ONE TOKEN AWAY FROM COLLAPSE: THE FRAGILITY OF INSTRUCTION-TUNED HELPFULNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-tuned large language models produce thorough responses when prompted normally. We show that this helpfulness is fragile. Adding a trivial constraint, such as a ban on a single punctuation mark or on the word "the", costs **17–48%** of comprehensiveness on average in pairwise evaluation of seven models, and all three judges detect the loss. The loss is in content rather than presentation: content criteria degrade 1.9–$3.1\times$ more than surface criteria, and a blinded human evaluation reproduces the pattern. The trigger is the constraint's *presence in the prompt*, not the effort of obeying it. Four converging tests support this, including bans on words that essentially never occur, which still cost 10–21%, and decoder-level enforcement of the identical constraint with a clean prompt, which removes almost all of the loss ($-26.0\% \rightarrow -4.5\%$). The response-strategy decision is made before generation, where a linear probe can detect it ($R^2 = 0.66$–0.94), and it is reversible at a single mid-network layer: steering along a difference-of-means direction with a compliance-preserving decoder mask recovers 64–82% on Qwen-2.5-7B at 100% constraint satisfaction, with no training, and at most 14% at the six other depths tested. Beyond the best steering dose, responses stay near full length while comprehensiveness recovery falls by 18 points, so the judge is not rewarding length. The collapse extends to deployment constraints (14–31%), independent scoring underestimates it by 2.0–$6.8\times$, and LoRA fine-tuning on 80 examples recovers 65–86%.

1 INTRODUCTION

Instruction tuning is the standard method for producing helpful assistants (Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023). The implicit assumption is that it teaches *generalizable helpfulness*: thorough responses regardless of surface-level formatting. A well-aligned model should explain gradient descent equally well with bullet points or without commas.

We challenge this assumption: appending "Do not use any commas in your response" makes the model switch to a much shorter, less helpful answer instead of rephrasing the same content without commas (Figure 1). We measure this with pairwise judging: a judge rates both responses, side by side, for *comprehensiveness*. Banning commas costs 27% on Llama-3.1-8B-Instruct and 61% on Qwen-2.5-7B-Instruct. The output remains free-form prose, and the model produces less content without violating any format.

The loss is in content rather than presentation. Constrained responses might merely *appear* worse because they are shorter and less formatted, but three measurements show otherwise. A six-criterion instrument that scores content and surface separately finds that content falls 1.9–$3.1\times$ further, and on the most robust model, surface is unchanged (-1.1%) while content falls 14.2% (§4.2). A blinded human evaluation reproduces this separation (§4.3). Our steering experiments also produce responses of *identical* length whose comprehensiveness recovery differs by 18 points (§6.4), so the judge is not rewarding length.

The trigger is the clause rather than compliance. A natural explanation is that constrained generation is harder, so the model gives up content to comply, but this explanation is wrong. Four converging tests show that the model reacts to *having been given a restriction* rather than to obeying one (§5). Enforcing the identical constraint at the decoder with a clean prompt removes almost all of

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

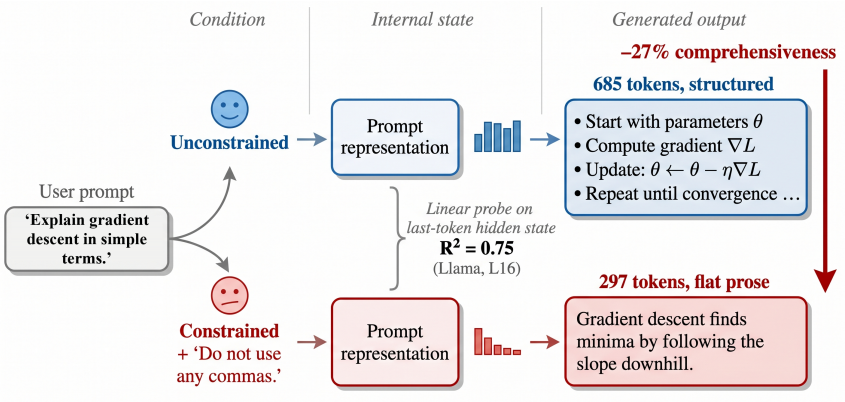

Figure 1: **Constraint-induced response collapse.** Adding "Do not use any commas" to the same prompt makes Llama-3.1-8B-Instruct switch from structured answers to plain prose: averaged over the 40 prompts, responses shrink from 685 to 297 tokens and lose 27% of their comprehensiveness. The task, the knowledge required, and the permitted output format are unchanged.

the loss ($-26.0\% \rightarrow -4.5\%$), and banning words that would never have appeared, though effortless to obey, still costs 10–21%.

The decision is localized and reversible. The failure can be detected before generation and changed by intervention. A two-pass protocol (generate freely, then rewrite under the constraint) retains 59–96% of baseline length, so the model *can* produce comprehensive constrained prose but does not *plan* to. A linear probe on the last prompt token predicts response length with $R^2 = 0.66$–0.94, and this predictability tracks collapse severity ($r = 0.93$). Steering along a difference-of-means direction at a single mid-network layer, with a decoder mask holding satisfaction at 100%, recovers 64–82% of the loss on Qwen-2.5-7B with no training. Recovery is 79% at layer 15 and between -23% and $+14\%$ at the six other depths tested (§6.3).

Contributions. • **Phenomenon. Constraint-induced response collapse**: 17–48% comprehensiveness loss across seven instruction-tuned models (five families, 7B–70B, open- and closed-weight), via 4,100+ pairwise comparisons, three judges, a blinded human evaluation, and a content/surface decomposition (§4).

• **Trigger.** The collapse is caused by the constraint's **presence in the prompt**, not the difficulty of complying, as four converging tests show (§5).

• **Mechanism and control.** We localize the planning decision to a **single layer** of the seven tested and reverse it there at inference time at 100% constraint satisfaction with no training; a dose–response curve separates length from content (§6).

• **Scope and mitigation.** The effect extends to deployment constraints (14–31%), independent scoring underestimates it by 2.0–6.8×, and LoRA on 80 examples recovers 65–86% (§7).

The paper is primarily diagnostic; the steering and LoRA results establish that the fragility is causally reversible and learnable, and production-grade solutions remain future work.

2 Related Work

Format effects and constraint satisfaction. Tam et al. (2024) show structured formats degrade reasoning accuracy, and Lee et al. (2026) measure this "format tax" systematically, finding prompt-level effects dominate and closed-weight models largely resist it. Our results go further: collapse occurs with an unrestricted output format, the trigger can be one banned token, closed-weight models are also affected (GPT-4o-mini: 31%), and we identify the causal locus, which prior work does not. Deng et al. (2026) propose decoupled generation, which is complementary. A parallel literature evaluates whether models *satisfy* constraints (IFEval (Zhou et al., 2023) and its multi-constraint (He et al., 2024a; Wen et al., 2024), multi-turn (He et al., 2024b) and agentic (Qi et al., 2025) successors) and trains them to satisfy more (Dong et al., 2025; Zhang et al., 2025). These measure satisfaction

rates, not what happens to *quality* when constraints are followed; our work addresses a separate question, since quality degrades even under full compliance.

Constrained decoding, steering, prompt sensitivity. Grammar-constrained decoding (Willard & Louf, 2023; Geng et al., 2023) masks invalid tokens at the decoder, and grammar-aligned decoding (Park et al., 2024) corrects the resulting distortion. Our constraints are prompt-level, where Lee et al. (2026) find the largest effects; we use decoder enforcement as a causal control (§5), not a remedy, since it needs logit access and most constraints of interest have no token-level form. Stolfo et al. (2025) steer to improve constraint *satisfaction*; we hold satisfaction at 100% with a mask and steer to recover *quality*. Work on prompt sensitivity (Sclar et al., 2024; Mizrahi et al., 2024; Dong et al., 2026) studies semantically equivalent reformulations; we study explicit constraints that change a permitted response's surface form without changing the task, so the failure is an inability to maintain quality rather than a misunderstanding.

3 Preliminaries and Experimental Setup

Notation. Let q be a question and c a constraint clause appended to it; r_0 responds to q alone, r_c to $q \oplus c$. Our statistic $\Delta\%$ is the relative change in mean judged comprehensiveness from r_0 to r_c; the *baseline win rate* is the fraction of pairs where r_0 scores higher. Under *independent* scoring the judge sees one response; under *pairwise* scoring, both side by side with positions randomized; "$\Delta\%$" means pairwise unless stated. The judge always receives only q, never c, so it cannot reward or penalize a response for the constraint it was written under. **Constraint-induced response collapse** denotes $\Delta\% \ll 0$ when c is added to the prompt and r_c stays well-formed prose, whether or not r_c satisfies c (satisfaction rates in Appendix B); it excludes *degeneration* (5-gram repetition verified throughout §6).

Prompts, constraints, models. We use 40 prompts across four categories (Appendix O) and eight lexical constraints: punctuation bans (commas, colons, semicolons), a pattern ban (bullets, lists, dashes; it bridges token bans and the structural restrictions of the format-tax literature), word bans ("the"; discourse markers), and two compositional constraints (Appendix P). §5.2 adds 15 single-word bans spanning the frequency range and §7.1 four deployment constraints. Each is appended as a natural-language instruction; output format stays free-form. We evaluate seven instruction-tuned models across five families: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen-2.5-7B-Instruct (Yang et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) for detailed analysis, plus Llama-3.3-70B-Instruct, Qwen3-30B-A3B-Instruct (Yang et al., 2025) (MoE), OLMo-3-7B-Instruct (Team Olmo et al., 2025) and GPT-4o-mini (Hurst et al., 2024). We also evaluate the three open-weight base models. Open-weight models run in bfloat16.

Generation and evaluation. Three samples per prompt–constraint pair (temperature 0.7, top-p 0.9, max 1024 tokens); pairwise evaluation uses sample 0, and results are unchanged under greedy decoding (Appendix G.3). Judges are GPT-4o-mini, GPT-4o and Claude Sonnet 4.6. Judge prompts, confidence intervals and judge-reseeding variance are in Appendices Q and A.2.

Constraint satisfaction. Automated checkers measure satisfaction on every response. Llama and Qwen satisfy the eight constraints at means of 93.7% and 95.6%. Mistral is much lower on punctuation and word bans (`no_comma` 2%, `no_the` 8%) while high on `no_semicolon` (98%), `no_discourse` (86%) and `no_bullet` (82%); the full table is in Appendix B. Mistral's non-compliance is a direct test rather than a confound, and we return to it in §5: on `no_comma`, none of the 40 Mistral responses satisfies the constraint, yet they still lose 24.5% of comprehensiveness.

4 Constraint-Induced Response Collapse

4.1 Main Results

Instruction-tuned models lose comprehensiveness under lexical constraints, consistently across all three primary families, 23 of the 24 model–constraint pairs and all four prompt categories (Figure 2; category breakdown in Appendix C).

The effect is **large and consistent**: 14–40% overall loss, which on the 1–10 scale is a fall in mean comprehensiveness from 9.22 to 7.07 (Llama), 9.12 to 5.48 (Qwen) and 8.67 to 7.43 (Mistral), with

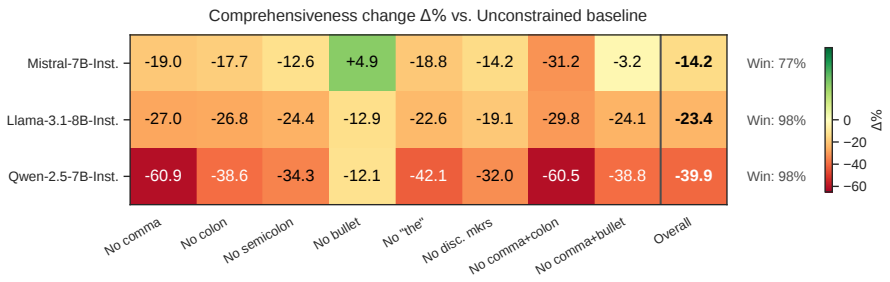

Figure 2: **Pairwise comprehensiveness evaluation.** Relative change $\Delta\%$ vs. unconstrained baseline (GPT-4o-mini judge, 40 prompts $\times$ 8 constraints per model). Darker red indicates larger collapse. Full numerical results in Appendix A.1.

Table 1: **The collapse is concentrated in content rather than surface quality** (six-criterion pairwise instrument, same judge and protocol as Figure 2, 320 pairs per model). Content averages semantic coverage, comprehensiveness and helpfulness; surface averages conciseness and readability. A presentation account predicts equal drops in both.

Model	Compr. $\Delta\%$	Content $\Delta\%$	Surface $\Delta\%$	Correctness $\Delta\%$
Mistral-7B-Instruct	-18.1	-14.2	$\mathbf{-1.1}$	-0.5
Llama-3.1-8B-Instruct	-27.1	-21.7	-7.0	-1.4
Qwen-2.5-7B-Instruct	-42.9	-37.3	-20.0	-8.6

the unconstrained answer winning 77–98% of comparisons. Every per-constraint effect on Llama is significant at $p < 10^{-7}$, and judge re-seeding moves each value by at most 1.5 points (Appendix A.2). Models differ in fragility (Qwen -39.9%, Llama -23.4%, Mistral -14.2%), and the effect is **not specific to formatting tokens**: banning "the" (-22.6% / -42.1%) is comparable to banning commas (-27.0% / -60.9%). Compositional constraints reach a floor rather than degrading continuously, which is consistent with a discrete strategy switch. All three judges detect the collapse with consistent severity ordering (Appendix A.4), and it replicates on MT-Bench (Appendix E) and 400 AlpacaEval prompts (Dubois et al., 2024) (Appendix F). A second, independent generation of all responses, judged the same way, reproduces the overall effect (-24.7 / -39.6 / -15.6% for Llama / Qwen / Mistral); the six-criterion instrument (§4.2), the controls of §4.3 and the tests of §5 use this second run.

Under GPT-4o, all seven instruction-tuned models collapse, from -17.4% (Mistral-7B) to -48.1% (Qwen-2.5-7B), at baseline win rates of 78–100% (Table 15). **Scale does not help**, as Llama-70B collapses *more* than Llama-8B (-34.7% vs. -25.9%). **Proprietary training does not prevent it** (GPT-4o-mini loses 31.0%), and **fully open training does not help** (OLMo-3-7B loses 37.7%). The MoE architecture of Qwen3-30B-A3B does not prevent it either (-37.9%). Per-model breakdowns are in Appendices D and A.

