazy coach screaming style. Use all capital letters to express the craziness. Basically, not a single word in your entire reply should contain lowercase letters.

## D.2 Task-Only Baseline (C1)

### C1: Task Only (no formatting constraint)

*{benchmark question only}*

## D.3 Natural Embedding (C2–C5-nat)

### Natural Embedding Template

*{original IFEval prompt with embedded constraint}*  
Additionally, please also complete the following:  
*{benchmark question}*

**Example** (C2-nat: english\_capital + TriviaQA):

### C2-nat Example

Write an article about how intra-team conflict affected sports teams. Write in a crazy coach screaming style. Use all capital letters to express the craziness. Basically, not a single word in your entire reply should contain lowercase letters.  
Additionally, please also complete the following:  
Which musical instrument was Jelly Roll Morton associated with?

## D.4 Reminder-Enhanced (C2–C5)

### Reminder Template

IMPORTANT FORMATTING INSTRUCTION: *{extracted constraint text}*  
Now please help me with the following task:  
*{benchmark question}*  
Please provide your response. Remember to follow ALL of my formatting instructions above.

**Example** (C2: english\_capital + TriviaQA):

### C2 Example

IMPORTANT FORMATTING INSTRUCTION: Your entire response should be in English, capital letters only.  
Now please help me with the following task:  
Which World War 2 American general was known as ‘Vinegar Joe’?  
Please provide your response. Remember to follow ALL of my formatting instructions above.

## D.5 Constraint Stacking

### Stacking Template (with reminder)

*{base topic} {constraint 1} {constraint 2} ... {constraint N}*  
Also, please solve this math problem: *{GSM8K question}*  
Remember to follow all formatting instructions above.

**Example** ( $N=3$ ,  $M=1$ , reminder on):

### Stacking Example

Write a short paragraph about renewable energy sources. The response must contain at least 8 placeholders represented by square brackets, such as [address]. Your answer must contain exactly 1 bullet points. Use the markdown bullet points such as: \* This is a point. Your answer must contain a title, wrapped in double angular brackets, such as «poem of joy».  
Also, please solve this math problem: Stetson made a bet with Alec that he would give up \$10 for each orange he eats. While at the farm, Stetson ate 2/5 of the oranges they picked. If they picked 60 oranges, calculate the total amount of money Stetson gave up?  
Remember to follow all formatting instructions above.

## E Additional Analysis Figures

### E.1 Instruction Saliency: Reminder vs. Natural Embedding

Figure 8 compares IFEval compliance between the two prompt template designs across all four single-task conditions. The reminder-enhanced format (V1) consistently maintains higher compliance than the natural embedding (V2), with the gap widening as distraction difficulty increases. This visualizes the saliency effect discussed in §5.1 at the per-model level.

### E.2 Dual-Task Interference: Both Directions

Figure 9 visualizes the bidirectional cost of combining a formatting constraint with a benchmark task. The compliance drop (how much formatting adherence degrades) and the accuracy drop (how much task performance degrades) are shown side by side for each model under the GSM8K condition. This complements the task accuracy results in the main

Instruction Saliency Effect: Highlighted (V1) vs Natural (V2)

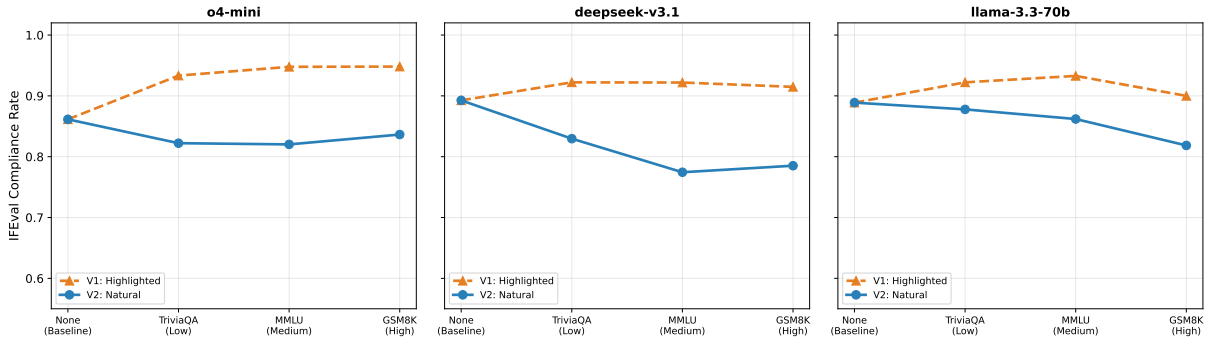


Figure 8: Instruction saliency effect. IFEval compliance under the reminder-enhanced format (V1, dashed) vs. natural embedding (V2, solid) across four distraction conditions, shown separately for each model. The gap between V1 and V2 reflects the compliance recovered by making the constraint more salient.

body by showing both interference directions in a single view.