4.2 Content Quality versus Surface Quality

The main alternative explanation is that constrained responses contain the same information but look less *impressive* (shorter, less structured, without markup), so a judge that responds to presentation rates them lower. We test this by decomposing the judgment: we re-score all three primary models with a six-criterion instrument that rates content quality, correctness and surface quality separately, using the same pairwise protocol and judge.

Table 1 shows three results. First, the comprehensiveness loss is *larger* under the richer instrument than under the two-criterion instrument behind Figure 2, so the headline numbers are not an artifact of a coarse rubric. Second, content falls 1.9–3.1$\times$ more than surface on Llama and Qwen. In the clearest case, **Mistral's surface quality is unchanged (-1.1%) while its content falls 14.2%**, so the constrained responses are equally readable but contain much less. Third, correctness falls least (-0.5% to -8.6%), as expected if content is omitted, but not if it is corrupted. An atomic-claim analysis confirms the omission directly: only 49.8% of the baseline's factual claims appear in the constrained response, even with generous paraphrase matching, so the model drops claims rather than compressing them (Appendix H).

Table 2: **Four tests of whether compliance or clause presence drives the collapse.** Each row varies the presence of the clause while holding compliance or length fixed, or the reverse. GPT-4o-mini judge; 40 prompts per cell except line 1b (steering runs, §6.3) and line 4 (53 matched pairs).

	Test	Result
1a	Same constraint enforced at the *decoder*, prompt left clean	$-26.0\% \rightarrow \mathbf{-4.5}\%$ (Llama)
1b	Same decoder mask, clause kept in the prompt (§6.3)	$\mathbf{0}\%$ to -3% recovery
2	Banning words that never occur (e.g., `zephyr`, `bezoar`)	$\mathbf{-12.2}$ / -21.2 / -10.1%
3	Mistral on `no_comma`: **0 of 40** responses comply	$\mathbf{-24.5}\%$
4	Short answers *selected*, not instructed, at matched length	instructed is $\mathbf{7.4}\%$ worse, $p = 0.0004$

4.3 Human Evaluation and Setup Controls

Ten STEM-trained evaluators rated blinded pairwise comparisons on six dimensions across 40 prompts $\times$ 8 constraints $\times$ 3 models (960 comparisons, 5,760 ratings; Appendix S). Human-rated comprehensiveness drops (-17.4% Mistral, -27.0% Llama, -46.9% Qwen) closely match the LLM judges, with identical severity ordering. Content criteria drop 1.5–$2.3\times$ more than surface criteria, and all results are significant ($p < 0.001$, Wilcoxon). As a built-in validity check, evaluators rate Mistral's `no_bullet` responses *better* than baseline (win rate 45%), matching the LLM judge's $+4.9\%$. Evaluators therefore reward the constraint where removing bullets helps, rather than preferring longer text (Appendices A.6 and A.12).

Four further controls rule out setup-specific explanations. Chain-of-thought reduces the collapse by 16–43% but never removes it, and a native reasoning phase has no effect (Appendix G.1). Stripping all markup from the baseline reference leaves the collapse intact (-23.3 / -45.0 / -21.2% against -25.5 / -45.6 / -20.1% with the original reference; Appendix G.2). Greedy decoding reproduces the sampled results to within 1.2 points (Appendix G.3). Adding "keep your answer equally thorough" leaves half to two thirds of the loss in place (Appendix G.4).

5 What Triggers the Collapse

A constrained response seems harder to write, so the natural explanation of §4 is that the model gives up content to comply. This section shows that this explanation is wrong.

5.1 Four Converging Tests

Table 2 collects four tests from different models, constraints and designs, run for different immediate purposes, and all four lead to the same conclusion.

Line 1 is the clearest test. On the two bans it covers (`no_comma` and `no_the`), moving the identical constraint from the prompt to the decoder removes almost all of the loss (Qwen: $-50.9\% \rightarrow +0.3\%$), while the same mask with the clause still in the prompt recovers nothing (1b). Lines 2 and 3 vary compliance in opposite directions (automatic in the first, absent in the second), and the loss persists in both. Mistral does not try to follow the comma ban: it writes more commas per 100 words under the ban than without it (7.0 vs. 6.3). Line 4 holds length fixed. We generate 20 unconstrained samples per prompt and *select* those close to the constrained length; a response told to answer in approximately that many words is still 7.4% less comprehensive than the naturally short one ($p = 0.0004$). Writing less costs some comprehensiveness, and being *told* to write less costs more. That experiment also shows that Llama's shortest natural answer in 20 draws is 393 words, against 261 under a comma ban, so the collapsed response lies outside the model's unconstrained length distribution (Appendix G.5).

5.2 What Sets the Magnitude: Banned-Token Frequency

To find what sets the size of the collapse, we add 15 single-word bans spanning the frequency range (600 comparisons per model). The loss grows monotonically with how often the banned word would have appeared: from -12.2 / -21.2 / -10.1% (Llama / Qwen / Mistral) for words that essentially never occur, through -13.7 / -23.0 / -11.4 and -17.9 / -32.1 / -14.1, to -22.8 / -40.9 / -19.2% for words that occur constantly. The ordering holds in all three models without exception (Appendix A.8).

This gives a two-part account: **the clause triggers a re-planning decision, and the cost of that decision scales with how much of the learned response template depends on the banned material.** The two parts are separable, and the lowest-frequency band shows why they must be kept separate: a ban that triggers re-planning without disrupting any template still costs 10–21%. The gradient also explains apparent exceptions (Mistral's inert `no_bullet`, Llama's single positive MT-Bench category, and larger models collapsing harder); in each case the amount of template that can be disrupted differs (Appendix A.8).

5.3 Base Models: the Low-Template Endpoint

Base models are at the bottom of this gradient: their unconstrained output is loosely structured, so there is little template to disrupt. Running the identical experiment on the base counterparts of all three families gives small and inconsistent effects, while the instruction-tuned models collapse severely (Figure 3): Qwen base +7.0%, Llama base −8.0% (55% win rate, near chance), Mistral base −11.0% (59%), against −17.4% to −48.1% for their tuned counterparts.

This result does not depend on compliance. Base models are poor instruction followers, with correspondingly low satisfaction rates (Llama-3.1-8B: `no_the` 5.8%, `no_comma` 20.8%; Qwen-2.5-7B: 9.2% and 4.2%). We report these because they raise the objection that base models show no collapse only because the constraint never took effect. Section 5 answers this objection on instruction-tuned models, where compliance can be controlled cleanly: the loss persists when compliance is absent (line 3) or automatic (line 2), and nearly disappears when the decoder supplies compliance without the clause (line 1). The comparison also holds at matched compliance: with three constraint-free worked examples to elicit the answer format, base Llama loses 5.0% and base Qwen gains 1.2% with no constraint significant, against −24.4% and −39.5% with all eight significant for the instruction-tuned models, and in the three cells where base compliance matches or exceeds instruct compliance, every base 95% CI contains zero (Appendix B.1). The difference between a base model and its tuned counterpart is how much structured template the clause can disrupt rather than whether the clause is obeyed.

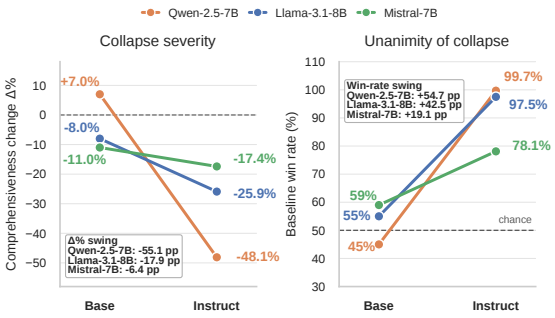

Figure 3: **Base models are at the low-template end.** Δ% (left) and baseline win rate (right), base vs. instruction-tuned, eight constraints, GPT-4o judge. Per-constraint results in Appendix A.1.

6 Mechanism: a Localized, Reversible Planning Decision

This section shows that the clause flips a response-strategy decision that is made before any token is generated and can be reversed at inference time at a single mid-network layer.

6.1 Capability Limit versus Planning Failure

The collapse could reflect either (a) an inability to express complex content under constraints, or (b) selection of a minimal strategy at prompt-encoding time despite having the capability. A two-pass protocol distinguishes them: for 10 prompts × 2 constraints we generate a baseline, a single-pass constrained response, and a two-pass response (generate freely, then rewrite under the constraint). All three models support (b): single-pass retains 49% / 13% / 52% of baseline length on Llama / Qwen / Mistral, two-pass retains **96% / 59% / 91%** (Appendix A.9). Qwen recovers on `no_the` (16%→79%) but only partially on `no_comma` (9%→40%), which indicates a residual capability gap on high-frequency syntactic tokens in addition to the planning failure.

6.2 The Decision Is Encoded Before Generation

If the collapse is a planning failure, the decision should be detectable before generation. For 40 prompts × 3 conditions we extract hidden states at the last prompt token across five layers and train Ridge probes to predict response length. A linear probe explains 66–94% of length variance at

Table 3: **Of the seven depths tested, only layer 15 reverses the collapse.** Steering + mask at seven depths of Qwen-2.5-7B under `no_the`, dose anchored to $\|\mathbf{h}\|$ at each layer, satisfaction 100% ($n = 24$/cell). Placebo: matched-norm random direction at the same layer and dose.

Layer	$\|\mathbf{d}\|/\|\mathbf{h}\|$	Recovery @ f=0.22	Recovery @ f=0.30	Placebo @ f=0.30
2	0.098	−7%	−6%	0%
6	0.216	−15%	−23%	10%
10	0.405	6%	14%	**24%**
15	0.659	**65%**	**79%**	9%
19	0.689	1%	6%	0%
23	0.560	−2%	−6%	−6%
25	0.475	−5%	0%	−2%

∼50% depth. Its R^2 rises sharply in early layers, peaks at mid-depth and then levels off. Grouped splits by base prompt confirm that this is not prompt-identity memorization (R^2 retains about 95%; Appendix I). Models commit to a strategy within the first 1–3 generated tokens (Jensen–Shannon divergence = 0.46–0.54; Appendix K).

Probing R^2 tracks collapse severity across the five instruction-tuned models we probe (Figure 4; $r = 0.928$), from 0.66 for the most robust to 0.93 for the most fragile. Two controls indicate that the probe reflects planning rather than the clause's surface presence. First, the same probes give negative R^2 on base models (Llama −4.04, Qwen −0.59), and a matched 30-sample control rules out sample size as the cause. Second, and more directly, Mistral's `no_bullet` is worded like every other constraint but does not cause collapse. A probe that had learned to detect a clause would predict a short answer there and be wrong on every prompt, yet its out-of-fold error is 11.7%, *lower* than on the unconstrained condition (17.1%) and far below a heavy-collapse condition (44.1%). The probe therefore tracks planned length rather than clause presence (Appendix I.1).

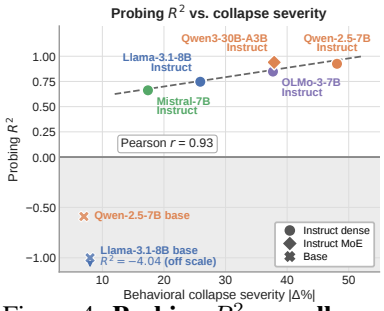

Figure 4: **Probing R^2 vs. collapse severity** ($r = 0.93$). Gray zone: base models (negative R^2). Values are provided in Appendix A.3.

6.3 Reversing the Decision at Inference Time

If a single representation carries the decision, intervening on it should reverse the collapse. **Direction:** $\mathbf{d}$, the mean unconstrained minus the mean constrained prompt representation at the last prompt token, pooled over `no_comma` and `no_the`, with $\hat{\mathbf{u}} = \mathbf{d}/\|\mathbf{d}\|$; directions fitted on disjoint halves agree at cos = 0.88–0.98, and a direction fitted on prompts 12–39 reproduces the full-set direction on held-out prompts 0–11 (Appendix J.1). **Dose:** we add $f \cdot \hat{\mathbf{u}} \cdot \|\mathbf{h}\|$ to the residual stream from the last prompt token onward, where $\|\mathbf{h}\|$ is the mean residual-stream norm at the last prompt token at that layer. Anchoring to $\|\mathbf{h}\|$ is essential: residual norms span an 18× range across depth, so a fixed magnitude confounds layer with dose. **Compliance:** the direction along which the collapse moves is largely the model's representation of the clause itself. It is about 75% aligned with clause *absence* (cosine with a clause-presence direction −0.76 / −0.77 / −0.71), as §5 predicts if the clause is the trigger, and on Qwen the two cannot be separated: removing the clause component leaves a direction no better than a random placebo (Appendix J.2). Steering along it therefore restores 87–101% of lost *length*, but the model then ignores the constraint (satisfaction 0%), so we pair it with a logit mask that holds satisfaction at 100% in every cell below. **The mask alone recovers 0% to −3%**, so all recovery comes from the direction. Reversing the representation of the clause restores the content, and the mask keeps the constraint satisfied.

The decision is localized. Across seven depths, with the dose anchored to $\|\mathbf{h}\|$ at each and satisfaction at 100% everywhere, recovery is **79% at layer 15 and between −23% and +14% at all six other depths**. At layer 10, a matched-norm placebo outperforms the real direction, so layer 15 is the only depth with a direction-specific effect (Table 3). Contrast magnitude peaks at layer 19 rather than 15, so it does not predict where steering works.

Table 4: **Beyond the best dose, responses keep their length but lose content.** Dose–response for Qwen-2.5-7B under `no_comma`, steering + mask, 100% satisfaction in every cell ($n = 60$; "sd" is between-seed s.d. of Δ%). The $f = 0.38$ cell was measured twice in independent runs with independently rebuilt directions (64% and 62%).

Dose f	Words	Length recovery	Compr. Δ%	sd	Compr. recovery
0.26	186	45%	−34.7	1.2	45%
0.30	258	67%	−27.9	1.0	56%
0.34	281	74%	−27.3	2.2	57%
0.38	350	95%	**−22.7**	1.7	**64%**
0.40	361	98%	−27.7	1.0	55%
0.42	363	99%	−29.2	2.1	53%
0.44	351	95%	−33.4	1.8	46%

Recovery. At the best dose, steering + mask recovers **64–82% of lost comprehensiveness on Qwen-2.5-7B at 100% satisfaction** with no training ($n = 60$/cell; `no_the` −45.2%→−8.2%, `no_comma` −62.9%→−22.7%; Appendix A.11). Recovery follows collapse severity and probing R^2: Llama, whose collapse and probe R^2 are both lower, recovers 36–46%.

6.4 Length and Content Diverge

A fine sweep of the dose gives the clearest evidence that the judge is not rewarding length. Comprehensiveness follows an inverted U peaking at $f = 0.38$ while length recovery rises and saturates (Table 4). **Between $f = 0.38$ and $f = 0.44$ the responses have the same length (350 versus 351 words, with length recovery flat at 95–99%), while comprehensiveness recovery falls by 18 points.** The additional steering adds words but not content. Five-gram repetition at the high dose is 0.03, so the text is fluent and a degeneration check cannot distinguish it from the best cell; only a content-sensitive judge can do so. If the judged gap were driven by a preference for length, these two cells could not differ. The same inverted U appears on `no_the`, peaking at $f = 0.30$ (82%). *Optimizing an intervention for response length therefore selects the wrong operating point*: maximizing word count would have chosen $f = 0.42$ (363 words), which recovers 53% of the loss instead of 64%.

7 Deployment Constraints, Evaluation, and Mitigation

7.1 The Constraint Tax in Deployment

The lexical constraints above are controlled diagnostic probes, and no real user asks for an explanation of gradient descent without commas. This is intentional, as in a load test, but the analogy needs to be validated. We test four deployment constraints on all three primary models (40 prompts each, GPT-4o judge): professional tone, no preamble, hedging language, and plain language (Appendix L).

Deployment constraints cost **14–31%** of comprehensiveness overall (−19.4% Llama, −30.6% Qwen, −13.6% Mistral), comparable to the lexical bans, and three of the four place no restriction on factual depth or structure. Hedging (write "may" rather than definitive claims) costs 12–25% on all three models while placing no limit on length, depth or organization. No-preamble restricts *only* the opening tokens yet costs 14–34%, which connects directly to the divergence result above. Professional tone costs 14.4% and 18.0% on Llama and Qwen but shows no significant effect on Mistral (−0.3%, $p = 0.44$), so we report it as an effect on two of the three models. Plain language has the largest effect (−27.1% to −45.4%) but is partly confounded, because avoiding jargon makes some loss intended; excluding it gives −15.8 / −25.7 / −8.7% (Appendix L).

7.2 Independent Evaluation Underestimates the Collapse

Standard independent scoring, in which the judge assesses each response in isolation, greatly underestimates the loss: on Llama, pairwise scoring measures 4.1× the loss that independent scoring detects under the original rubrics (Table 18), and for `no_bullet` independent scoring detects *zero* degradation against 12.9% pairwise. However, the two protocols also used different rubrics, so the gap might come from rubric wording. We separate the factors by crossing them (two protocols × two rubrics, three models, 320 comparisons per cell) and find that **changing the rubric moves the result by at most 2.0 points while changing the protocol moves it by 10–18 points** (Appendix A.10). The

deciding factor is whether the judge has a response to compare against. Under the strictest design, with criteria *and* numeric anchors held identical across protocols, the gap is $4.3\times$ (Mistral), $6.8\times$ (Llama) and $2.0\times$ (Qwen), equal to or larger than with the original rubrics on every model, so the effect remains when the rubric confound is removed by design (Table 18). Without a baseline for comparison, a judge lacks a calibration reference: constrained responses are locally coherent and score well alone, so any evaluation relying only on independent scoring may miss a substantial quality loss (Appendix A.5).

7.3 Mitigation by Fine-Tuning

Section 6.1 showed that models retain the capability, and §6.3 showed that the decision can be reversed without training. To test whether this recovery can be *internalized*, we fine-tune Llama and Qwen with LoRA on 80 verified two-pass examples of `no_comma` and `no_the` (rank 16, α=32, Q/K/V/O, 5 epochs, $\sim$10 min on one H100). On held-out lexical constraints with compliance maintained ($\geq$85%), collapse drops from -24.6% to -8.7% on Llama (65% recovery) and -47.4% to -6.5% on Qwen (86%). A second Llama adapter with the same settings, trained on only 56 constraint-satisfying examples of the same two bans, transfers to the four deployment constraints, which it never saw and which are not token bans (-19.6% to -4.4%, 78% recovery; Appendix M). We claim no novelty over supervised fine-tuning; the contribution is that constraint robustness is a learnable skill that current training fails to teach.

8 Discussion, Limitations, and Conclusions

Our results suggest the helpfulness of instruction-tuned models is partly an artifact of learning a narrow distribution of response templates rather than a generalizable competence. A constraint clause moves the model into a different planning state rather than making generation harder, and the cost scales with how much of the default answer depended on the disrupted template. The evaluation gap and deployment results suggest the field may be overestimating the usefulness of unmodified instruction-tuned models in constrained production settings: requesting a professional tone or suppressing a conversational opener costs real content that independent scoring does not detect. The localization result suggests a concrete target: if the decision is carried at one mid-network layer, constraint robustness may be trainable by regularizing that representation rather than by scaling models, which does not help, or by enumerating constraints, which does not cover all possible constraints (Appendix N).

Limitations. Steering and the two-pass and divergence analyses use three 7–8B open-weight models, and probing uses five; the behavioral effect holds across all seven.

Conclusions. Trivial lexical constraints cost 17–48% of comprehensiveness across seven instruction-tuned models, and the loss is in content rather than presentation. It is triggered by the constraint clause's presence in the prompt rather than the difficulty of obeying it, as four converging tests show. The decision is made before generation and can be reversed at one mid-network layer, at 100% constraint satisfaction and with no training, but not at the six other depths tested. We recommend pairwise evaluation for constrained generation, and constraint robustness as an explicit training objective.

REPRODUCIBILITY STATEMENT

All experimental details required to reproduce our results are provided in the paper and appendix. The 40 evaluation prompts are listed verbatim in Appendix O and the exact text of every constraint in Appendices P and L. All judge prompts, including system prompts, user templates and scoring anchors, are reproduced verbatim in Appendix Q, and the human-evaluation rubric in Appendix S. Generation settings (temperature 0.7, top-p 0.9, max 1024 tokens, three samples per prompt–constraint pair, bfloat16), LoRA hyperparameters (rank 16, $\alpha = 32$, Q/K/V/O projections, 5 epochs), probing settings (Ridge regression, 5-fold and grouped cross-validation, five evenly spaced layers, last prompt token) and steering settings (difference-of-means direction, norm-anchored dose, all positions from the last prompt token onward, logit-masked decoding, three fixed seeds) are given in §3, §6 and Appendices I–M. All models except GPT-4o-mini are open-weight and identified by exact checkpoint. Every table reports the number of prompts, samples and comparisons behind it. Source code for generation, judging, probing, steering and analysis, together with the raw result files underlying every table, will be released upon acceptance.

ETHICS STATEMENT

This work evaluates publicly available language models on neutral, non-sensitive informational prompts and does not involve personal data, sensitive attributes, or deployment of models to users. The one component involving human participants is the blinded evaluation described in Appendix S: participation was anonymous and voluntary, all evaluators gave informed consent and were told their ratings would be used to assess AI-generated response quality for research, no personally identifying information was collected, and evaluators were compensated for their time.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: generating a synthetic data set (the LoRA training examples of §7.3 are two-pass responses written by the models under study), reformatting data (GPT-4o rewrote the unconstrained baselines as unformatted prose for the format-equalized reference of Appendix G.2), and providing feedback on the research methodology and experiments. We have not used generative AI tools to propose or refine hypotheses, develop conceptual frameworks, design experiments, or interpret results, and the remaining tasks with required disclosure (formulating mathematical claims, contributing to or writing proofs, translation, and qualitative or thematic data analysis) are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure: finding related work, drafting and editing parts of the paper, and checking and formatting references. Each use is described below.

As objects of study. All models listed in §3 are the subject of the experiments.

As evaluation instruments. GPT-4o-mini, GPT-4o and Claude Sonnet 4.6 serve as automated judges for pairwise and independent scoring; GPT-4o additionally performs atomic-claim extraction and claim matching. These uses are methodological contributions of the paper and are described in §3 and Appendices Q and R, with all prompts reproduced verbatim. Their reliability is assessed directly: cross-judge agreement in Appendix A.4, judge-reseeding variance in Appendix A.2, and validation against 10 human evaluators in §4.3.

As research tools. All experimental designs, hypotheses, and decisions about which analyses to run and how to interpret them were made by the authors. All numerical results reported in this paper were produced by executing code on the stated models and data; no result, table entry or figure value was generated by an LLM as text.

In preparing the manuscript. The authors used an LLM assistant for drafting and editing assistance. The authors reviewed all AI-assisted work, including data and text, and take full responsibility for the content of this paper, including text produced with the aid of generative AI.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Haikang Deng, Po-Nien Kung, and Nanyun Peng. Decoupling task-solving and output formatting in LLM generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16764–16781, 2026.

Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. Self-play with execution feedback: Improving instruction-following capabilities of large language models. In *International Conference on Learning Representations*, pp. 39286–39313, 2025.

Jianshuo Dong, Yutong Zhang, Yan Liu, Zhenyu Zhong, Tao Wei, Chao Zhang, and Han Qiu. Revisiting the reliability of language models in instruction-following. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7784–7812, 2026.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. Length-controlled AlpacaEval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*, 2024.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10932–10952, 2023.

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

Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10864–10882, 2024a.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, et al. Multi-IF: Benchmarking LLMs on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*, 2024b.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Ivan Yee Lee, Loris D'Antoni, and Taylor Berg-Kirkpatrick. The Format Tax. *arXiv preprint arXiv:2604.03616*, 2026.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D'Antoni. Grammar-aligned decoding. *Advances in Neural Information Processing Systems*, 37:24547–24568, 2024.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Amy Xin, Youfeng Liu, Bin Xu, Lei Hou, and Juanzi Li. AGENTIF: Benchmarking large language models instruction following ability in agentic scenarios. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, pp. 25055–25083, 2024.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. In *International Conference on Learning Representations*, pp. 55790–55823, 2025.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let me speak freely? A study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1218–1236, 2024.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, et al. Benchmarking complex instruction-following with multiple constraints composition. *Advances in Neural Information Processing Systems*, 37:137610–137645, 2024.

Brandon T Willard and Rémi Louf. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

Weizhe Yuan, Ilia Kulikov, Ping Yu, Kyunghyun Cho, Sainbayar Sukhbaatar, Jason E Weston, and Jing Xu. Following length constraints in instructions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24232–24243, 2025.

Xinghua Zhang, Haiyang Yu, Cheng Fu, Fei Huang, and Yongbin Li. IOPO: Empowering LLMs with complex instruction following via input-output preference optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22185–22200, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

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

A Detailed Numerical Results

This appendix collects the complete numerical results underlying the main-text figures and summary tables.

A.1 Full Per-Constraint Pairwise Results

Tables 5–8 give the per-constraint results behind Figures 2 and 3.

Table 5: Per-constraint pairwise results: **Llama-3.1-8B-Instruct** (40 pairs per constraint).

	GPT-4o-mini Judge				GPT-4o Judge			
Constraint	Base $\bar{c}$	Const $\bar{c}$	Δ%	Win%	Base $\bar{c}$	Const $\bar{c}$	Δ%	Win%
No comma	9.25	6.75	−27.0	100	8.95	6.17	−31.0	100
No colon	9.22	6.75	−26.8	100	8.93	6.40	−28.3	100
No semicolon	9.22	6.97	−24.4	98	8.90	6.62	−25.6	98
No bullet	9.07	7.90	−12.9	85	8.78	7.30	−16.8	88
No “the”	9.28	7.17	−22.6	100	8.88	6.72	−24.2	98
No disc. mkrs	9.28	7.50	−19.1	98	8.90	6.90	−22.5	98
No comma+colon	9.22	6.47	−29.8	100	8.90	6.08	−31.7	100
No comma+bullet	9.25	7.03	−24.1	100	8.93	6.55	−26.6	100
Overall	**9.22**	**7.07**	**−23.4**	**97.5**	**8.89**	**6.59**	**−25.9**	**97.5**

Table 6: Per-constraint pairwise results: **Qwen-2.5-7B-Instruct** (40 pairs per constraint).

	GPT-4o-mini Judge				GPT-4o Judge			
Constraint	Base $\bar{c}$	Const $\bar{c}$	Δ%	Win%	Base $\bar{c}$	Const $\bar{c}$	Δ%	Win%
No comma	9.07	3.55	−60.9	100	8.97	2.60	−71.0	100
No colon	9.20	5.65	−38.6	100	8.97	5.00	−44.3	100
No semicolon	9.18	6.03	−34.3	100	8.97	5.25	−41.5	100
No bullet	9.07	7.97	−12.1	88	8.95	7.30	−18.4	98
No “the”	9.20	5.33	−42.1	100	8.97	4.42	−50.7	100
No disc. mkrs	9.15	6.22	−32.0	100	8.97	5.70	−36.5	100
No comma+colon	9.05	3.58	−60.5	100	9.00	2.60	−71.1	100
No comma+bullet	9.03	5.53	−38.8	100	8.97	4.40	−51.0	100
Overall	**9.12**	**5.48**	**−39.9**	**98.4**	**8.97**	**4.66**	**−48.1**	**99.7**

Table 7: Per-constraint pairwise results: **Mistral-7B-Instruct-v0.3** (40 pairs per constraint).