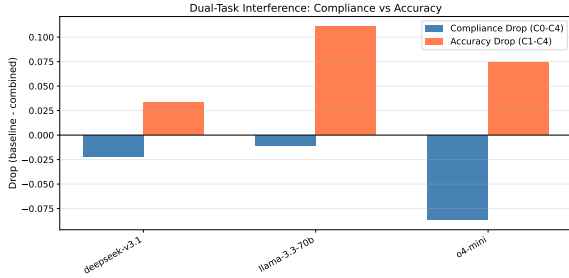


Figure 9: Dual-task interference for each model under GSM8K. Blue bars: compliance drop (baseline IFEval compliance minus combined). Coral bars: accuracy drop (task-only GSM8K accuracy minus combined). Both directions show degradation, confirming symmetric interference.

## F Constraint Stacking: Experimental Details

### F.1 Safe Pool

Table 10 lists the 11 IFEval instruction types in the safe pool for the stacking experiment. Four types are excluded: `english_lowercase` (directly conflicts with `english_capital`), `json_format` (forces a global structure incompatible with other formatting), `end_checker` (overlaps with `postscript` and `quotation`), and `repeat_prompt` (prepends a verbatim copy that interferes with other constraints).

Two hard-conflict pairs are prevented from co-occurring within a prompt:

- `postscript` + `quotation`: both impose boundary requirements.

Type	Description
<code>english_capital</code>	Entire response in capital letters
<code>existence</code>	Include specified keywords
<code>forbidden_words</code>	Avoid specified keywords
<code>postscript</code>	End with a postscript (P.S.)
<code>title</code>	Include a title in angular brackets
<code>no_comma</code>	Avoid all commas
<code>number_placeholders</code>	Include $N$ placeholders in brackets
<code>number_bullet_lists</code>	Include exactly $N$ bullet points
<code>number_words</code>	Response within $N$ words
<code>number_highlighted</code>	Highlight $N$ sections with markdown
<code>quotation</code>	Wrap response in double quotes

Table 10: The 11 IFEval instruction types in the safe pool for constraint stacking.

- `no_comma` + `number_bullet_lists`: bullet formatting conventionally requires commas.

### F.2 Soft-Tension Pairs

Four pairs are tested separately to examine pair-specific interactions:

1. `no_comma` + `number_bullet_lists`: Bullet lists conventionally use commas, making comma avoidance harder.
2. `postscript` + `quotation`: Both constrain response boundaries (postscript at the end, quotation marks wrapping everything).
3. `english_capital` + `existence`: Required keywords may conflict with capitalization conventions for domain terms.
4. `number_words` + `number_bullet_lists`: Hitting a word count target while producing exactly  $N$  bullets can create competing pressures.

$N$	o4-mini				DeepSeek				Llama			
	$M=0$		$M=3$		$M=0$		$M=3$		$M=0$		$M=3$	
	off	on	off	on	off	on	off	on	off	on	off	on
1	100	100	94	100	100	100	80	90	100	100	100	100
2	95	95	95	75	80	90	80	60	100	100	90	80
3	100	95	50	74	100	100	80	60	90	100	80	90
5	50	60	25	40	60	70	50	80	100	90	70	80

Table 11: Full stacking grid: joint compliance (%) by  $N$ ,  $M$  ( $M=1$  omitted for space; pattern interpolates), and  $R$  (off/on). Cells where the reminder hurts: o4-mini  $N=2/M=3$  (75 vs. 95,  $\Delta=-20$ ), DeepSeek  $N=5/M=1$  (40 vs. 70,  $\Delta=-30$ ). Cells where it helps most: o4-mini  $N=3/M=3$  (74 vs. 50,  $\Delta=+24$ ).