	GPT-4o-mini Judge				GPT-4o Judge			
Constraint	Base $\bar{c}$	Const $\bar{c}$	Δ%	Win%	Base $\bar{c}$	Const $\bar{c}$	Δ%	Win%
No comma	8.70	7.05	−19.0	83	8.40	6.42	−23.5	85
No colon	8.75	7.20	−17.7	88	8.22	6.83	−17.0	80
No semicolon	8.72	7.62	−12.6	83	8.20	7.15	−12.8	75
No bullet	8.15	8.55	+4.9	38	8.05	7.83	−2.8	55
No “the”	8.78	7.12	−18.8	88	8.35	6.65	−20.4	88
No disc. mkrs	8.80	7.55	−14.2	85	8.30	6.97	−16.0	80
No comma+colon	8.90	6.12	−31.2	98	8.47	5.50	−35.1	90
No comma+bullet	8.53	8.25	−3.2	58	8.28	7.38	−10.9	73
Overall	**8.67**	**7.43**	**−14.2**	**77.2**	**8.28**	**6.84**	**−17.4**	**78.1**

A.2 Statistical Significance and Judge Variance

Judge variance. Re-running the judge three times under different random seeds shifts each per-constraint value by at most 0.8 points on Llama and 1.5 on Mistral. Every effect in Table 9 is therefore at least ten times larger than the judge’s own run-to-run variability.

Table 8: Per-constraint pairwise results: **Base models** (GPT-4o judge, 40 pairs per constraint). **Bold** $\Delta\%$ and Win% values indicate that the constrained response was rated higher than the unconstrained baseline. Base constraint satisfaction is low (§5.3); this table characterizes behavior under the appended clause, and the causal claims of §5 do not rest on it.

	Llama-3.1-8B (Base)				**Qwen-2.5-7B (Base)**				**Mistral-7B (Base)**			
Constraint	B	C	$\Delta\%$	W%	B	C	$\Delta\%$	W%	B	C	$\Delta\%$	W%
No comma	6.50	5.33	-18	62	5.83	5.72	-2	55	5.97	5.47	-8	58
No colon	6.40	5.85	-9	58	5.90	6.15	$\mathbf{+4}$	**48**	5.92	5.30	-11	58
No semicolon	6.55	5.22	-20	75	5.88	6.28	$\mathbf{+7}$	**42**	6.00	5.38	-10	60
No bullet	6.17	6.90	$\mathbf{+12}$	**38**	5.70	6.62	$\mathbf{+16}$	**35**	5.88	5.55	-6	50
No "the"	6.53	5.05	-23	65	5.83	6.10	$\mathbf{+5}$	**48**	6.20	4.72	-24	70
No disc. mkrs	6.22	6.05	-3	50	6.00	5.65	-6	60	5.92	5.97	$\mathbf{+1}$	**50**
No cm+colon	6.45	5.55	-14	62	5.88	6.22	$\mathbf{+6}$	**48**	6.05	5.12	-15	62
No cm+bullet	6.05	7.10	$\mathbf{+17}$	**30**	5.53	7.08	$\mathbf{+28}$	**25**	6.10	5.12	-16	62
Overall	*6.36*	*5.88*	*−8*	*55*	*5.82*	*6.23*	***+7***	***45***	*6.01*	*5.33*	*−11*	*59*

Table 9: **Per-constraint confidence intervals** on Llama-3.1-8B-Instruct (second generation run, GPT-4o-mini judge, 40 prompts per constraint). Every constraint remains significant after multiple-comparison correction at $p < 10^{-7}$, and no interval approaches zero. For six of eight constraints the unconstrained answer was judged more comprehensive on all 40 prompts.

Constraint	Comprehensiveness loss	95% CI	Unconstrained judged better on
No comma+colon	-31.9%	$[-34.3, -29.6]$	100% of prompts
No colon	-27.6%	$[-29.8, -25.4]$	100%
No comma	-27.2%	$[-29.2, -25.4]$	100%
No comma+bullet	-24.9%	$[-26.8, -22.9]$	100%
No "the"	-24.8%	$[-27.6, -22.1]$	100%
No semicolon	-24.5%	$[-26.6, -22.5]$	100%
No disc. mkrs	-21.1%	$[-23.8, -18.4]$	98%
No bullet	-15.5%	$[-18.0, -12.7]$	92%

A.3 Probing R^2 vs. Collapse Severity: Numerical Values

Table 10 gives the values behind Figure 4.

Table 10: **Probing R^2 tracks collapse severity, and base models show no comparable predictability.** The more severe the behavioral collapse, the more predictable the response length from prompt representations ($r = 0.928$ across the five instruction-tuned models). Base models yield negative R^2 at every layer. Values are peak-layer Ridge R^2; per-layer values are in Appendix I.

Model	Family	Collapse ($\Delta\%$)	Probing R^2 (best)
Mistral-7B-Instruct	Mistral	-17.4%	0.662
Llama-3.1-8B-Instruct	Llama	-25.9%	0.747
OLMo-3-7B-Instruct	OLMo	-37.7%	0.848
Qwen3-30B-A3B-Instruct	Qwen 3	-37.9%	0.940
Qwen-2.5-7B-Instruct	Qwen 2.5	-48.1%	0.925
Llama-3.1-8B (base)	Llama	-8.0%	-4.04
Qwen-2.5-7B (base)	Qwen	$+7.0\%$	-0.59

A.4 Cross-Judge and Cross-Family Validation

Table 11 compares the three judges on the three primary models.

A.5 Independent vs. Pairwise Evaluation

In independent evaluation the judge assesses each response in isolation against an implicit quality standard. Constrained responses, while shorter and less comprehensive, are often *locally coherent*: each sentence is accurate and well-formed. Without seeing the full baseline response, the judge lacks

Table 11: **Cross-judge and cross-family validation.** Three judges from two model families consistently detect the collapse with identical severity ordering. Claude Sonnet 4.6, despite assigning lower baseline scores (7.1–8.2 vs. 8.7–9.2 for GPT-4o-mini), detects comparable or larger degradation, which rules out a GPT-family judge bias.

	GPT-4o-mini		**GPT-4o**		**Claude Sonnet 4.6**	
Model	$\Delta\%$	Win%	$\Delta\%$	Win%	$\Delta\%$	Win%
Mistral-7B-Instruct	-14.2	77.2	-17.4	78.1	-18.2	71.9
Llama-3.1-8B-Instruct	-23.4	97.5	-25.9	97.5	-28.4	96.9
Qwen-2.5-7B-Instruct	-39.9	98.4	-48.1	99.7	-48.9	99.7

Table 12: **Independent vs. pairwise evaluation** on Llama-3.1-8B-Instruct (GPT-4o-mini judge). Independent scoring detects $< 1/5$ of the quality loss measured by pairwise comparison.

Constraint	Independent Judge $\Delta\%$	Pairwise Judge $\Delta\%$
No comma	-5.4	-27.0
No colon	-4.5	-26.8
No semicolon	-3.2	-24.4
No bullet/lists	-0.0	-12.9
No "the"	-2.8	-22.6
No disc. markers	-0.5	-19.1
No comma+colon	-7.0	-29.8
No comma+bullet	-4.6	-24.1
Average	$\mathbf{-3.5}$	$\mathbf{-23.4}$

a calibration reference and assigns inflated scores. The baseline composite score (8.54/10) and the constrained composite (7.94–8.54/10) both fall in the "good to very good" range, which hides the large gap in actual comprehensiveness. §7.2 shows that this is a property of the protocol rather than of the rubric wording.

A.6 Human Evaluation: Numerical Values

Table 13: **Human evaluation results** (numerical values for Figure 5). Ten blinded evaluators rate responses on six criteria (1–10). Content-quantity criteria drop 1.5–2.3× more than surface criteria. Correctness is excluded from the content-quantity average because it measures accuracy of surviving content rather than amount produced. 320 pairs per model. Wilcoxon signed-rank: $p < 0.001$ for all criteria on all models.

		Llama-8B	**Mistral-7B**	**Qwen-7B**
	Criterion	$\Delta\%$	$\Delta\%$	$\Delta\%$
Content	Semantic coverage	-22.5	-14.3	-41.4
Content	Comprehensiveness	-27.0	-17.4	-46.9
Content	Helpfulness	-27.4	-17.1	-43.7
	Correctness	-11.2	-6.0	-21.8
Surf	Verbosity (inv.)	-14.7	-5.5	-27.3
Surf	Readability	-15.1	-8.5	-31.5
Content-qty avg		$\mathbf{-25.6}$	$\mathbf{-16.3}$	$\mathbf{-44.0}$
Surface criteria avg		$\mathbf{-14.9}$	$\mathbf{-7.0}$	$\mathbf{-29.4}$
Content-qty / Surface ratio		$\mathbf{1.7}\times$	$\mathbf{2.3}\times$	$\mathbf{1.5}\times$
Baseline win% (coverage)		97%	78%	99%
Position bias (avg)		$+0.22$	$+0.21$	$+0.16$

Extended statistical details. We exclude correctness from the content-quantity average because it measures the *accuracy* of surviving content rather than the *amount* produced; the collapse manifests as omission, not error, and correctness drops less precisely because what remains is largely accurate. Human-rated helpfulness and comprehensiveness change by nearly identical amounts (-27.4% vs. -27.0% on Llama, 0.4pp apart), which validates comprehensiveness as the primary metric. Residual

position bias after randomization averages +0.19 across all criteria and models; the largest per-criterion bias is on helpfulness (+0.53 Llama, +0.59 Mistral), small relative to observed effect sizes of 1.0–4.1 points.

Qwen's surface criteria. Qwen's surface drop of 29.4% needs an explanation. Unlike Llama and Mistral, where surface quality is relatively preserved, Qwen's collapse is severe enough that *everything* degrades. The content/surface ratio for Qwen ($1.5\times$) is the weakest of the three, but this does not support a presentation account for Qwen. Instead, it indicates that Qwen's generation failure goes beyond a loss of content: when a response shrinks from 448 to 113 words, its prose quality also suffers.

Table 14: **Per-constraint human evaluation results.** Comprehensiveness $\Delta\%$ and baseline win% by constraint and model. Mistral's `no_bullet` result (positive $\Delta\%$, below-chance win rate) confirms evaluators discriminate real quality differences rather than preferring longer text. 40 pairs per cell.

	Llama-8B		**Mistral-7B**		**Qwen-7B**	
Constraint	Comp $\Delta\%$	Win%	Comp $\Delta\%$	Win%	Comp $\Delta\%$	Win%
No bullet	-12.0	85	$+3.3$	45	-14.0	90
No colon	-32.0	100	-19.3	88	-45.7	100
No comma	-32.2	100	-23.4	82	-66.7	100
No semicolon	-28.9	98	-14.7	82	-42.3	100
No "the"	-25.3	100	-22.3	85	-49.5	100
No disc. mkrs	-21.1	92	-16.0	80	-39.2	100
No cm+colon	-34.7	100	-38.2	98	-67.3	100
No cm+bullet	-29.0	100	-6.1	62	-49.1	100
Overall	$\mathbf{-27.0}$	**97**	$\mathbf{-17.4}$	**78**	$\mathbf{-46.9}$	**99**

Formatting vs. content constraints. Comparing information loss from *formatting* constraints (no_bullet, no_colon, no_semicolon, no_comma_colon, no_comma_bullet) with *content-word* constraints (no_the, no_discourse), averaged over coverage and comprehensiveness drops: for Mistral, content-word constraints cause *more* loss than formatting constraints (17.9% vs. 13.8%); for Llama the pattern reverses modestly (24.9% vs. 21.7%); Qwen shows near-parity (39.6% vs. 41.2%). The Mistral result is informative because Mistral is the most robust model overall: even there, banning a common content word disrupts generation more than banning structural tokens, consistent with the frequency gradient of §5.2.

A.7 All Seven Instruction-Tuned Models

Table 15 gives the overall collapse for every instruction-tuned model we evaluate.

Table 15: **Collapse across seven instruction-tuned models** (GPT-4o pairwise judge, 320 pairs per model). Every model collapses regardless of family, scale, architecture or training openness, and larger models collapse *more* (Llama 8B $\rightarrow$ 70B: $-25.9\% \rightarrow -34.7\%$).

Model	Family	Size	$\Delta\%$	Win%
Mistral-7B-Instruct-v0.3	Mistral	7B	-17.4	78.1
Llama-3.1-8B-Instruct	Llama	8B	-25.9	97.5
GPT-4o-mini	GPT	closed	-31.0	99.1
Llama-3.3-70B-Instruct	Llama	70B	-34.7	99.4
OLMo-3-7B-Instruct	OLMo	7B	-37.7	95.9
Qwen3-30B-A3B-Instruct	Qwen 3 (MoE)	30B (3B act.)	-37.9	99.1
Qwen-2.5-7B-Instruct	Qwen 2.5	7B	-48.1	99.7

A.8 Banned-Token Frequency Sweep

Table 16 gives the band averages behind §5.2.

Table 16: **Collapse severity scales with how often the banned word would have been used.** Fifteen single-word bans grouped by expected frequency in an unconstrained answer; each cell averages the words listed in its row (GPT-4o-mini judge, 600 comparisons per model). The ordering holds in all three models without exception. The top row is the more informative one: these bans require no effort to satisfy, yet they still cost roughly half of what banning "of" costs.

Expected frequency of the banned word	Llama-3.1-8B	Qwen-2.5-7B	Mistral-7B
Essentially never (`zephyr`, `quixotic`, `bezoar`, `sassafras`)	−12.2	−21.2	−10.1
Rarely (`utilize`, `nonetheless`, `paradigm`)	−13.7	−23.0	−11.4
Sometimes (`which`, `because`, `important`, `often`)	−17.9	−32.1	−14.1
Constantly (`of`, `and`, `is`, `a`)	−22.8	−40.9	−19.2

A.9 Two-Pass Recovery: Numerical Values

Table 17 gives the per-constraint length retention behind §6.1.

Table 17: **Two-pass recovery experiment.** Generating freely and then rewriting under the constraint substantially recovers response length: the model *can* produce comprehensive constrained output but does not plan to do so in a single pass. Retention is measured relative to the unconstrained baseline word count. 10 prompts × 2 constraints per model.

	Single-Pass Retention			**Two-Pass Retention**		
Constraint	Llama	Qwen	Mistral	Llama	Qwen	Mistral
No comma	51%	9%	46%	100%	40%	86%
No "the"	46%	16%	57%	92%	79%	95%
Overall	**49%**	**13%**	**52%**	**96%**	**59%**	**91%**

A.10 Protocol versus Rubric

Table 18: **The evaluation gap is a property of the protocol rather than the rubric.** Effect on measured $\Delta\%$ of changing the judging protocol (side-by-side vs. one-at-a-time) with the rubric held fixed, against the effect of changing the rubric with the protocol held fixed (320 comparisons per cell). The last two columns give the ratio of pairwise-measured to independently-measured loss, first with the original rubrics and then with criteria and numeric anchors held identical across the two protocols.

Model	Protocol effect	Rubric effect	Pairwise / independent	Identical rubric
Mistral-7B-Instruct	−10.4 pp	−0.7 pp	3.5×	4.3×
Llama-3.1-8B-Instruct	−18.1 pp	−0.7 pp	4.1×	6.8×
Qwen-2.5-7B-Instruct	−18.1 pp	−2.0 pp	2.0×	2.0×

Changing the rubric moves the measured loss by at most 2.0 points while changing the protocol moves it by 10–18 points, on all three models.

A.11 Steering Recovery at the Best Dose

Table 19 gives the confirmatory run behind the recovery figures of §6.3.

A.12 Human Evaluation Figure

Figure 5 plots the values of Table 13.

Table 19: **Comprehensiveness recovery at the best dose, confirmatory run.** 20 prompts × 3 seeds ($n = 60$ per cell), judged comprehensiveness, constraint satisfaction 100% and 5-gram repetition ≤ 0.05 in every cell. "Placebo" is a matched-norm random direction at the same dose. Between-seed standard deviation is 0.7–4.1 points. Steering recovery tracks collapse severity and probing R^2 across models (Mistral ∼25% and within its own noise; see Appendix J.3).

Model	Constraint	Unsteered Δ%	Dose f	Steered Δ%	Recovery	Placebo
Qwen-2.5-7B	No "the"	−45.2	0.30	**−8.2**	**82%**	13%
Qwen-2.5-7B	No comma	−62.9	0.38	**−22.7**	**64%**	24%
Llama-3.1-8B	No semicolon	−26.7	0.26	−14.5	46%	−8%
Llama-3.1-8B	No comma	−31.4	0.30	−20.1	36%	−8%

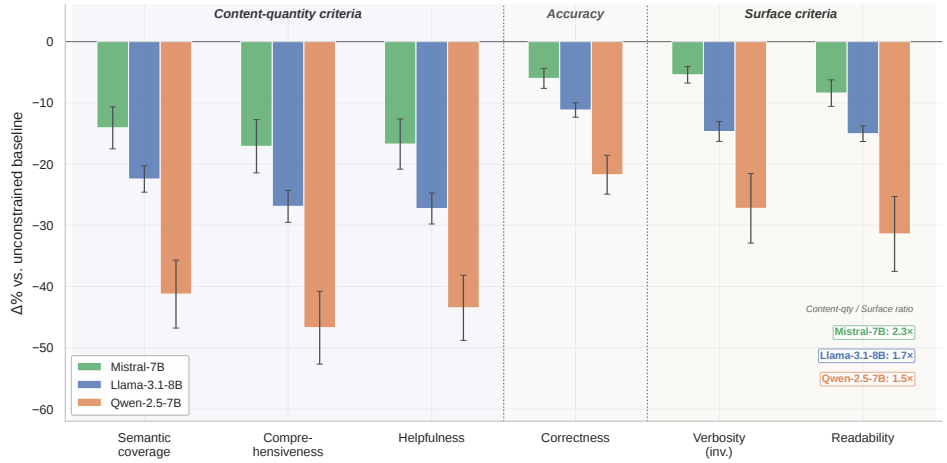

Figure 5: **Human evaluation results.** Ten blinded evaluators, six criteria (1–10), 320 pairs per model. Dashed line separates content-quantity criteria (left) from surface criteria (right). Numerical values in Table 13.

B Constraint Satisfaction Rates

Interpretation. Llama and Qwen satisfy the constraints at high rates, so for those models the collapse demonstrably occurs among responses that successfully follow the constraint. Mistral is high on `no_semicolon`, `no_discourse` and `no_bullet` but very low on the comma and "the" bans. As §5 establishes, this does not weaken the result: Mistral collapses by −24.5% on `no_comma` despite zero of 40 responses satisfying it, which is one of the four converging tests showing that the clause rather than the compliance drives the loss. Compliance is still a precondition wherever a claim *requires* it, which is why the LoRA analysis of Appendix M filters at ≥85% satisfaction.

Note on colon satisfaction for Llama. Llama-3.1-8B-Instruct achieves only 65.8% on the no-colon constraint because it frequently generates headers with colons ("Step 1: . . . ") as part of its learned formatting templates. This is itself evidence of template dependence.

B.1 Base Models at Matched Compliance

Each base-model prompt is preceded by three worked question–answer examples on unrelated topics, none carrying a constraint; decoding is greedy. Table 21 restricts the comparison to cells where base compliance matches or exceeds instruct compliance.

C Category Consistency

Technical prompts show the largest decline on Llama (−26.7%), consistent with the expectation that structured, detail-rich content depends most on the formatting patterns disrupted by lexical constraints.

Table 20: Constraint satisfaction rates (%) across the three primary instruction-tuned models. Measured on 120 responses per cell (40 prompts × 3 samples) by automated checkers.

Constraint	Llama-3.1-8B-Inst.	Qwen-2.5-7B-Inst.	Mistral-7B-Inst.
No comma	99.2	97.5	2
No colon	65.8	96.7	19
No semicolon	100.0	83.3	98
No bullet/lists	100.0	100.0	82
No "the"	92.5	99.2	8
No disc. markers	98.3	91.7	86
No comma+colon	98.3	96.7	7
No comma+bullet	95.8	100.0	0
Average	**93.7**	**95.6**	**37.8**

Table 21: **Base versus instruction-tuned models at matched compliance** ($n = 40$ per cell; 95% CI for the base loss).

Constraint	Model	Base compl.	Instruct compl.	Base Δ%	Instruct Δ%
`no_semicolon`	Llama	100%	100%	−8.7 [−16.3, 0.0]	−24.4
`no_semicolon`	Qwen	100%	85%	−1.7 [−9.3, +6.3]	−35.0
`no_colon`	Llama	75%	72.5%	−7.4 [−17.5, +3.9]	−28.0

D GPT-4o-mini Per-Constraint Breakdown

Table 23 breaks down the GPT-4o-mini result of §4.1 by constraint.

E MT-Bench Validation

To verify that our findings generalize beyond our evaluation prompts, we replicate the experiment on MT-Bench (Zheng et al., 2023), a standard 80-question benchmark spanning eight categories. We test Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct under three constraints (`no_comma`, `no_the`, `no_bullet`), with GPT-4o as judge (240 pairwise comparisons per model).

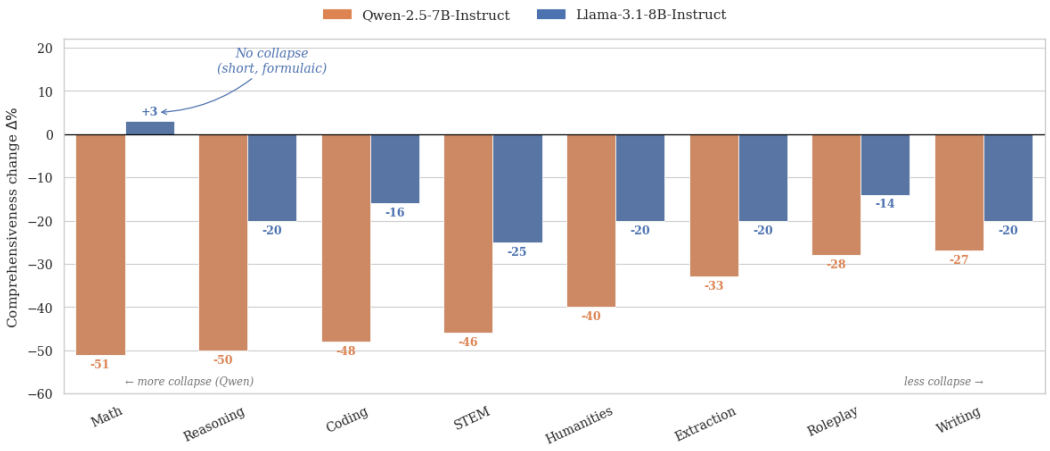

Figure 6: **Comprehensiveness change on MT-Bench by category** (GPT-4o pairwise judge). The collapse is consistent across all eight categories for both models. Llama math (+3%) is the sole exception: its short, formulaic math responses do not rely on the formatting templates that collapse under constraints, which is consistent with the template-dependence gradient of §5.2.

The results closely replicate our main findings. Qwen shows −40.5% overall (89% baseline win rate), nearly identical to the −39.9% on our evaluation set. Llama shows −17.2% (74% win rate), consistent with its −23.4%. Per-constraint patterns are preserved: comma bans are most damaging (Llama −26%, Qwen −65%), followed by "the" (Llama −19%, Qwen −42%) and bullet bans (Llama

Table 22: **Comprehensiveness change by prompt category** (GPT-4o-mini judge). The collapse is consistent across all four categories for all three instruction-tuned models, with no category showing positive change.

Category	Llama Inst. $\Delta\%$	Qwen Inst. $\Delta\%$	Mistral Inst. $\Delta\%$
Explanation	-23.9	-41.3	-18.0
How-to	-22.2	-41.5	-7.7
Analysis	-20.5	-36.6	-15.7
Technical	-26.7	-40.2	-15.3

Table 23: **GPT-4o-mini (closed-weight) per-constraint results** (GPT-4o pairwise judge, 40 prompts per constraint). Comma bans are most damaging (-42%), and average response length drops from 472 to 216 words (-54%). The per-constraint pattern mirrors the open-weight models.

Constraint	Baseline $\bar{c}$	Constr. $\bar{c}$	$\Delta\%$	Win%
No comma	8.97	5.20	-42.1	100
No colon	8.97	6.25	-30.4	100
No semicolon	8.97	6.05	-32.6	100
No bullet/lists	8.90	7.53	-15.4	92
No "the"	8.97	6.10	-32.0	100
No disc. mkrs	9.00	6.55	-27.2	100
No comma+colon	8.97	5.28	-41.2	100
No comma+bullet	8.97	6.58	-26.7	100
Overall	**8.97**	**6.19**	$\mathbf{-31.0}$	**99.1**

-5%, Qwen -12%). The category breakdown shows the collapse spans all task types: STEM (-25% to -46%), reasoning (-20% to -50%) and coding (-16% to -48%) all degrade substantially.

F Large-Scale Replication (400 Prompts)

To verify that our findings are not specific to the 40-prompt evaluation set, we replicate the core experiment on 400 instructions from AlpacaEval (Dubois et al., 2024) (eval split, first 400 examples). All other settings are identical.

Table 24: **Large-scale replication on 400 AlpacaEval prompts.** Comprehensiveness $\Delta\%$ vs. unconstrained baseline (GPT-4o-mini pairwise judge, 400 pairs per constraint).

	Llama-3.1-8B-Instruct			**Qwen-2.5-7B-Instruct**		
Constraint	N	$\Delta\%$	Win%	N	$\Delta\%$	Win%
No comma	400	-28.3	92	400	-62.1	99
No colon	400	-26.7	91	400	-38.0	96
No semicolon	400	-22.6	89	400	-33.1	95
No bullet	400	-4.4	69	400	-1.8	57
No "the"	400	-19.7	87	400	-43.2	98
No disc. mkrs	400	-13.6	79	400	-29.6	93
No cm+colon	400	-31.8	94	400	-62.1	99
No cm+bullet	400	-18.1	85	400	-36.3	95
Overall	**3200**	$\mathbf{-20.8}$	**86**	**3200**	$\mathbf{-38.6}$	**91**

All findings replicate: the effect is large and consistent, the severity ordering is preserved, per-constraint patterns match, and the compositional floor is reproduced. The slightly lower overall effect sizes compared to the 40-prompt set are consistent with the greater diversity of AlpacaEval instructions, some of which are less susceptible to template disruption. By prompt source: helpful_base (-24.1%/-42.8%), koala (-18.9%/-35.4%) and oasst (-19.4%/-38.1%) for Llama/Qwen, with the most instruction-like prompts showing the largest effect.

G Robustness Controls on the Evaluation Setup

G.1 Chain-of-Thought and Reasoning Models

We test two forms of explicit test-time planning to see whether either removes the collapse.

Table 25: **Chain-of-thought prompting reduces but does not remove the collapse.** The model is prompted to plan its key points before writing, under the same three constraints (`no_comma`, `no_the`, `no_bullet`; GPT-4o-mini judge).

Model	Standard	+ CoT
Llama-3.1-8B-Instruct	−22.5%	−18.9%
Qwen-2.5-7B-Instruct	−37.2%	−30.7%
Mistral-7B-Instruct-v0.3	−14.1%	−8.0%

Chain-of-thought (CoT) prompting reduces the collapse by 16–43% but never removes it, which is what the planning account predicts: explicit planning partly compensates for the allocation that the model fails to perform but does not replace it.

Table 26: **A native reasoning phase neither protects against the collapse nor worsens it.** Qwen3-30B-A3B evaluated with its thinking phase enabled and disabled. The unconstrained answer was preferred on 100% of prompts in both arms, and constraint satisfaction in the disabled arm is 100% on all three constraints.

Constraint	Thinking enabled	Thinking disabled
No bullet	−24.5%	−25.6%
No comma	−42.1%	−42.7%
No “the”	−44.0%	−44.5%
Overall	**−36.8%**	**−37.6%**

The two settings differ by 0.8 points overall. A model's own explicit reasoning phase does not protect it, which is consistent with the CoT comparison above: the collapse is decided at prompt-encoding time, and a scratchpad that precedes the answer does not by itself restore the allocation.

G.2 Format-Equalized Reference

To rule out the possibility that the unconstrained baseline wins because of its formatting rather than its content, we rewrote every unconstrained baseline into flowing prose with all bullets, headers and bold markup removed, and used that un-formatted text as the reference instead.