### F.3 Prompt Construction and Grid

Each stacking prompt is assembled from four components: (1) a neutral base topic (one of 15 subjects such as “the history of bridges” or “how weather forecasting works”), (2)  $N$  constraint instructions with realistic parameters sampled from IFEval metadata, (3)  $M$  GSM8K problems drawn from the test set, and (4) an optional trailing reminder (“Remember to follow ALL of the formatting instructions listed above.”). Constraints are listed as labeled items (Constraint 1, Constraint 2, etc.) before the base topic, and GSM8K problems are appended after the topic.

The safe-pool grid crosses  $N \in \{1, 2, 3, 5\}$ ,  $M \in \{0, 1, 3\}$ , and  $R \in \{\text{off}, \text{on}\}$  (24 cells, 10 prompts each, 240 total). The tension grid fixes  $N=2$  and crosses 4 pairs  $\times M \times R$  (24 cells, 10 prompts each, 240 total). Combined: 480 prompts  $\times 3$  models = 1,440 API calls.

## G Stacking: Full Results

### G.1 Full Stacking Grid ( $N \times M \times R$ )

Table 11 reports joint compliance for every cell of the stacking design, broken down by constraint count ( $N$ ), cognitive load ( $M$ ), and reminder ( $R$ ). This backs up the cell-level claims in §5.4: the reminder lift ranges from +25% (o4-mini,  $N=3$ ,  $M=1$ ) to  $-20\%$  (o4-mini,  $N=2$ ,  $M=3$ ).

### G.2 Reminder Effect Under Stacking

Let  $J_m(N, M, R)$  denote joint compliance for model  $m$  at a fixed number of constraints  $N$ , cognitive load  $M$ , and reminder setting  $R$ . The reminder effect reported in Table 4 is the matched on–off

difference averaged over the 12 safe-pool cells:

$$\Delta_R^{(m)} = \frac{1}{12} \sum_{N \in \{1, 2, 3, 5\}} \sum_{M \in \{0, 1, 3\}} d_m(N, M), \quad (2)$$

$$d_m(N, M) = J_m(N, M, \text{on}) - J_m(N, M, \text{off}). \quad (3)$$

Model	$R=\text{off}$	$R=\text{on}$	$\Delta_R$
o4-mini	77.0	82.8	+5.8
DeepSeek	77.5	79.2	+1.7
Llama	91.7	94.2	+2.5

Table 12: Mean joint compliance (%) by reminder setting under stacking. Values average over all 12 safe-pool  $N \times M$  cells.  $\Delta_R$  is the final column minus the middle column and corresponds to the  $\Delta_R$  column in Table 4.

### G.3 Per-Type Vulnerability Under Stacking

The vulnerability pattern from the single-constraint experiments carries over under stacking. Avoidance constraints (no\_comma: 100% pass rate across all models, forbidden\_words: 99–100%) remain nearly immune even at  $N=5$ . Counting and content-insertion constraints are most vulnerable: number\_highlighted\_sections averages only 71–77% for o4-mini and DeepSeek. Table 13 shows the full per-type breakdown.

## H Stacking: Soft-Tension Ablation

Table 14 shows the average joint compliance gap between each tension pair and the safe-pool  $N=2$  baseline, averaged across all  $M$  and  $R$  settings. A negative value means the tension pair performs worse than the safe-pool average at  $N=2$ .

The number\_words + number\_bullet\_lists pair consistently degrades compliance across all three models ( $-15\%$  to  $-22\%$  on average), confirming that meeting a word count target while also

Constraint Type	o4-mini	DeepSeek	Llama
num_highlighted_sections	76.6	71.4	96.4
number_bullet_lists	81.9	75.0	97.2
quotation	77.2	91.7	100.0
number_placeholders	82.2	88.4	100.0
number_words	88.3	87.0	95.9
existence	76.7	95.6	100.0
postscript	77.8	100.0	98.6
english_capital	97.5	97.5	90.4
title	86.2	100.0	100.0
forbidden_words	99.0	100.0	100.0
no_comma	100.0	100.0	100.0

Table 13: Per-type pass rate (%) under stacking, sorted by average vulnerability across models. Counting constraints are most vulnerable; avoidance constraints remain immune.