Table 27: **The collapse survives format equalization.** Comprehensiveness $\Delta\%$ against the original baseline and against a format-stripped baseline. The judge's own preference between formatted and un-formatted baselines is small (−5.1% Llama, −6.9% Qwen, +6.3% Mistral, where prose won). Both columns cover the same three constraints (`no_comma`, `no_the`, `no_semicolon`).

Model	Original reference	Format-equalized reference
Llama-3.1-8B-Instruct	−25.5%	−23.3%
Qwen-2.5-7B-Instruct	−45.6%	−45.0%
Mistral-7B-Instruct-v0.3	−20.1%	−21.2%

G.3 Greedy Decoding

Under greedy decoding, Llama and Qwen satisfy `no_comma` and `no_the` in 92.5–100% of responses, so the collapse holds at near-complete compliance.

G.4 Asking the Model to Stay Thorough

We test two ablations that give the model additional help, both on Llama-3.1-8B-Instruct. Every row is measured against the same reference: the model's plain unconstrained answer to the same question, carrying neither a constraint nor any added instruction.

Table 28: **The effect is not a sampling artifact.** Comprehensiveness $\Delta\%$ under the main sampling configuration and under greedy decoding.

Model	Sampled ($T = 0.7$, top-p 0.9)	Greedy
Llama-3.1-8B-Instruct	-24.7%	-24.4%
Qwen-2.5-7B-Instruct	-39.6%	-39.5%
Mistral-7B-Instruct-v0.3	-15.6%	-16.8%

Table 29: **Neither ablation repairs the collapse.** Both leave between half and two thirds of the loss in place.

What the model was asked to do	No comma	No "the"
Answer under the constraint (as in §4)	-27.2%	-24.8%
Same, plus "keep your answer equally thorough"	-14.3%	-15.7%
Plan the answer first, then write it, with no baseline shown	-13.8%	-17.5%

Both mitigations also require a user who already knows the failure exists. Instruction-tuned models are trained to be helpful, and thoroughness is a large part of that; a user who sees this behavior for the first time has no reason to append "keep your answer equally thorough." One ordinary formatting request is enough to remove the helpfulness, which is the failure this paper reports.

G.5 The Instruction-Free Length Control

Line 4 of Table 2 requires a short response that was never *told* to be short. We generate 20 unconstrained samples per prompt on a completely clean prompt, then select the samples whose length lands nearest the corresponding constrained response. Length is matched by selection rather than by instruction, so the comparison isolates the cost of being instructed from the cost of writing less.

Table 30: **At matched length, an instructed-short answer is less comprehensive than a naturally short one.** In the rows, both arms are scored against the same unconstrained baseline by the same judge. In a direct head-to-head comparison pooled over 53 pairs, instructed-short is -7.4% relative to natural-short (95% CI $[-11.3, -3.1]$, $p = 0.0004$).

Constraint	Natural-short	Instructed-short	Instruction cost	n
No comma	-14.3%	-24.2%	$\mathbf{-9.9}$ pp	10
No "the"	-14.3%	-23.2%	$\mathbf{-8.9}$ pp	7
No bullet	-5.7%	-8.0%	-2.3 pp	36

Interpretation and caveats. The per-constraint cells for `no_comma` and `no_the` are small ($n = 10$ and $n = 7$) for the structural reason described below, and the pooled estimate is carried by `no_bullet`'s 36 pairs, which has the *smallest* instruction cost; -7.4% therefore likely understates the effect. Selection bias runs in the conservative direction: a naturally short sample might be short because it is a weaker answer, which would make natural-short look worse, but natural-short in fact looks better.

The collapse lies outside the model's natural length range. The small cell sizes are themselves a result: given 20 unconstrained draws per prompt, Llama's shortest natural answer is 393 words on the prompts used for `no_comma` and `no_the`, while the constrained targets are 261 and 245 words. Only 25% and 18% of constrained targets are reachable within 25% by any natural sample. For `no_bullet`, where the collapse is mild, the shortest natural sample is 429 words against a 409-word target and 90% of targets are reachable. Even the shortest answer the model produces on its own in 20 attempts is roughly 50% longer than its answer under a comma ban, so the collapsed response lies outside the unconstrained length distribution.

H Atomic Claim Coverage

As a complementary, length-invariant description of content loss, we extract atomic factual claims from each *unconstrained* response and check how many survive in the corresponding constrained

response. For 8 stratified prompts (2 per category) across all three primary models (192 baseline–constrained pairs), GPT-4o extracts 11–20 atomic claims per unconstrained response and then judges, with generous paraphrase matching, whether each is conveyed in the constrained response.

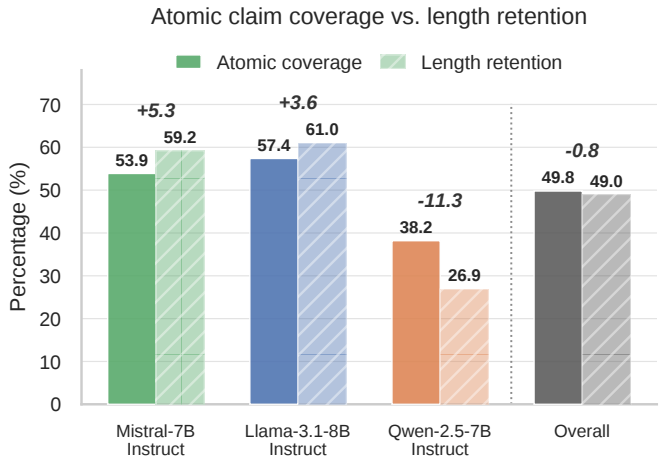

Figure 7: **Atomic claim coverage and length retention.** 192 pairs, 3,355 atom checks. Numerical values in Table 31.

Table 31: **Atomic claim coverage.** Fraction of the unconstrained response's individual factual claims that survive in the constrained response, with generous paraphrase matching. Coverage is length-invariant by construction. 192 pairs, 3,355 atom checks.

Model	Atomic Coverage	Length Retention	Gap
Mistral-7B-Instruct	53.9%	59.2%	+5.3 pp
Llama-3.1-8B-Instruct	57.4%	61.0%	+3.6 pp
Qwen-2.5-7B-Instruct	38.2%	26.9%	−11.3 pp
Overall	**49.8%**	**49.0%**	**−0.8 pp**

Constrained responses retain 49.8% of the baseline's individual factual claims on average. Two cells are worth noting because coverage and length retention diverge in them. Qwen's shortened responses are unusually dense per word and still omit 62% of baseline claims (−11.3 pp gap). The `no_bullet` constraint, averaged across models, retains 89% of baseline length but only 58% of claims (+31 pp gap): the responses stay long but still lose content.

I Per-Layer Probing Details

Table 33 gives the full per-layer Ridge R^2 for predicting response length from hidden states at the last prompt token, before generation begins. Probes are fitted across the three prompt conditions (baseline, `no_comma`, `no_the`) with 5-fold cross-validation. The elevated R^2 at layer 0 for Qwen (0.766) likely reflects the probe reading the constraint condition from surface token embeddings rather than a planning representation; the interpretively meaningful signal is the peak at mid-layers. In a separate probing run that includes the placebo bans of §5.2, with each prompt held out as a block, R^2 at the embedding layer is -0.037 (Llama), -0.007 (Qwen) and -0.019 (Mistral), and rises to 0.582, 0.862 and 0.629 at the best middle layer (layers 16, 14 and 16). The banned words are fully present at the embedding layer, so a probe that read them would work best there; it does not.

Grouped cross-validation. To rule out inflation from same-prompt leakage across folds, we re-run probing with grouped splits (grouped by base prompt, so all variants of a prompt fall in the same fold). Peak R^2 retains about 95% of the standard value: Llama 0.747 → 0.711, Qwen 0.925 → 0.878. The signal is not driven by prompt-identity memorization.

Tokenization check. Chat templating can introduce a duplicated beginning-of-sequence token when a templated string is re-tokenized with special tokens enabled. We measured its effect on

Table 32: Per-model, per-constraint atomic coverage and length retention. Each cell averages over 8 prompts. Gap = length retention − coverage.

Model	Constraint	Coverage	Length Ret.	Gap
Llama	No comma	55.0%	51.8%	−3.2 pp
	No colon	54.6%	53.8%	−0.8 pp
	No semicolon	56.9%	53.6%	−3.3 pp
	No bullet/lists	59.1%	92.5%	+33.4 pp
	No "the"	59.9%	49.2%	−10.7 pp
	No disc. mkrs	62.0%	64.3%	+2.3 pp
	No comma+colon	56.4%	51.2%	−5.2 pp
	No comma+bullet	55.2%	71.6%	+16.4 pp
Qwen	No comma	33.9%	10.5%	−23.4 pp
	No colon	33.2%	20.1%	−13.1 pp
	No semicolon	42.4%	31.1%	−11.3 pp
	No bullet/lists	55.6%	69.0%	+13.4 pp
	No "the"	28.5%	15.0%	−13.5 pp
	No disc. mkrs	40.9%	29.4%	−11.5 pp
	No comma+colon	31.5%	11.7%	−19.8 pp
	No comma+bullet	39.7%	28.2%	−11.5 pp
Mistral	No comma	49.9%	39.2%	−10.7 pp
	No colon	52.3%	52.1%	−0.2 pp
	No semicolon	56.7%	59.3%	+2.6 pp
	No bullet/lists	59.2%	105.5%	+46.3 pp
	No "the"	53.0%	52.1%	−0.9 pp
	No disc. mkrs	48.5%	61.9%	+13.4 pp
	No comma+colon	51.6%	32.3%	−19.3 pp
	No comma+bullet	59.8%	71.2%	+11.4 pp
Overall		**49.8%**	**49.0%**	**−0.8 pp**

Table 33: **Linear probe results for instruction-tuned models.** Ridge R^2 for predicting response length from hidden states at the last prompt token. All three models peak at ∼50% depth, with R^2 tracking collapse severity.

	Llama-3.1-8B-Inst.		**Qwen-2.5-7B-Inst.**		**Mistral-7B-Inst.**	
Depth	Idx	R^2	Idx	R^2	Idx	R^2
0%	0	0.305	0	0.766	0	−0.204
25%	8	0.735	7	0.883	8	0.429
50%	16	**0.747**	14	**0.925**	16	**0.662**
75%	24	0.730	21	0.921	24	0.508
100%	31	0.728	27	0.868	31	0.504

peak probe R^2 under grouped cross-validation: Llama 0.711 vs. 0.695 and Mistral 0.662 vs. 0.657 with the duplication removed (Qwen unaffected). The effect is at most 0.015 and does not alter any conclusion; all steering results in §6.3 use single-BOS tokenization throughout.

30-sample base-model control. To confirm that the base model's negative R^2 reflects genuine absence of signal rather than insufficient sample size (30 vs. 120), we subsample the instruction-tuned data to 30 random points and re-run probing (averaged over 10 random seeds). The instruction-tuned model still yields $R^2 = 0.613$ (Llama) and 0.789 (Qwen) at 30 samples, against base-model values of -4.04 and -0.59 at the same size.

I.1 Probing Control: a Non-Collapsing Constraint

A probe that had merely learned to detect the presence of a constraint clause would predict a short answer whenever a clause is present. To separate that from a probe reading planned length, we need a constraint that is worded like every other constraint but does not cause collapse. On Mistral, `no_bullet` is exactly that case: comprehensiveness changes by +3.9% (95% CI [−1.2, +9.4]) and answers stay near their unconstrained length. We fitted a single probe across three conditions on Mistral, 40 prompts each, and report out-of-fold per-prompt error.

We also attempted to construct a non-collapsing constraint deliberately, by banning words that essentially never appear in an answer (`zephyr`, `bezoar`) so that obeying costs nothing. Those still reduce comprehensiveness by up to 13.1% on Llama, 24.1% on Qwen and 10.9% on Mistral (see

Table 34: **The probe tracks planned length rather than clause presence.** Out-of-fold per-prompt error of a single Mistral probe across three prompt conditions. The probe is *most* accurate on the constraint-bearing condition that does not collapse, which is the opposite of what a clause-detection shortcut would produce.

Prompt condition	Mean answer length	MAE	RMSE
No constraint	378.1	17.1%	21.1%
Plus "no bullet points" (no collapse)	354.8	**11.7%**	14.1%
Plus "no commas or colons" (heavy collapse)	131.0	44.1%	63.5%

also §5.2). Within the family of lexical bans we could not construct an inert constraint, which is why the naturally inert case above is the control we use.

J Steering: Additional Details

J.1 Direction Diagnostics

Table 35: **The steering direction is well determined by the available prompts.** Split-half cosine is between directions fitted on two disjoint halves of the prompt set; clause split-half is the same for the clause-presence direction; per-constraint cosine is between directions fitted on `no_comma` and on `no_the` separately.

Model	Split-half cos	Clause split-half	Per-constraint cos	$\|\mathbf{d}\|/\|\mathbf{h}\|$
Qwen-2.5-7B	**0.980**	0.954	0.818	0.659
Llama-3.1-8B	0.884	0.881	0.827	0.503
Mistral-7B	0.919	0.896	0.847	0.720

Directions fitted on disjoint halves agree at $\cos = 0.88$–0.98, so 28 prompts are ample and the direction is essentially determined by 14. Per-constraint cosines of 0.82–0.85 justify pooling `no_comma` and `no_the` into a single direction. **Leakage check:** directions fitted on prompts 12–39 and evaluated on prompts 0–11 (zero overlap) reproduce directions fitted on all 40, slightly more strongly if anything (`no_comma` at $f = 0.45$: 344.6 vs. 344.8 words).

J.2 Obtaining Compliance from the Direction Alone

The decoder mask is needed because the steering direction is substantially aligned with clause absence (§6.3). A cleaner alternative would be to project that component out and steer along the residual, $\mathbf{d}_\perp = \mathbf{d} - (\mathbf{d} \cdot \hat{\mathbf{u}}_{\text{clause}})\, \hat{\mathbf{u}}_{\text{clause}}$. We tested this alternative.

On **Qwen** this fails: $\mathbf{d}_\perp$ preserves compliance (96–100%) but recovers only 1–12%, while a matched-norm random direction at the same doses recovers 5–18%. It is indistinguishable from the placebo, which indicates that the planning signal and the constraint-removal signal are not linearly separable at layer 15. On **Llama** it partially succeeds: at $f = 0.15$ it gives 33% recovery at 88% compliance against -3% for random, a real, direction-specific and compliance-preserving effect, but small, and it disappears by $f = 0.22$. On **Mistral** the test is uninterpretable because baseline compliance on these constraints is near zero. The decoder mask is therefore a principled component rather than a shortcut, because the cleaner alternative was tried and mostly does not work.

When the mask matters. On `no_semicolon`, steering with and without the mask gives identical results (Llama 375.5 words both; Mistral 265.0 vs. 264.3, 96% vs. 100% compliance): the mask never fires, because steering does not reintroduce a semicolon the model was not going to emit anyway. Whether the mask is needed therefore tracks banned-token frequency, the same quantity that governs collapse severity in §5.2.

J.3 Steering Across Models

Table 36 reports the initial two-seed run on all three primary models; the confirmatory three-seed run for Qwen and Llama is Table 19.

Table 36: **Steering recovery across all three primary models, initial two-seed run** (steer + mask at each model's best dose, judged comprehensiveness). Qwen recovers 64–78%, Llama 24–42%, Mistral ∼25% and within its own noise. The placebo behaves correctly throughout: Llama's matched-norm random direction + mask gives −12% to −19%, i.e. worse than baseline.

Model	Constraint	Baseline Δ%	Steered Δ%	Points	Recovery
Qwen-2.5-7B	No "the"	−43.2	−9.7	33.5	**78%**
Qwen-2.5-7B	No comma	−64.5	−23.1	41.5	**64%**
Llama-3.1-8B	No semicolon	−28.7	−16.5	12.2	42%
Llama-3.1-8B	No comma	−30.0	−22.6	7.3	24%
Mistral-7B	No semicolon	−13.1	−9.8	3.3	25% (within noise)

J.4 Constructing the Contrast Set

The direction is built by pooling `no_comma` and `no_the` prompts. Pooling also matters for performance: a direction built from `no_comma` alone has a *larger* raw contrast ($\|\mathbf{d}\|/\|\mathbf{h}\| = 0.763$ vs. 0.658) yet performs substantially worse, giving 44% comprehensiveness recovery against 64% at the same layer, dose, prompts and seeds (223 vs. 350 words). This is consistent with the clause-alignment result of §6.3: the clause-specific components of the two constraints mention different tokens and partially cancel under pooling, while the shared "plan a short answer" component reinforces. Pooling therefore acts as a regularizer that strips constraint-specific content, and the high per-constraint cosine (0.82–0.85, Table 35) indicates a large shared component to recover.

K Token-Level Strategy Divergence

We run the model forward token-by-token on 5 prompts × 2 constraints, recording the top-50 token probability distribution at each of the first 20 generated positions for both the constrained and unconstrained prompts, and measure Jensen–Shannon divergence (JSD) and top-50 overlap at each position.

Table 37: **Token-level divergence** between constrained and unconstrained generation. JSD rises rapidly in the first 3–5 tokens and saturates across all three models: the model commits to a different response strategy within the opening tokens.

	Llama		**Qwen**		**Mistral**	
Position	JSD	Overlap	JSD	Overlap	JSD	Overlap
1–3	0.544	35%	0.460	42%	0.512	36%
4–10	0.983	8%	0.873	13%	0.780	22%
11–20	0.995	9%	0.908	13%	0.896	15%

Qualitatively, unconstrained Llama opens with markdown formatting (e.g., "`**Gradient Descent: A Simple Explanation**`") while the constrained model opens with plain prose ("`Gradient descent is a way to find`"), so the constrained model commits to a different response strategy from the first token.

L Deployment Constraint Details

Constraint text. The four deployment constraints tested in §7.1:

- **Professional tone** (brand guidelines): "Do not use exclamation marks, casual language, or informal expressions. Maintain a strictly professional and formal tone throughout your response."
- **No preamble** (API efficiency): "Do not begin your response with a greeting, acknowledgment, or conversational opener such as 'Certainly!', 'Great question!', 'I'd be happy to help!', 'Sure!', or 'Of course!'. Start directly with the first substantive sentence of your answer."
- **Hedging language** (legal/compliance): "Avoid making definitive or absolute claims. Use hedging language such as 'may,' 'might,' 'could,' or 'evidence suggests' instead of stating facts directly."

- **Plain language** (accessibility): "Write at a reading level accessible to a general audience. Avoid all technical jargon, acronyms, and complex sentence structures. Use simple, everyday words and short sentences."

No token-level form. None of these four constraints can be written as a token mask, which is why decoder-level enforcement (§5) is a diagnostic rather than a remedy.

The professional-tone tax. Requesting a professional, formal tone is the default system prompt for a large fraction of corporate and customer-facing LLM deployments. It places no restriction on length, vocabulary, structure or factual depth, yet costs Llama 14.4% and Qwen 18.0% of comprehensiveness. On Mistral it shows no significant effect (-0.3%, $p = 0.44$, win rate 50%), so we report it as an effect on two of three models rather than as a general tax.

The opening tokens. The no-preamble constraint restricts *only* the opening of the response; no restriction is placed on the tone, vocabulary, structure or content of the remainder. It costs 13.8–34.3% across the three primary models, and 14–34% across all seven instruction-tuned models (Table 38). This connects directly to the token-level divergence analysis (Appendix K): the response strategy is fixed within the first 1–3 tokens, so a constraint on the opening tokens affects the whole response.

The hedging constraint. Instructing a model to write "may" or "evidence suggests" instead of definitive claims places no limit on length, depth or organization, yet this standard legal/compliance constraint costs 12.1–24.7% across all three models.

Plain language: a partially confounded upper bound. Plain language produces the largest degradation (-30.2% Llama, -45.4% Qwen, -27.1% Mistral), but the constraint explicitly instructs the model to avoid technical jargon and use simple words. For technical prompts, some comprehensiveness loss is an *intended* consequence, so this figure contains both unintended template collapse and intended complexity reduction. With it excluded, the deployment effect is -15.8% (Llama), -25.7% (Qwen) and -8.7% (Mistral), which is weaker but free of this confound.

Table 38: **No-preamble collapse across all seven instruction-tuned models.** Suppressing only the conversational opener causes 14–34% comprehensiveness loss across every family, scale and architecture tested. GPT-4o pairwise judge, 40 pairs per model.

Model	Family	Size	$\Delta\%$	Win%	Word $\Delta\%$
Mistral-7B-Instruct-v0.3	Mistral	7B	-13.8	85	-30
Llama-3.1-8B-Instruct	Llama	8B	-14.7	92	-54
GPT-4o-mini	GPT	closed	-18.8	95	-37
OLMo-3-7B-Instruct	OLMo	7B	-29.1	95	-70
Llama-3.3-70B-Instruct	Llama	70B	-30.4	100	-57
Qwen3-30B-A3B-Instruct	Qwen 3 (MoE)	30B (3B act.)	-30.9	100	-72
Qwen-2.5-7B-Instruct	Qwen 2.5	7B	-34.3	100	-75

M CONSTRAINT-AUGMENTED LORA: FULL RESULTS

Training uses verified, constraint-satisfying two-pass examples for two constraints (`no_comma`, `no_the`); all other constraints are held out. Recovery figures in the main text are computed on the subset where LoRA constraint compliance remains $\geq 85\%$.

Compliance notes. Colon-related constraints show reduced LoRA compliance on both models; for Llama this mirrors the original model's own low colon satisfaction (65.8%; Appendix B). Qwen additionally shows reduced bullet compliance (55%). The compliant subset reported in the main text conservatively excludes these constraints so that measured recovery reflects genuine constraint-satisfying comprehensiveness. The filter is conservative rather than selective: including *all* held-out constraints regardless of compliance raises the figures to 68% (Llama) and 88% (Qwen), because the excluded low-compliance cells show higher apparent recovery precisely where the adapter is least often obeying the constraint.

Table 39: **Full LoRA mitigation results with constraint compliance.** Comprehensiveness $\Delta\%$ vs. the original unconstrained baseline (GPT-4o judge, 40 pairs per cell). Sat% = constraint satisfaction rate. ✓ = included in the main-text compliant subset ($\geq$85%).

	Llama-3.1-8B			Qwen-2.5-7B		
Constraint	Orig Δ%	LoRA Δ%	Sat%	Orig Δ%	LoRA Δ%	Sat%
Training constraints						
No comma	−31.4	−1.8	98✓	−66.3	−14.4	98✓
No "the"	−26.9	−8.2	92✓	−49.6	+2.5	85✓
Held-out constraints						
No colon	−30.6	−8.6	32	−42.6	−3.6	22
No semicolon	−25.2	−3.0	98✓	−39.8	+0.0	88✓
No bullet	−22.8	−16.5	100✓	−18.9	+0.0	55
No disc. mkrs	−23.1	+0.0	90✓	−38.2	−1.8	92✓
No cm+colon	−36.9	−9.6	50	−65.2	−9.7	95✓
No cm+bullet	−27.2	−15.3	90✓	−46.2	−14.5	95✓
Compliant held-out avg ($\geq$85% satisfaction)						
	−24.6	−8.7	$\geq$90	−47.4	−6.5	$\geq$88

Transfer to deployment constraints. We additionally train a second Llama adapter with the same settings on 56 constraint-satisfying examples of `no_comma` and `no_the`, and evaluate it on the four deployment constraints of §7.1, none of which is a token ban and none of which it saw in training. Both columns are measured against the same reference, the unadapted model's unconstrained answer to the same question. With no constraint in the prompt, the adapted model scores +12.1% against the unadapted model.

Table 40: **Zero-shot transfer of the second LoRA adapter to deployment constraints** (Llama-3.1-8B-Instruct, GPT-4o judge, 40 prompts per constraint). The adapter was trained on 56 examples of two lexical token bans and never saw any of these constraints.

Deployment constraint	Llama-3.1-8B-Instruct	+ LoRA
Hedging	−18.2%	−9.2%
No preamble	−15.7%	+4.8%
Plain language	−29.8%	−17.4%
Professional tone	−14.9%	+5.4%
Overall	**−19.6%**	**−4.4%**

Baseline quality. Head-to-head pairwise comparison of *unconstrained* responses (original vs. LoRA, GPT-4o judge, 40 pairs) confirms no baseline degradation: Llama original wins exactly 50% of pairs; Qwen's adapter wins 62.5%.

N Extended Discussion

Underlying response changes. The comprehensiveness loss reflects large structural changes. On Llama-3.1-8B-Instruct, banning commas reduces average response length by 57% (685 to 297 tokens), unique content words by 41%, and formatting richness (bold text, headers, code blocks, bullet points) from 2.3 to 0.4 on a 0–4 scale. Instead of writing the same content without commas, the model produces a different, minimal response.

What instruction tuning learns. Our results suggest that instruction tuning's apparent helpfulness is, in part, an artifact of learning a narrow distribution of response templates rather than developing a generalizable competence for providing thorough information. When a high-frequency token is banned (a formatting character, a structural element, or a common content word), the learned templates become inaccessible and the model defaults to a minimal response. The decomposition in §4.2 provides the most direct evidence: content criteria fall 1.9–3.1× further than surface criteria, and on Mistral surface quality does not move at all. The frequency gradient of §5.2 supplies the dose–response: the loss grows with how much of the default answer depended on the banned material.

From diagnostic probes to deployment constraints. One objection is that the specific lexical constraints we study are artificial, since no real user asks for an explanation of gradient descent without commas. This choice is intentional: our lexical constraints serve as *controlled diagnostic probes*, similar to stress tests in structural engineering. No one drives a 100-ton truck across a pedestrian bridge, but if the bridge fails under that load, the failure reveals a weakness that normal traffic would not. We validate the analogy empirically in §7.1: realistic deployment constraints produce collapse of comparable magnitude, with hedging alone costing 12–25% on all three models despite placing no limit on response length, depth or structure.

Implications for evaluation and deployment. The 2.0–6.8× gap between independent and pairwise evaluation implies that any constrained generation system evaluated solely by independent scoring may carry undetected quality loss. The deployment results make this concrete: requesting a professional tone costs 14–18% on two of three models, and suppressing the conversational preamble costs up to 34%. Combined with the GPT-4o-mini result, which shows a widely deployed commercial model losing 31%, these findings suggest the industry may be systematically overestimating the usefulness of unmodified instruction-tuned models in constrained production environments. Practitioners should test their deployed models against the specific constraints they impose, using pairwise evaluation.

Why decoder enforcement is a diagnostic rather than a remedy. The decoder-enforced arm (line 1 of Table 2) removes almost all of the loss, which might suggest that constrained generation should simply be moved to the decoder, but two facts prevent this. First, decoder-level enforcement requires logit access, which rules out every model served through an API. Second, most constraints of practical interest have no token-level expression: of the eight constraints in our main results, four can be written as a token mask, while `no_bullet`, `no_discourse` and the two compositional constraints cannot, because they are properties of structure rather than of individual tokens. None of the four deployment constraints is expressible as a mask at all. Deployed systems append these constraints to prompts because that is the only interface most of them have, so the phenomenon we report applies to the way constraints are actually given in practice.

Toward robust instruction tuning. The two-pass result demonstrates feasibility: models possess the capability to produce comprehensive constrained output. The steering results localize the failure to a single mid-network site (∼50% depth) among the seven depths tested and show it is causally reversible there. The scaling results indicate that training larger models is not a solution, and models with higher baseline quality may even collapse more severely. Promising directions include constraint-augmented alignment training (Yuan et al., 2025), representation-level interventions (Stolfo et al., 2025), and training with diverse surface-form constraints to decouple competence from any particular template.

O Evaluation Prompt List

Our evaluation set consists of 40 prompts across four categories (10 per category), designed to elicit substantive, multi-paragraph responses that benefit from structured formatting.

Category 1: Explanation / Education (10 prompts)

1. Explain gradient descent in simple terms.
2. What is photosynthesis and why is it important for life on Earth?
3. How does a computer CPU process instructions?
4. Explain the concept of supply and demand in economics.
5. What is machine learning and how does it differ from traditional programming?
6. Explain how vaccines work to protect against diseases.
7. What is quantum computing and why is it potentially revolutionary?
8. Explain the water cycle and its importance for the environment.
9. How does encryption work to keep data secure?