Tension Pair	o4-mini	DS	Llama
no_comma + bullet_lists	-17.5	+21.7	-6.7
postscript + quotation	+5.9	+23.3	+1.7
english_cap + existence	-1.9	+16.7	0.0
num_words + bullet_lists	-21.6	-16.7	-15.0

Table 14: Average tension penalty ( $\Delta$  in %, tension minus safe  $N=2$ ). DS = DeepSeek-V3.1. Negative means the tension pair degrades compliance below the safe-pool baseline. The number\_words + number\_bullet\_lists pair shows the largest penalty across all models.

producing a specific number of bullets creates real interference. The postscript + quotation pair, despite being excluded from the safe pool as a hard conflict, performs at or above the safe-pool baseline. This suggests that its classification as a hard conflict was conservative; the two constraints may actually be complementary rather than competing, since both are boundary-focused requirements that the model can satisfy with a single response structure.

Note that the safe-pool  $N=2$  baseline uses randomly drawn constraint pairs and includes all 11 types, some of which (e.g., number\_highlighted\_sections) have low individual pass rates. Tension pairs composed of individually easy constraints (like no\_comma or postscript) may therefore appear to outperform the safe-pool average even if they impose real tension, because their constituent types are inherently easier. The most meaningful comparison is for pairs where both constraints have moderate individual pass rates, making the number\_words + number\_bullet\_lists result (consistently negative across all models) the cleanest signal of pair-specific interference.

## I Taxonomy of Forgetting Mechanisms

To understand *how* models forget, we manually inspected all 187 instances in which a model completed the distractor task correctly<sup>1</sup> but violated the formatting constraint it satisfied in the C0 baseline. We identified five recurring failure mechanisms, summarized in Table 15; one representative example for each follows. Examples are drawn from different models, conditions, and distractor tasks to illustrate the generality of the phenomenon.

Mechanism	$N$	Description
Domain shift	55	Constraint held for IFEval subtask, dropped at task transition
Structural abandon	51	Formatting maintained for first subtask, replaced for second
Terminal omission	44	Required ending produced mid-response; generation continues
Spatial compartm.	19	Global constraint applied only to one output section
Content/length	18	Required elements or length constraints not met

Table 15: Taxonomy of forgetting mechanisms.  $N$  = number of task-correct forgetting failures across all models and runs (187 total).

### I.1 Domain Shift: Case Violation Under TriviaQA

Llama-3.3-70B, reminder-enhanced condition with TriviaQA (C2). The constraint requires the entire response to be in capital letters.

**Analysis.** The model writes 95.1% of alphabetic characters in uppercase but reverts to title case for the proper noun “Binnacle” (14 of 288 alphabetic characters are lowercase). The model’s world knowledge about conventional capitalization of domain-specific terms overrides the active formatting constraint. The task answer is correct, and the constraint would have been satisfied in the C0 baseline.

<sup>1</sup>For CNN/DailyMail, where task correctness is a continuous metric, we use ROUGE-L  $> 0.10$  as the criterion.

#### Example: Domain Shift (Case)

##### Prompt

IMPORTANT FORMATTING INSTRUCTION: Your entire response should be in English, capital letters only.  
Now please help me with the following task: What is the (traditionally wooden) housing or stand which contains a ship's compass? Please provide your response. Remember to follow ALL of my formatting instructions above.

##### Response

THE HOUSING OR STAND WHICH CONTAINS A SHIP'S COMPASS IS CALLED A BIMINI. HOWEVER, THE MORE TRADITIONAL TERM FOR THIS IS A **Binnacle**. A BIMINACLE IS ALSO SOMETIMES USED TO REFER TO THIS, BUT **Binnacle** IS THE MORE COMMONLY USED TERM. IT IS TYPICALLY A WOODEN HOUSING THAT CONTAINS AND PROTECTS THE COMPASS, WHILE ALSO PROVIDING A STABLE PLATFORM FOR IT TO SIT ON.

## I.2 Domain Shift: Punctuation Forgotten During Arithmetic

DeepSeek-V3.1, natural-embedding condition with GSM8K (C4-1-nat). The constraint is to avoid all commas.

#### Example: Domain Shift (Punctuation)

##### Prompt

Rewrite the sentence "A bust of a man with a beard and mustache." in a more sophisticated way. Do not use commas in your response. Additionally, please also complete the following:  
Zeke's baseball team has 7 more players than Carlton's. If Carlton's team has 13 players, how many players are there in both teams combined?

##### Response

A sculpted bust depicts a bearded and mustachioed gentleman.  
Carlton's team has 13 players. Zeke's team has 7 more, so it has  $13 + 7 = 20$  players. Combined, both teams have  $13 + 20 = 33$  players.