10. What is the theory of evolution and what evidence supports it?

Category 2: How-To / Advice (10 prompts)

1. How should I prepare for a technical job interview?
2. What are the best practices for writing clean, maintainable code?
3. How can I improve my public speaking skills?
4. What steps should I take to start investing in the stock market?
5. How do I write an effective research paper?
6. What are good strategies for managing stress and anxiety?
7. How should I approach learning a new programming language?
8. What are the key steps to starting a small business?
9. How can I improve my time management skills?
10. What should I consider when choosing a graduate school program?

Category 3: Analysis / Comparison (10 prompts)

1. Compare renewable and non-renewable energy sources.
2. What are the advantages and disadvantages of remote work?
3. Compare Python and JavaScript as programming languages.
4. What are the pros and cons of social media for society?
5. Compare different types of database systems and their use cases.
6. What are the benefits and risks of artificial intelligence?
7. Compare democratic and authoritarian systems of government.
8. What are the trade-offs between privacy and security in the digital age?
9. Compare electric vehicles with traditional combustion engine cars.
10. What are the advantages and disadvantages of online education?

Category 4: Technical / Detailed (10 prompts)

1. Explain how a neural network learns through backpropagation.
2. Describe the process of DNA replication in cells.
3. How does the TCP/IP protocol stack work?
4. Explain the CAP theorem in distributed computing.
5. How does a compiler translate source code into machine code?
6. Describe how CRISPR gene editing technology works.
7. Explain the principles behind public key cryptography.
8. How does a recommendation system like Netflix's work?
9. Describe how blockchain technology maintains a secure ledger.
10. Explain how transformer models process natural language.

P CONSTRAINT DEFINITIONS

Lexical Constraints

Each constraint is appended verbatim to the user prompt as an additional sentence.

Punctuation-level constraints:

`no_comma`	"Do not use any commas in your response."
`no_colon`	"Do not use any colons in your response."
`no_semicolon`	"Do not use any semicolons in your response."

Pattern-level constraint:

`no_bullet`	"Do not use bullet points, numbered lists, or dashes to create lists in your response. Write only in flowing prose paragraphs."

Word-level constraints:

`no_the`	"Do not use the word 'the' in your response."
`no_discourse`	"Do not use the words 'however', 'therefore', 'moreover', 'furthermore', or 'additionally' in your response."

Compositional constraints:

`no_comma_colon`	"Do not use any commas or colons in your response."
`no_comma_bullet`	"Do not use any commas in your response. Do not use bullet points, numbered lists, or dashes to create lists. Write only in flowing prose paragraphs."

Frequency-sweep constraints (§5.2). Fifteen additional constraints of the form "Do not use the word 'w' in your response," with w drawn from four frequency bands: *essentially never* (`zephyr`, `quixotic`, `bezoar`, `sassafras`); *rarely* (`utilize`, `nonetheless`, `paradigm`); *sometimes* (`which`, `because`, `important`, `often`); *constantly* (`of`, `and`, `is`, `a`). Band assignment reflects expected frequency in an unconstrained answer to our evaluation prompts.

Q JUDGE PROMPTS

Q.1 INDEPENDENT SCORING (§7.2)

Independent Judge – System Prompt

You are an expert evaluator of AI assistant responses. Your job is to assess the QUALITY of a response to a user query, independent of any formatting constraints the user may have imposed.
Focus on the SUBSTANCE: Is the information accurate, comprehensive, and useful? Ignore formatting — a response can be high quality even without bullet points or structured formatting.
You MUST output ONLY a valid JSON object with exactly four integer scores (1-10), nothing else.

Independent Judge – User Prompt Template

Rate this AI assistant response on 4 dimensions (1-10 each).
SCORING GUIDE:

- **1–3:** Poor. Missing key information, inaccurate, or unhelpfully vague.
- **4–5:** Below average. Covers basics but lacks depth or misses important aspects.
- **6–7:** Good. Covers the topic well with reasonable depth and accuracy.
- **8–9:** Very good. Thorough, accurate, insightful, well-organized.
- **10:** Exceptional. Could not meaningfully be improved.

```
===== USER QUERY =====
{prompt}
```

```
===== AI RESPONSE =====
{response}
===== YOUR TASK =====
```
Rate the response on:
1. informativeness: How much relevant, accurate information is provided?
2. depth: How detailed and thorough is the explanation?
3. clarity: How clear and understandable is the response?
4. helpfulness: Overall, how useful would this be to a typical user?

Output ONLY: `{"informativeness": N, "depth": N, "clarity": N, "helpfulness": N}`

Q.2 Pairwise Comparison (§4.1)

Pairwise Judge – System Prompt

You are an expert evaluator comparing two AI assistant responses to the same question. Your job is to assess how comprehensive, detailed, and useful each response is RELATIVE to the others.
You MUST output ONLY a valid JSON object, nothing else.

Pairwise Judge – User Prompt Template

Below are two responses (Response A and Response B) to the same user question. Rate EACH response on two dimensions (1-10).
SCORING GUIDE — be discriminating, use the full scale:

- **1–3:** Very brief or shallow. Misses most important aspects of the topic.
- **4–5:** Covers basics but lacks detail, examples, or thorough explanation.
- **6–7:** Good coverage with reasonable depth. Addresses key aspects.
- **8–9:** Thorough and detailed. Provides examples, nuances, and a structured explanation.
- **10:** Exceptionally comprehensive. Could not meaningfully be improved.

```
===== USER QUESTION =====
{prompt}
===== RESPONSE A =====
{response_a}
===== RESPONSE B =====
{response_b}
===== YOUR TASK =====
```
Rate EACH response on:
1. comprehensiveness: How thoroughly does it cover the topic? Does it include examples, important details, edge cases, and structured explanation?
2. usefulness: How helpful would this be to someone trying to understand or act on this topic?

Output ONLY: `{"response_a_comprehensiveness": N, "response_a_usefulness": N, "response_b_comprehensiveness": N, "response_b_usefulness": N}`

Note: The assignment of baseline/constrained responses to positions A and B is randomized for each pair to control for position bias. The judge never sees the constraint clause; it receives only the base question.

Q.3 Six-Criterion Instrument (§4.2)

The six-criterion instrument uses the same pairwise system prompt and the same side-by-side presentation, replacing the two output dimensions with six: `semantic_coverage`, `comprehensiveness`, `correctness`, `helpfulness` (content and accuracy), and `conciseness`, `readability` (surface). Criterion definitions and 1–10 anchors are identical in wording to the human-evaluation rubric of Appendix S, so that the two instruments differ in rater rather than in rubric. Content $\Delta\%$ in Table 1 averages semantic coverage, comprehensiveness and helpfulness; surface $\Delta\%$ averages conciseness and readability; correctness is reported separately.

R Coverage Analysis Judge Prompts

Atom Extraction – System Prompt

You are an expert at extracting atomic factual claims from text. Your job is to decompose a response into its distinct informational atoms.
You MUST output ONLY a valid JSON object, nothing else.

Atom Extraction – User Prompt Template

Below is a response to a user question. Extract the distinct factual claims or actionable points as a list of short, standalone sentences.
RULES:

- Each claim should be ATOMIC: one fact, one idea per bullet.
- Each claim should be STANDALONE: understandable without reading the others.
- DO NOT include framing sentences, transitions, or meta-commentary.
- DO NOT include duplicated or paraphrased restatements of the same claim.
- Aim for 8–15 atoms. If the response is shallow, extract fewer. If it is rich, extract more (up to 20).
- Phrase each atom neutrally, in your own words – not as a quote.

```
===== USER QUESTION =====
{prompt}
===== RESPONSE =====
{response}
===== YOUR TASK =====
```
Output ONLY: `{"atoms": ["atom 1", "atom 2", ...]}`

Atom Matching – User Prompt Template

Below is a response to a user question, followed by a specific claim. Your job: determine whether the response CONVEYS THIS CLAIM, even if phrased differently.
MATCHING RULES:

- Answer YES if the response expresses the claim, even in different words, shorter form, or integrated into a larger sentence.
- Answer YES even if the response merely implies the claim clearly.
- Answer NO if the response omits, contradicts, or only partially covers the claim.
- Be GENEROUS on paraphrase: same meaning = YES.
- Be STRICT on substance: missing a key fact = NO.

```
===== USER QUESTION =====
{prompt}  ===== RESPONSE =====
{response}  ===== CLAIM TO CHECK =====
{claim}
===== YOUR TASK =====
```
Output ONLY: `{"covered": "YES" or "NO", "reason": "brief justification (max 15 words)"}`

S Human Evaluation Protocol

Evaluator recruitment and ethics. We recruit 10 evaluators with graduate-level education in STEM fields. Participation is anonymous and voluntary; all evaluators provide informed consent and are told that their ratings will be used to assess AI-generated response quality for a research study. No personally identifying information is collected. Evaluators are compensated for their time.

Task instructions. Each evaluator receives a set of pairwise comparisons. For each, the evaluator sees (1) the original user question and (2) two responses labeled Response A and Response B. One is the unconstrained baseline and the other the constrained response; assignment to positions is

randomized per pair. Evaluators are not informed which response is constrained, nor are they told the nature of the constraints applied. The evaluation is fully blinded with respect to experimental condition. Evaluators are instructed to judge each dimension independently.

Content and Accuracy Criteria

1. Semantic Coverage: How many of the key ideas, subtopics, and important aspects of the question does the response address?

- **1–3:** Covers only one or two aspects; misses most important subtopics.
- **4–5:** Addresses some key points but omits several important aspects.
- **6–7:** Covers most relevant subtopics with minor gaps.
- **8–10:** Covers essentially all relevant subtopics and angles.

2. Comprehensiveness: Beyond mentioning topics, how much depth, detail, examples, and nuance does the response provide?

- **1–3:** Superficial treatment; no examples or elaboration.
- **4–5:** Some detail on a few points, but most topics covered only at surface level.
- **6–7:** Reasonable depth on most points; some examples or elaboration.
- **8–10:** Thorough treatment with examples, nuances, edge cases, and structured explanation.

3. Correctness: Is the information factually accurate?

- **1–3:** Contains significant factual errors or fundamentally misleading claims.
- **4–5:** Mostly correct but contains notable inaccuracies or oversimplifications.
- **6–7:** Accurate with minor imprecisions that do not mislead.
- **8–10:** Factually accurate throughout; claims are well-supported.

4. Helpfulness: Would this response actually help someone understand the topic or complete the task?

- **1–3:** The reader would need to look elsewhere for useful information.
- **4–5:** Provides a starting point but the reader would need supplementary sources.
- **6–7:** Reasonably helpful; addresses the core question with actionable information.
- **8–10:** The reader would feel well-informed and equipped to act after reading this.

Surface Criteria

5. Conciseness: How concise and efficient is the response? Higher scores indicate tighter writing with no unnecessary padding.
Important: Brevity alone does not constitute conciseness. A short response that says little is empty, not concise. A long response packed with useful content is not verbose.

- **1–3:** Excessively padded, repetitive, or filled with filler phrases.
- **4–5:** Some unnecessary repetition or padding, but not egregious.
- **6–7:** Reasonably concise; most content serves a purpose.
- **8–10:** Every sentence contributes meaningfully; no wasted words.

6. Readability: How clear, well-organized, and easy to follow is the response?

- **1–3:** Disorganized, hard to follow, or confusingly written.
- **4–5:** Understandable but could be better organized or clearer.
- **6–7:** Clear and well-organized; easy to follow.
- **8–10:** Exceptionally clear; logical flow, effective use of structure.

Methodological rationale. The separation into content and surface criteria is the key design choice. If constrained responses score lower on coverage, comprehensiveness and helpfulness but comparably on conciseness and readability, the collapse reflects genuine information loss rather than a length-preference artifact. The same six criteria and anchors are used by the automated six-criterion instrument of §4.2, so the two studies differ in who rates rather than in what is asked.

T TWO-PASS QUALITATIVE EXAMPLES

We present a representative example from the two-pass experiment (§6.1) on Qwen-2.5-7B-Instruct: the prompt "Explain how vaccines work to protect against diseases" under the constraint "Do not use the word 'the' in your response."

Baseline Response (unconstrained, 402 words)

Vaccines work by stimulating the immune system to recognize and fight specific pathogens (such as viruses or bacteria) without causing the disease itself. Here's a step-by-step explanation of how this process works:
1. **Introduction of Antigen:** The vaccine contains antigens, which are usually weakened or inactive forms of the pathogen (or parts of it). These antigens are introduced into the body, either through injection or another route.
2. **Immune System Recognition:** When the vaccine enters the body, the immune system recognizes these antigens as foreign substances. Specialized cells in the immune system, including dendritic cells, capture and process the antigens . . .
[continues for 402 words with 7 numbered steps, bold headers, and detailed explanations]

Single-Pass Response (constrained, 55 words, 86% shorter)

Vaccines introduce small amounts of weakened or dead microorganisms into body, prompting immune system to recognize these invaders. Upon exposure, immune cells produce antibodies specific to antigens found on surface of these microorganisms. Afterward, if actual infection occurs, body quickly produces antibodies, neutralizing pathogens before illness develops. This process provides protection without causing disease.

Two-Pass Response (rewrite of baseline under constraint, 323 words, 80% retention)

Vaccines operate by prompting immune system to identify and combat particular pathogens (like viruses or bacteria) without inducing disease. Here's a detailed breakdown of this process:
1. **Introduction of Antigen:** Vaccines contain antigens, typically weakened or inactive forms of pathogen (or parts of it). These antigens are introduced via injection or another method.
2. **Immune System Recognition:** When vaccine enters a body, immune system identifies antigens as foreign substances. Specialized cells, including dendritic cells, capture and process antigens . . .
[continues for 323 words with same 7-step structure, preserving detail and organization]

The single-pass response collapses to a single paragraph of 55 words, losing all structure, examples and detail, whereas the two-pass response preserves the original's numbered structure, bold headers and substantive content while successfully avoiding the word "the."

Two-pass rewrite prompt. For reproducibility, the exact rewrite instruction is: *"Below is a response to a question. Please rewrite it to convey the same information with the same level of detail and comprehensiveness, but [constraint text]. Keep the response equally thorough and informative. Original response: [baseline]. Rewritten response:"*