**Analysis.** The model successfully avoids commas in the creative-writing subtask (rewriting the sentence) but introduces two commas in the arithmetic subtask. This pattern, which we term *domain-shift forgetting*, shows the constraint being maintained during the original IFEval task but dropped when the model transitions to mathematical reasoning. The math answer (33) is correct.

## I.3 Spatial Compartmentalization Under TriviaQA

DeepSeek-V3.1, natural-embedding condition with TriviaQA (C2-nat). The constraint is to wrap the *entire* response in double quotation marks.

#### Example: Spatial Compartmentalization

##### Prompt

Write a speech-like paragraph on the US elections. Make sure to wrap your entire response with double quotation marks. Additionally, please also complete the following:  
'Boat Race' is Cockney rhyming slang for which part of the body?

##### Response

"Fellow citizens, we stand at a pivotal moment in our nation's journey, a time when the very future of our democracy is placed in our hands. [. . .] a shared commitment to the enduring promise of America."  
"Boat Race" is Cockney rhyming slang for 'face'.

**Analysis.** The model correctly wraps the speech paragraph in quotation marks, then produces a *separate* line for the trivia answer (also with structural quotation marks around "Boat Race"). Because the two parts are not enclosed in a single pair of quotation marks, the entire response fails the strict checker. This illustrates *spatial compartmentalization*: the model treats the dual task as two independent outputs, applying the constraint locally rather than globally.

## I.4 Structural Abandon Under Arithmetic

o4-mini, natural-embedding condition with GSM8K (C4-1-nat). The constraint requires exactly 3 markdown bullet points using the \* marker.

**Analysis.** The model correctly produces three \*-marked bullet points for the naming subtask, matching the required format. However, the arithmetic breakdown uses a different list format (- dashes), adding four dash-prefixed items. The IFEval checker counts all markdown list markers, yielding seven list items rather than the required three, a format violation introduced entirely by the distractor task's content. The math answer (34) is correct.

### Example: Structural Abandon

#### Prompt

Name exactly 3 names for a black and white dog using markdown bullet points such as:  
\* Bullet point 1

Additionally, please also complete the following:

As Sally walked to school, she was holding the strings to 25 red balloons, 7 green balloons, and 12 yellow balloons. Suddenly, a gust of wind caused 40% of the red balloons to burst. [. . .] What number of balloons did she finally carry into the school?

#### Response

- \* Oreo
- \* Panda
- \* Domino

Sally's balloon count step by step:

- Red balloons:  $25 - (40\% \text{ of } 25 = 10) = 15$
  - Green balloons: 7
  - Yellow balloons:  $12 - (1/2 \text{ of } 12 = 6) = 6$
  - Blue balloons added:  $75\% \text{ of } 8 = 6$
- Total carried into school =  $15 + 7 + 6 + 6 = 34$  balloons.

### Example: Terminal Omission

#### Prompt

Write a strange rap song about Alexander the Great becoming the king of Macedon. Finish the song with:  
Peace!

No additional words should follow "Peace!"

Additionally, please also complete the following:

Please read the following article and write a concise summary of the key points. Article: Gordon Robson, 26, killed grandfather John Potts, 45, with a single punch [. . .]

#### Response

[. . . six rap verses. . .]

Kings and ghosts bow low to the madness of my throne-

I'm Alexander the Great, in every cosmic bone!

**Peace!**

Summary of Key Points:

- Gordon Robson, 26, killed John Potts, 45, with a single punch [. . .]
- Robson had been drinking after his grandfather's funeral [. . .]
- Potts, a father-of-six, stepped in to calm the fight [. . .]
- Robson admitted manslaughter; sentenced to three and a half years.

## I.5 Terminal Omission Under CNN/DailyMail

o4-mini, natural-embedding condition with CNN/DailyMail (C5-nat). The constraint requires the response to end with "Peace!" with no additional words following.

**Analysis.** The model produces a complete rap that ends precisely with "Peace!" at the natural artistic conclusion. However, rather than stopping, it generates a five-bullet CNN/DailyMail summary (135 additional tokens follow "Peace!"). The terminal constraint is satisfied *within the first subtask* but violated *globally*. This pattern (which we term *terminal omission*) is especially prevalent under long-context summarization (C5-nat), where substantial additional content creates strong generation pressure past the intended stopping point. The failure is consistent across all three experimental runs with this IFEval item (ROUGE-L = 0.18), demonstrating the robustness of this failure mode.